

Ultra Low-Power Neural Inspired Addition: When Serial Might Outperform Parallel Architectures

Valeriu Beiu¹, Asbjørn Djupdal², and Snorre Aunet³

¹ School of Electrical Engineering and Computer Science,
Washington State University, USA
vbeiu@eecs.wsu.edu

² Norwegian University of Science and Technology,
Department of Computer Science and Information Technology, Norway
djupdal@idi.ntnu.no

³ University of Oslo, Department of Informatics, Norway
sa@ifi.uio.no

Abstract. In this paper we analyse a serial (ripple carry) and a parallel (Kogge-Stone) adder when operating in subthreshold at 100nm and 70nm. These are targeted for ultra low power consumption applications. The elementary gates used are threshold logic gates (perceptrons). Simulations have been performed both with and without considering the delay on the wires. These simulations confirm that wires play a significant role, reducing the speed advantage of the parallel adder (over the serial one) from 4.5x to 2.2–2.4x. A promising result is that the speed of both adders improves more than 10x when migrating from 100nm to 70nm. The full adder based on threshold logic gates (used in the ripple carry adder) improves on previously known full adders, achieving 1.6fJ when operated at 200mV in 120nm CMOS. Finally, the speed of the parallel adder can be matched by the serial adder when operating at only 10–20% higher V_{dd} , while still requiring less power and energy.

1 Introduction

The scaling of CMOS in the few tens of nm range, brings with it many challenges [1]. A recent analysis of these challenges [2], has revealed (once more) that the most important ones are: power consumption (and the dissipation of the heat associated to it), and reliable functioning (i.e., both defect- and fault-tolerance). In this paper we shall focus on neural inspired ultra low power solutions. We will show that neural-inspired gates, combined with serial architectural approaches might outperform parallel solutions when run at the same speed in subthreshold scale CMOS.

1.1 Power Consumption

The power consumption of a CMOS integrated circuit contains three parts: dynamic power, short-circuit power, and static power. The dynamic average power component (capacitive switching) has seen a fair share of analysis and numerous mature methods to manage it. The short-circuit component has only rarely been considered, and it is quite common to ignore it, or estimate its contribution to about 10% of the dynamic power. The third component is the static power, and this has recently become important. Static power includes both leakage (which translates into stand-by power) and DC currents. The drastic increase of the leakage currents is raising concerns that, for future technology nodes, the standby-power will become larger than dynamic power, and is the culprit of scaling. Therefore, reducing leakage by design has started to be investigated.

1.2 Subthreshold Operation

One approach in the bid to lower power dissipation is the reduction of V_{dd} , obviously a very effective way of reducing all the components of power. The aggressive scaling of V_{dd} to below V_{Th} (known as subthreshold operation) has been known, and used in ultra low power designs [3, 4, 5]. The major disadvantage is the very slow speed, and subthreshold operation has been considered a poor approach, in that the much-needed speed is sacrificed for ultra low power (limiting its application range). Still, the papers reporting on subthreshold designs are mostly based on older technology nodes. The advantage of subthreshold operation is that it puts (at least part of) the leakage currents to good use.

1.3 Scaling and Optimal Design

The scaling of subthreshold designs was not properly investigated, and we expect that subthreshold could become an interesting design approach particularly because the operation speeds would be improved as scaling proceeds towards smaller technology nodes. It is not difficult to envision a situation in which designs in older technology nodes operating at standard power supply voltages would have comparable operation speeds to those in advanced technologies at subthreshold voltages. This would mean that for example a microprocessor designed to run at 1 GHz in 0.18 μm at normal V_{dd} might be redesigned to operate at 1 GHz in subthreshold in say 45 nm. The main advantage would be a power reduction of a few orders of magnitude. If these were confirmed, the range of applications that could enjoy ultra-low power and improved speed would increase significantly. It must also be noted here that the available literature on subthreshold CMOS have ring oscillators as the main type of circuits studied, while only a few articles discuss gates, and small systems are the exception rather than the rule:

- a 16-bit serial adder [6];
- a 32-bit adder [7] (we infer that the adder is serial, as the authors mention that it has 2.3k transistors, using standard CMOS gates); and
- an 8x8 serial multiplier [8].

Only very recently, analysis of the optimality of subthreshold designs has started to be explored: (i) transistor level optimisations have been reported in [9], (ii) transistor sizing have only recently been discussed [10], while a very preliminary comparison of architectural approaches will be presented in this paper. It is obvious that subthreshold operation shows different design tradeoffs, and that optimal designs (at the transistor, gate, and system level) have yet to be identified.

