UNIVERSITY OF WESTMINSTER

WestminsterResearch
http://www.wmin.ac.uk/westminsterresearch

An *asynchrobatic*, radix-four, carry look-ahead adder.

David J. Willingham
Izzet Kale

School of Electronics and Computer Science

Copyright © [2008] IEEE. Reprinted from PRIME: 2008 PhD Research in Microelectronics and Electronics. Proceedings. Istanbul, Turkey, June 22–25, 2008. IEEE, pp. 105-108. ISBN 9781424419838.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Westminster's products or services. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.
Users are permitted to download and/or print one copy for non-commercial private study or research. Further distribution and any use of material from within this archive for profit-making enterprises or for commercial gain is strictly forbidden.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of the University of Westminster Eprints (http://www.wmin.ac.uk/westminsterresearch).

In case of abuse or copyright appearing without permission e-mail wattsn@wmin.ac.uk.

An *Asynchrobatic,* radix-four, carry look-ahead adder

David J. Willingham and Izzet Kale
Applied DSP and VLSI Research Group,
University of Westminster,
London, Great Britain.
D.Willingham@wmin.ac.uk & kalei@wmin.ac.uk

Abstract—**A low-power, *Asynchrobatic* (asynchronous, quasi-adiabatic), sixteen-bit, radix-four, parallel-prefix adder circuit is presented. The results show that it is an efficient, low power design, and that as would be expected with an asynchronous design, its performance is determined by its operating conditions. On a 0.35µm CMOS process, under "typical" process conditions, operating at an effective frequency of 22MHz, an addition can be performed using 69pW, with 48.3pW used by the control logic and 20.7pW by the data-path.**

I. Introduction

Asynchrobatic logic is a low-power design methodology that combines an asynchronous stepwise charging controller with a quasi-adiabatic data-path. In the authors' previous work [1], it has been shown that it is possible to implement simple inverter or buffer chains using this design methodology. This work extends that initial presentation and demonstrates that more complex data-path structures can be implemented using this novel low-power technology. To that end, this paper presents the design and simulation evaluation of a sixteen-bit, radix-four, carry look-ahead adder. In section the background of *Asynchrobatic* logic is presented. Sections III concentrates on the design and testing of the adder. The results are presented in Section IV.

II. Asynchrobatic Logic

Asynchrobatic logic uses an asynchronous Step-Wise Charging (SWC) controller to drive what are in effect the local power-clock signals of dual-rail adiabatic logic families including Efficient Charge Recovery Logic (ECRL) [2] (also known as 2n-2p logic), 2n-2n2p logic [3] or Positive Feedback Adiabatic Logic (PFAL) [4]. This allows a data-path constructed of *Asynchrobatic* processing pipelines to be created. The asynchronous controller uses a Muller C--element [5] to drive a generator which creates a series of pulses. The duration of the pulses is controlled by N- and P-bias voltages, which are used to control a series of current-starved invertors. These pulses are routed to a SWC circuit [6] which progressively connects the local power-clock signals from V_{ss} to V_{dd} via a series of tank capacitors. Once the local power-clock is connected to V_{dd}, the handshake signals can be sent to the previous and subsequent stages. Once the next stage has completed its processing, the order of the pulses is reversed to recover the charge to the tank capacitors. In the *Asynchrobatic* design style, the use of four-phase asynchronous signaling perfectly complements the four charging and discharging phases of the previously mentioned adiabatic logic families. Fig. 1 shows an ECRL buffer, Fig. 2 a 2n-2n2p buffer and Fig. 3 a PFAL buffer. These could be converted to inverter configurations by simply swapping the A_H and A_L labels. Fig. 4 shows the asynchronous Muller C-element controller and Fig. 5 shows the SWC circuit.

III. Adder Design

The adder style chosen for this demonstration was the parallel prefix structure [7]. However, because of the nature of the *Asynchrobatic* pipeline, it was decided to use a radix-four structure rather than the more common radix-two structure, as this reduces the number of stages in the *Asynchrobatic* pipeline, thus making the design more efficient. For this demonstration circuit, a Skylansky adder [8] was used. For adders larger than 16-bits wide, it is likely that fan-out will become a problem, if a Skylansky adder is used. However, due to the dual-rail nature of *Asynchrobatic* logic, the amount of wiring would become problematic if the Kogge-Stone structure was used. Therefore for wider adders, it is suggested that a novel, higher-radix extension of Knowles adders [9] is used. The use of Higher-Radix Knowles Adders (HRKA) would allow a designer to trade-off the capacitive load from the fan-out against the wiring flux, which due to the dual-rail nature of the design is something that could become problematic in wider designs.