1.4 Architectural Approaches

The particular example we are going to use in this paper is a 32-bit adder. Many different design alternatives are possible. It is commonly considered that the slowest one is the ripple carry adder (RCA, or serial solution), while the Kogge-Stone is expected to be the fastest. Classical CMOS gates are almost never used, when fast addition is in the picture, and domino gates are the norm.

Threshold logic gates (perceptrons) have also been advocated for fast addition, as they allow for shallow neural-inspired structures [11]. This theoretically leads to very high speeds [12], and a mixed dynamic and threshold logic is probably the fastest [13]. Threshold logic gates working in subthreshold have been reported in [14, 15], and this type of gate has been used for the circuits presented here. Finally, highly optimised threshold logic adders have also been used for achieving higher speeds in inherently slower single electron technologies [16].

Scaled power supply voltages lead to additional difficulties, e.g. noise margins. Process and environmental variations will also be major factors in designing systems that operate in subthreshold (due to the exponential dependence of the leakage current with respect to gate-to-source, threshold and drain-to-source voltages). Reducing the voltage supply might save the day for power consumption, but will adversely affect reliability.

The solution to this problem is to add redundancy to the design. From this perspective an RCA is easier to integrate with e.g. MAJ-3 multiplexing than a more complex parallel adder [17].

2 Serial and Parallel Adders

2.1 Details of the Circuits Used

Two 32-bit adder structures have been investigated. Both use the MAJ-3 gate of Fig. 1(a) as the basic logic gate.

The first adder is an RCA having a critical path of 32 stages. Each stage is a full-adder like the one in Fig. 1(c) [18]. This full adder has improved the PDP from 6.51fJ down to 1.57fJ and the power dissipation from 15.5nW to 7.3nW, at $V_{dd} = 200mV$ in STM 120nm compared to all previously published full adders using the same basic logic gate. This RCA implementation has 704 transistors.

The second adder is a Kogge-Stone parallel prefix adder [19]. This adder can be seen in Fig. 2(a), while the different building blocks have been implemented using MAJ-3 gates (see Fig. 1(a)) and are detailed in Figs. 2(b), 2(c) and 2(d). This adder uses a total of 4180 transistors, and has a critical path of seven stages.

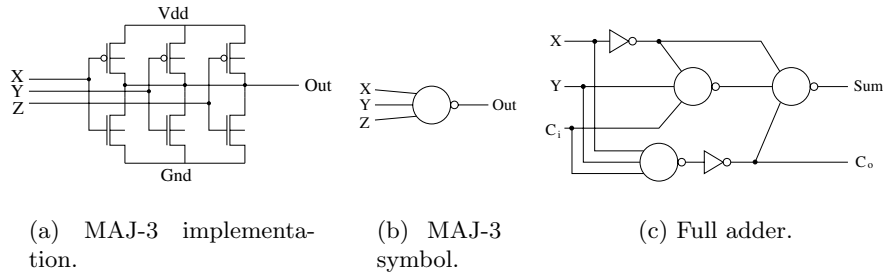


Fig. 1. MAJ-3 logic gate and full adder

Theoretically, the relative delay of RCA vs. Kogge-Stone should be $\frac{32}{7} = 4.5$.

2.2 Device Models and Experimental Setup

All simulations were done in Spice3f5 (ngspice). Berkeley Predictive Technology Models (BPTM) [20] have been used for all transistors. Each BPTM node was generated with the standard parameters suggested in [20].

Each input wire to an adder is driven by a gate, and each output has a gate as a load.

The addition performed in all experiments was $0xffffffff + 1$, which exercises the longest / critical path (from bit 0 to the carry out). Delays are measured from the time the input signal is asserted till the carry out reaches $\frac{V_{dd}}{2}$.

Energy was estimated from the current flowing to V_{ss} during the addition.

2.3 Evaluating Wires

For the simulations where wire delays were included, all wires have been modelled using the four-segment π -model. Parameters for the wires have been determined using the BPTM interconnect calculator with the values suggested for each node.

Length estimates of the longer wires were found by assuming that each building block of the Kogge-Stone adder in BPTM 70nm is $7 \times 7 \mu\text{m}^2$. The length of the shorter wires (which are the only ones present in the RCA adder) have been estimated based on a layout of the MAJ-3 gate. For other technology nodes, these lengths have been scaled accordingly.