The radix-four adder consists of an input stage of half-adders which create the Generate and Propagate signals, two stages of Look-ahead logic, and a final output stage of exclusive-OR gates. The higher-radix structure has been previously suggested for both Kogge-Stone adders [10] and Skylansky adders [8]. Compared to a radix-two version, which would require six *Asynchrobatic* pipeline stages, this adder uses only four. This trade-off uses a more complex logical implementation that requires more inter-stage wiring, but should be both faster and more power efficient because there are less controller stages which consume most of the power used in this design style. To fully exploit the potential gains of this approach, a very wide data-path widths with

This research was partially funded by a Quintin Hogg Research Scholarship from the University of Westminster.

Authorized licensed use limited to: University of Westminster. Downloaded on March 12,2010 at 05:36:09 EST from IEEE Xplore. Restrictions apply.

complex pipeline stages will need to be deployed in designs undertaken using this design style. Whilst even fewer pipeline stages could be used by increasing the radix further. This was not done in order maintain circuit reliability by keeping the number of series nFETs to four or less.

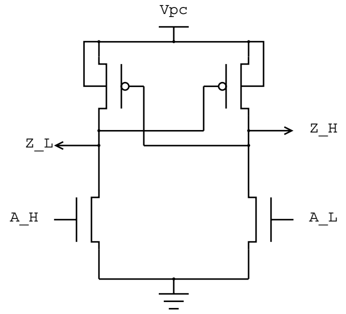

Figure 1. An ECRL buffer [2].

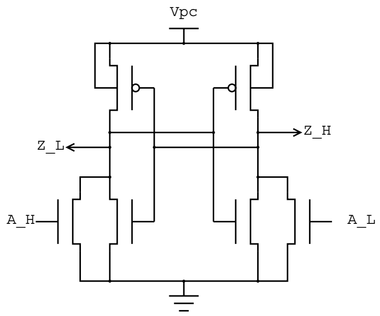

Figure 2. A 2n-2n2p buffer [3].

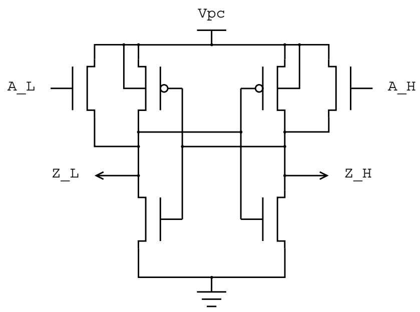

Figure 3. A PFAL buffer [4].

A. Adder cells

This adder structure uses the following data-path cells: buffer, two-input XOR, two-, three-, and four-input AND, and two-, four-, and six-level AND-OR type structures. The construction of the evaluation structures of the three most complex gates {two-input XOR, four-input AND and six-level AND-OR) are shown in Fig. 6, Fig. 7 & Fig. 8. From these, the design of the other gates can be easily derived. These cells were implemented using the PFAL design style.

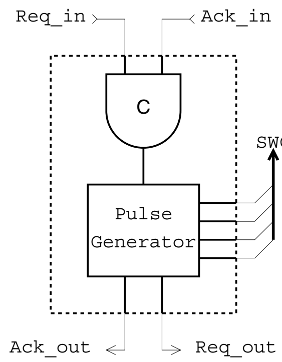

Figure 4. An asynchronous controller & pulse generator [1].

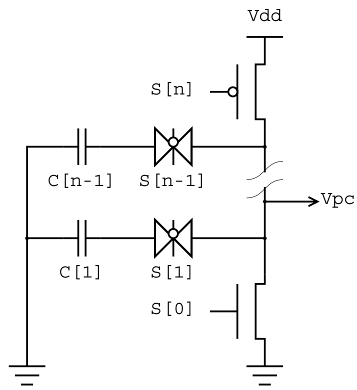

Figure 5. A SWC circuit [14].

They are combined to form the half-adder, constructed from a two-input AND and a two-input XOR, and the parallel-prefix Propagate/Generate Logic circuits. A fourth-order Propagate/Generate circuit (PG4) is constructed from an AND4 and a six-level AND-OR, whilst a first-order version (PG1) is simply a pair of buffers. Due to the dual-rail nature of these cells, this relatively small demonstration circuit shows that the majority of common combinational data-path functions are viable. However, based upon the previous caveat of no more than four series nFETs, it can be seen that not only is every possible logic function of four or less inputs viable, but that other potentially useful logic functions like multi-stage AND-OR and eight-way MUX can be implemented. Furthermore, due to the dual-rail nature of this logic style, a complete four-input library can be implemented with relatively small number of cells. With only 222 different cells required to implement every one of the 65,536 functions (including degenerate functions with one or more static inputs) of four inputs.

With the exception of the Exclusive-OR gate, which was designed using a Reduced Ordered Binary Decision Diagram (ROBDD) [12] method, these cells were designed using the Quine-McClusky [13] method. This allowed the six-level AND-OR structures which have seven inputs to be implemented with no more than four series nFETs.

Authorized licensed use limited to: University of Westminster. Downloaded on March 12,2010 at 05:36:09 EST from IEEE Xplore. Restrictions apply.

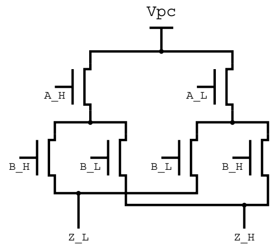

Figure 6. nFET tree for two-input XOR.

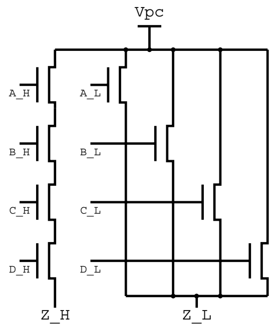

Figure 7. nFET tree for four-input AND.

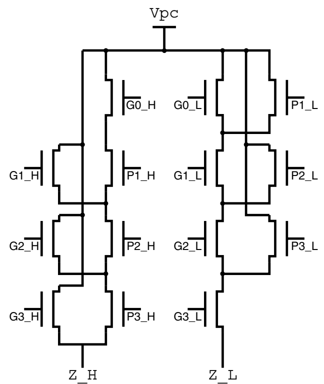

Figure 8. nFET tree for four-level AND-OR.

The control logic is constructed using the asynchronous SWC circuit detailed above, and implemented using three tank capacitors each having a capacitance of 10pF. The choice of this value is a trade-off between the stability of the tank-capacitor voltage verses the time taken to supply the initial charge, and was arrived at by simulation studies. Furthermore, this can be achieved with on-chip capacitors in today's CMOS processes. For simplicity in this example the Carry input has been tied to zero and it has been assumed that validity of both the main adder inputs is represented by a single handshake signal, but in a more complex system, an asynchronous *join*-function could be implemented if each input had its own handshake signal. This could easily be done by using the appropriate multi-input C-element within the control logic. The high-level structure of the complete adder is shown in Fig. 9, the boxes labeled "HA" represent half adder circuits, the boxes labeled "X" represent XOR gates, and the boxes labeled with numbers represent the Generate/Propagate logic of that order.

B. Modeling the adder

The adder was initially described using Verilog to check that it was functionally correct, and then modeled using SPICE to allow functional circuit-level simulation. The use of Verilog models allow both a high level model and a cell accurate model to be created; the model could also be extended to switch-level modeling which would allow fully accurate, dual-rail models to be created. The cell accurate model implements the individual quasi-adiabatic cells as a rising-edge triggered flop with logic-processing inputs. This can be extended to incorporate a reset action on the outputs triggered by the negative edge of the local-power clock. The incorporation of the reset action adds an extra beneficial cross-check.

The SPICE implementation used Alcatel (AMIS) 0.35µm models. The current simulations were performed using pre-layout netlists, and do not include any parasitic elements.