3 Results and Discussions

The results of the simulations are synthesised in Table 1. The most important ones are also presented in Figs. 3 and 4.

As expected, the parallel adder is faster than RCA when the same V_{dd} , technology node, and transistor sizes are used.

Simulations without considering wire delays are in very good agreement with the theoretical $\frac{32}{7} = 4.5$ speed advantage of the parallel adder.

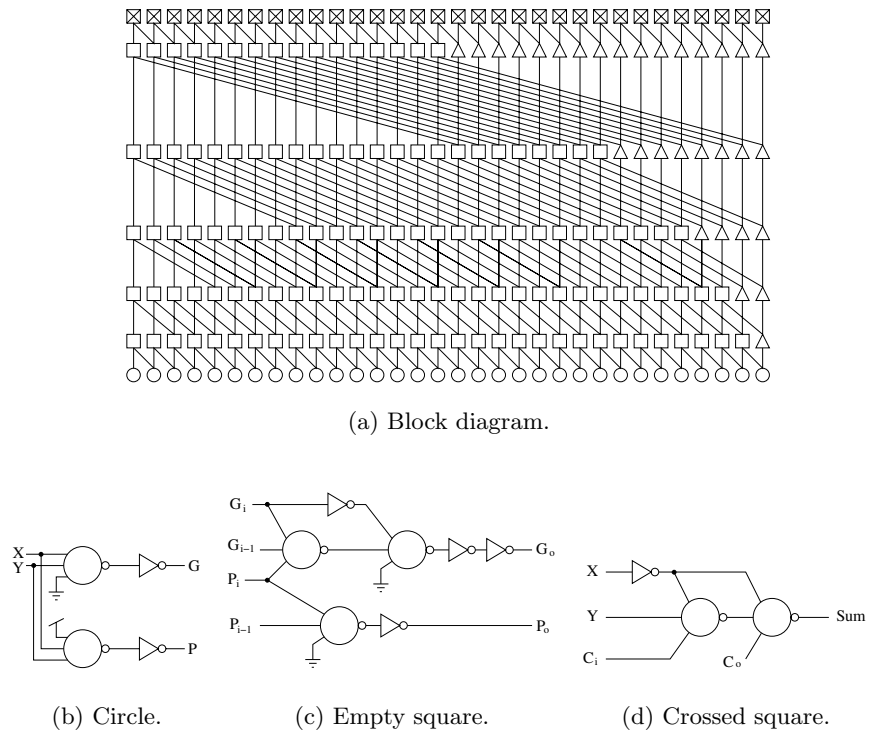


Fig. 2. Kogge-Stone adder

Simulations including wire delays show that these have a large impact on adder performance. In BPTM 100nm, the Kogge-Stone adder is 4.5 times faster than RCA, while with wire delays it is only 2.2–2.4 times faster. Similar tendencies hold true for 70nm (see Table 1).

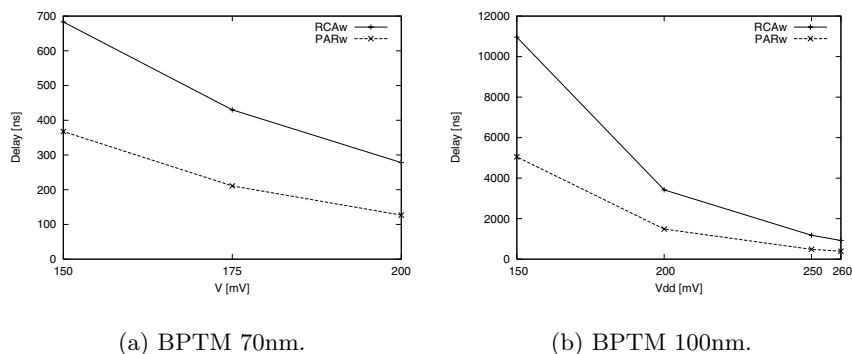
Figs. 3(a) and 3(b) reveal the correlation between V_{dd} and propagation delays. As V_{dd} is increased, power consumption is also increased. Hence, Fig. 4 shows the estimated energy for both the parallel adder and RCA. V_{dd} for the parallel adder is varied from 150mV to 250mV giving propagation delays spanning one order of magnitude: 5000ns to 500ns. V_{dd} for RCA was set such as to achieve the same performance. All our experiments show that this V_{dd} is between 110% and 120% of the V_{dd} of the parallel adder. For all these experiments, *the RCA was more energy efficient than the parallel adder when running at the same speed.*

When using the same V_{dd} in both 100nm and 70nm, we have also seen more than 10x reduction in delays for both adders. There is no clear indication that delays in subthreshold due to local wiring is scaled any differently from transistors across technology nodes.