C. Testing the adder

The adder was tested by driving it with vectors generated using two differently-seeded Linear Feedback Shift Registers (LFSR), one to drive each of the adder's inputs. This ensured that identical data-streams were presented to each adder input in all simulation runs, irrespective of the operating conditions of the circuit under test. The control logic was connected so that the adder would run freely at a speed determined by the Process, Voltage and Temperature (PVT) conditions. The adder was tested at nominal voltage (3.3V) in the fast (ff, -40°C), typical (tt, 25°C) and slow (ss, 125°C) corners, in four skew corners (sf or fs, -40°C or 125°C) and at typical with different levels of bias applied to the delay circuits in the controller's pulse generator.

IV. Results

The power and performance figures were obtained from the netlist-only fast-SPICE (Mentor Graphic's Eldo Mach) simulations, and were calculated according to (1).

$$P = \frac{V}{(T_1 - T_0)} \int_{T_0}^{T_1} I \, dt \qquad (1)$$

It can be clearly seen that the effective operational frequency is dependent upon both the PVT conditions and the control voltage applied to the delay elements. This confirms that the design is operating asynchronously. It can also be seen that the tank capacitors converge to an operating voltage, which again is dependent upon the PVT conditions and the control voltage, but also shows minor data-dependency. Under typical PVT conditions (tt, 3.3V, 25°C), the power consumption of a single cycle of a single SWC is 12.1pW and the power used in the adder circuit is 20.7pW. Although a full range of process conditions were analysed,

Authorized licensed use limited to: University of Westminster. Downloaded on March 12,2010 at 05:36:09 EST from IEEE Xplore. Restrictions apply.

the results presented in Table I keeps the voltages fixed at nominal value of 3.3V. This is to keep the bias voltages identical in all cases. Results are presented for fast, slow, typical and skew corners. Table II shows the effect of varying the bias voltage.

TABLE I. PERFORMANCE OVER PVT CONDITIONS.

Corner †	**Effective Frequency (Hz)**	**Controller Power (W)**	**SWC Circuit Power (W)**
{tt, 25°C}	2.20×10^{7}	4.83×10^{-11}	2.07×10^{-11}
{ff, -40°C}	4.99×10^{7}	4.55×10^{-11}	2.16×10^{-11}
{fs, -40°C}	2.11×10^{7}	5.37×10^{-11}	2.61×10^{-11}
{fs, 125°C}	2.38×10^{7}	4.77×10^{-11}	2.34×10^{-11}
{sf, -40°C}	1.96×10^{7}	4.46×10^{-11}	2.65×10^{-11}
{sf, 125°C}	2.09×10^{7}	4.22×10^{-11}	2.46×10^{-11}
{ss, 125°C}	9.96×10^{6}	4.61×10^{-11}	2.14×10^{-11}

† V_{dd}=3.3V V_{bias}=900mV

TABLE II. PERFORMANCE WHEN VARYING V_{BIAS}.

V_{bias} (V) ‡	**Effective Frequency (Hz)**	**Controller Power (W)**	**SWC Circuit Power (W)**
0.850	1.51×10^{7}	5.19×10^{-11}	1.96×10^{-11}
0.900	2.11×10^{7}	4.88×10^{-11}	2.02×10^{-11}
0.950	2.74×10^{7}	4.63×10^{-11}	2.13×10^{-11}

‡ PVT {tt, 3.3V, 25°C}

V. CONCLUSIONS

It has previously been shown that the *Asynchrobatic* logic style can be used to implement simple data-path structures like inverter and buffer chains. This opus extends the work described in that paper and demonstrates that within necessary process-related design constraints, arbitrarily complex logic functions can be implemented using *Asynchrobatic* logic. It also suggests a method for creating wider higher-radix adders by extending Knowles Adders to higher radices, allowing the designer to find an appropriate trade-off between wiring flux and fan-out.