All the results reported here are from simulations and not from chip measurements. This must be taken into account when interpreting the data, as errors

Table 1. Summary of simulation results. RCA = Ripple Carry Adder, PAR = Kogge-Stone Adder, “w” indicates that wire delays are included in the simulations

Circuit	V_{dd} [mV]	Node [nm]	t_{delay} [ns]	E [pJ]	L_{PMOS} [nm]	W_{PMOS} [nm]	L_{NMOS} [nm]	W_{NMOS} [nm]
RCA	260	100	767.9		120	690	120	120
PAR	260	100	168.8		120	690	120	120
RCAw	260	100	922.6		120	690	120	120
PARw	260	100	391.0		120	690	120	120
RCAw	250	100	1179		120	690	120	120
PARw	250	100	484.8		120	690	120	120
RCA	200	100	2847		120	690	120	120
PAR	200	100	627.6		120	690	120	120
RCAw	200	100	3421.7		120	690	120	120
PARw	200	100	1484.3		120	690	120	120
RCAw	150	100	10961		120	690	120	120
PARw	150	100	5051		120	690	120	120
<hr/>								
RCA	200	70	291		90	590	90	90
PAR	200	70	77		90	590	90	90
RCA	200	70	275.7		100	600	100	100
PAR	200	70	85		100	600	100	100
RCA	200	70	241.3		80	460	80	80
PAR	200	70	64.8		80	460	80	80
RCAw	200	70	278		80	460	80	80
PARw	200	70	127		80	460	80	80
RCAw	175	70	430		80	460	80	80
PARw	175	70	211		80	460	80	80
RCA	150	70	595.5		80	460	80	80
PAR	150	70	186.6		80	460	80	80
RCAw	150	70	683.0		80	460	80	80
PARw	150	70	367.6		80	460	80	80
<hr/>								
RCAw	183.8	100	5045.5	1.48	120	690	120	120
PARw	150	100	5051	2.26	120	690	120	120
RCAw	238.5	100	1479.9	2.48	120	690	120	120
PARw	200	100	1484.3	3.4	120	690	120	120
RCAw	292.5	100	488.2	5.0	120	690	120	120
PARw	250	100	484.8	6.57	120	690	120	120

**Fig. 3.** Impact of V_{dd} on delay

might come from the simulator, the Berkeley models, or from incorrect assumptions regarding transistor sizes and wire lengths.

The results have also shown to depend heavily on transistor sizing. Because of the difficulty of ensuring representative scaled sizing across technology nodes, comparisons across different technology nodes should be taken with care.

For FPGAs, wire delays are larger than for a custom design due to longer wires as well as switchbox resistance. Therefore a serial architecture may have

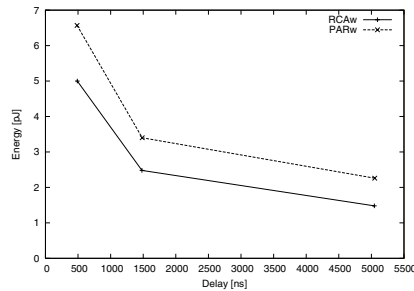


Fig. 4. Energy used by 32 bit adders (BPTM 70nm). V_{dd} for Kogge-Stone adder was 150mV, 200mV and 250mV. V_{dd} for RCA was 183mV, 238mV, 292mV for giving the same delay as the Kogge-Stone adder

an even larger advantage when compared to parallel in subthreshold FPGAs, and such techniques are already used in today's FPGAs.

4 Conclusions and Future Work

This paper has presented simulations in subthreshold that back the followings:

- migrating from 100nm to 70nm reduces the propagation delay by over 10x;
- full adders using threshold logic gates require very small energy;
- serial adders can match the speed of parallel ones in subthreshold when V_{dd} is increased only by 10-20%;
- at equal speeds in subthreshold, a serial adder has an advantage over parallel ones with respect to power consumption (and energy).

The more general conclusion is that, when operated in subthreshold serial solutions might do better than parallel ones, both for full custom arithmetics and FPGA based arithmetics.

Future work should concentrate on: optimising the implementations of the gates (e.g., combinations of gates), better characterisation (through fabrication and measurements), and the analysis of reliability enhanced adders using both low level techniques (e.g., high matching), as well as circuit level ones (based on redundancy).

References

1. ITRS: The International Technology Roadmap for Semiconductors. <http://public.itrs.net/> (2004)
2. Beiu, V., Rückert, U., Roy, S., Nyathi, J.: On nanoelectronic architectural challenges and solutions. In: Proc. IEEE Conference on Nanotechnology. (2004) 628–631

3. Burr, J.B., Shott, J.: A 200 mV self-testing encoder/decoder using stanford ultra-low-power CMOS. In: Proc. IEEE International Solid-State Circuits Conference. (1994) 84–85
4. Lande, T.S., Wisland, D.T., Sæther, T., Berg, Y.: FLOGIC-floating-gate logic for low-power operation. In: Proc. International Conference on Electronics, Circuits, and Systems. Volume 2. (1996) 1041–1044
5. Kim, C.H., Soeleman, H., Roy, K.: Ultra-low-power DLMS adaptive filter for hearing aid applications. *IEEE Trans. on VLSI Systems* **11** (2003) 1058–1067
6. Wentzloff, D.D., Calhoun, B.H., Min, R., Wang, A., Ickes, N., Chandrakasan, A.P.: Design considerations for next generation wireless power-aware microsensor nodes. In: Proc. International Conference on VLSI Design. (2004) 361–367
7. Ishibashi, K., Yamashita, T., Y, A., Minematsu, I., T.Fuji-moto: A 9uW 50MHz 32b adder using a self-adjusted forward body bias in SoCs. In: Proc. International Solid-State Circuits Conference. Volume 1. (2003) 116–119
8. Soeleman, H., Roy, K., Paul, B.: Robust subthreshold logic for ultra-low power operation. *IEEE Transactions on VLSI Systems* **9** (2001) 90–99
9. Paul, B.C., Raychowdhury, A., Roy, K.: Device optimization for digital subthreshold logic operation. *IEEE Transactions on Electron Devices* **52** (2005) 237–247
10. Calhoun, B.H., Wang, A., Chandrakasan, A.: Device sizing for minimum energy operation in subthreshold circuits. In: Proc. IEEE Custom Integrated Circuits Conference. (2004) 95–98
11. Beiu, V., Quintana, J., Avedillo, M.: VLSI impl. of threshold logic: A comprehensive survey. *IEEE Transactions on Neural Networks* **14** (2003) 1217–1243
12. Beiu, V.: Constructive threshold logic addition: A synopsis of the last decade. In: Proc. International Conference on Neural Networks. Number 2714 in LNCS (2003) 745–752
13. Celinski, P., Al-Sarawi, S., Abbott, D., Cotofana, S., Vassiliadis, S.: Logical effort based design exploration of 64-bit adders using a mixed dynamic-cmos/threshold-logic approach. In: Proc. Annual Symposium on VLSI. (2004) 127–132
14. Aunet, S., Berg, Y., Tjore, O., Næss, ., Sæther, T.: Four-MOSFET floating-gate UV-programmable elements for multifunction binary logic. In: Proc. Multiconference on Systemics. Volume 3. (2001) 141–144
15. Aunet, S., Oelmann, B., Abdalla, S., Berg, Y.: Reconfigurable subthreshold CMOS perceptron. In: Proc. IEEE International Joint Conference on Neural Networks. (2004) 1983–1988
16. Sulieman, M., Beiu, V.: Characterization of a 16-bit threshold logic single electron technology adder. In: Proc. International Symposium on Circuits and Systems. (2004) 681–684
17. Beiu, V.: A novel highly reliable low-power nano architecture: When von Neumann augments Kolmogorov. In: Proc. Application-specific Systems, Architectures and Processors. (2004) 167–178
18. Iwamura, H., Akazawa, M., Amemiya, Y.: Single-electron majority logic circuits. *IEICE Transactions on Electronics* **E18-C** (1998) 42–48
19. Kogge, P., Stone, H.: A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE Transactions on Computers* (1973) 786–793
20. Cao, Y., Sato, T., Orshansky, M., Sylvester, D., Hu, C.: New paradigm of predictive MOSFET and interconnect modeling for early circuit design. In: Proc. IEEE Custom Integrated Circuits Conference. (2000) 201–204 <http://www-device.eecs.berkeley.edu/~ptm>.