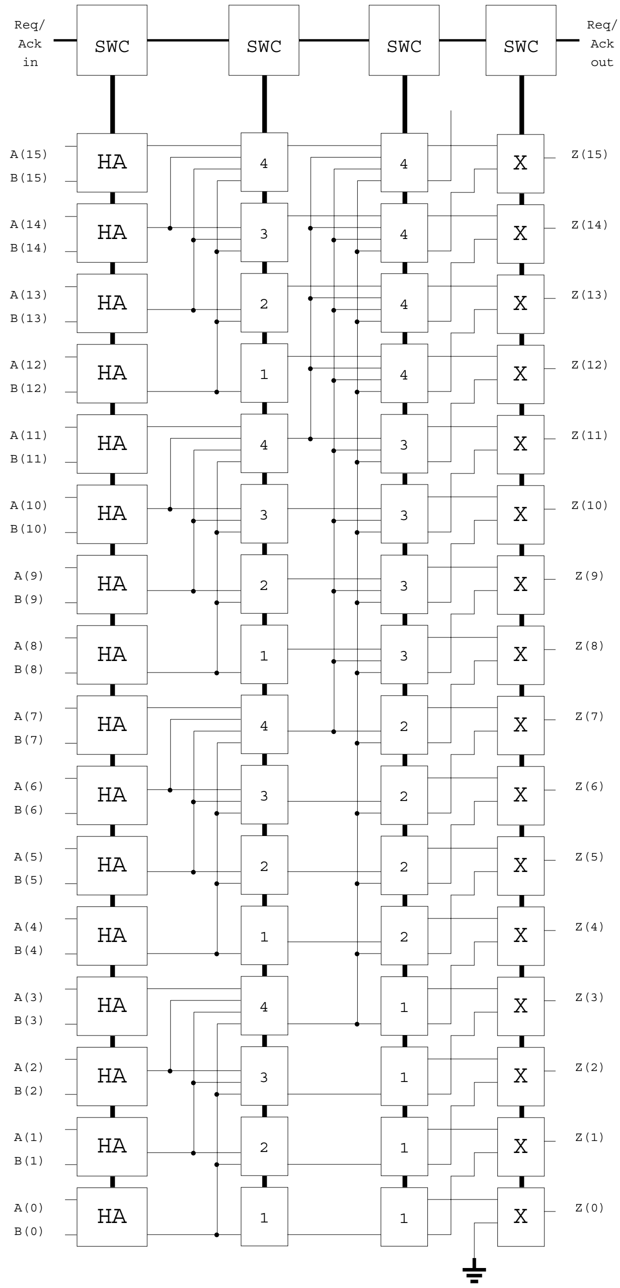

Figure 9. Top-level structure of the adder.

REFERENCES

[1] D.J. Willingham and I. Kale, "Asynchronous, quasi-adiabatic (*Asynchrobatic*) logic for low-power very wide data width applications", Proc. ISCAS 2004, vol. 2, pp. 257-260.

[2] Y. Moon and D. Jeong, "An efficient charge recovery logic", IEEE J-SSC, vol. 31(4), pp. 514-522, 1996.

[3] J.S. Denker, "A review of adiabatic computing", Proc. ISLPE 1994, pp. 94-97.

[4] A. Vetuli, S.D. Pascoli and L.M. Reyneri, "Positive feedback in adiabatic logic", Elec. Lett., vol. 32(20), pp. 1867-1869, 1996.

[5] D.E. Muller and W.S. Bartky, "A theory of asynchronous circuits", Proc. Int. Symp. Theory of Switching, Havard University Press, 1959.

[6] L.J. Svensson and J.G. Koller, "Driving a capacitive load without dissipating fCV^2", Proc. ISPLE 1994, pp. 100-101.

[7] R.E. Ladner and M.J. Fischer, "Parallel prefix computation", J-ACM, vol. 27, no.4, pp. 831-833, 1980.

[8] J. Skylansky, "Conditional sum addition logic", IRE T. Elec. Comp., vol. 9(6), pp. 226-231, June 1960.

[9] S. Knowles, "A family of sdders", Proc. Symp. Comp. Arith. 1999, pp. 30-34.

[10] F.K. Gurkaynak, Y. Leblebicit, L. Chaouati and P.J. McGuinness, "Higher radix Kogge-Stone parallel prefix adder architectures", Proc. ISCAS 2000, vol. 5, pp. 609-612.

[11] D.J. Willingham, "Adiabatic CMOS 8x8 multiplier", MSc report, University of Westminster, 1999.

[12] R.E. Bryant, "Graph-based algorithms for boolean function manipulation", IEEE Trans. Comp., vol. 35(8), pp. 677-691, 1986.

[13] K.M. Chu and D.L. Pulfrey, "Design procedures for differential cascode voltage switch logic circuits", IEEE J-SSC, vol. 21(6), pp. 1082-1087, 1986.

Authorized licensed use limited to: University of Westminster. Downloaded on March 12,2010 at 05:36:09 EST from IEEE Xplore. Restrictions apply.