Using positive feedback adiabatic logic to implement reversible Toffoli gates.

David J. Willingham
Izzet Kale

School of Electronics and Computer Science

Copyright © [2008] IEEE. Reprinted from Ellervee, Peeter, Jervan, Gert and Nielsen, Ivan Ring, (eds.) 26th Norchip Conference, Tallinn, Estonia, 17 - 18 November 2008. Formal proceedings. IEEE, pp. 5-8. ISBN 9781424424924.

Using Positive Feedback Adiabatic Logic to implement Reversible Toffoli Gates

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

Abstract*—A Reversible, Positive Feedback Adiabatic Logic Circuit is presented, which by implementing the universal Toffoli Gate demonstrates that reversible logic circuits can be created and implemented using this adiabatic logic family. When compared to circuits with similar circuit structures that do not incorporate complete recovery logic, the use of reversible structures shows a reduction in energy losses by a mean of just under 63%.

Index Terms—**Adiabatic logic, charge-recovery logic, low-power circuit techniques, reversible computation.**

I. Introduction

A significant proportion of recent research into charge recovery logic has focused upon implementing or using irreversible, quasi-adiabatic logic families. These various quasi-adiabatic logic families are already capable of operating with power dissipations that are substantially less than the well-known result for static CMOS logic (1). Instead, they operate with power dissipations related to the threshold voltage of the devices used to implement the circuit, rather than the supply voltage. However, as will be shown in this work, by creating a complete recovery path and through the use of reversible logic, it is possible to reduce this power consumption further still.

$$P = f\,C_L\,V_{DD}^2 \qquad (1)$$

Where the power (P) is proportional to the operating frequency (f), the capacitive load (C_L) and the square of the Voltage (V_{DD}).

This paper presents the results obtained from comparisons of simulations of a single Toffoli Gate circuit embedded in the middle of a buffer pipeline. An implementation in the more common quasi-adiabatic logic style was compared against an identical implementation, modified by the inclusion of complete recovery paths, to produce a fully reversible adiabatic logic family.

II. Adiabatic and Quasi-Adiabatic Logic

In the last decade of the twentieth century, a substantial number of adiabatic and quasi-adiabatic logic families were proposed. Some of the first of these families used diodes [1] and had inherent non-adiabatic losses, these will not be considered further. Early diode-less quasi-adiabatic logic families like Efficient Charge Recovery Logic (ECRL) [2] (also known as 2n-2p [3]) and 2n-2n2p [3] achieved recovery of the power-supply clock down to the threshold voltage of their pMOS devices. Further proposals like Energy Efficient Logic (EEL) [4] extended this concept by allowing full recovery through the use of extra nMOS devices, but unfortunately this necessitated the use of additional external pulse generating logic. Finally, with creation of Positive Feedback Adiabatic Logic (PFAL) [5], a viable reversible adiabatic logic family was introduced. However, further publications of designs using this family [6][7][8][9][10] do not appear to have considered or explored the full potential of this reversibility. Another work that also suggested the incorporation of a complete recovery was the proposal of the Efficient Adiabatic Charge-Recovery Logic (EACRL) [11]. In the description of this logic style, it was suggested that recovery could only be achieved if the subsequent logic stage was dependent only upon the current gate, which is now shown to be an excessive restriction.

The above families, and several others not explicitly mentioned in this paper are all loosely based upon Differential Cascode Voltage Switch Logic (DCVSL) [12]. However, there are other styles of adiabatic logic that are not based upon DCVSL. Another style of adiabatic, reversible logic has been previously demonstrated [13]. However, this was based upon Reversible Energy Recovery Logic (RERL) [14], and as such, requires six- or eight-phase clocking. The proposed style in this paper utilizes only four-phase clocking, which makes the energy recovery structures easier to implement, and can also means such systems could be viably implemented using *Asynchrobatic* logic [15][16] techniques.

1-4244-2493-1/08/$20.00 ©2008 IEEE

III. Positive Feedback Adiabatic Logic

Like many adiabatic logic families, PFAL is a dual-rail logic family based upon a pair of cross-coupled inverters that are supplied using a power-clock, rather than a static DC power-supply. The configuration of the evaluation logic is what makes PFAL an ideal family to implement fully reversible logic. This logic is constructed from nMOS devices attached between the power-clock and the outputs. These nMOS devices take complementary inputs and are constructed to produce a low-resistance path between the power-clock and the asserted output. The non-asserted output should be left with a high-impedance path to power-clock, and will be pulled low by the cross-coupled n-type devices. This means that the function is evaluated when there is sufficient differential between the two outputs, but far more importantly means that by using reverse-flowing data, the outputs can be more completely recovered. This should allow losses to be reduced to leakage. The design of logic functions for all dual-rail adiabatic logic families can be achieved by using the same procedures that may be used for DCVSL. These methods are either a Quine-McClusky style [17] or one based upon Ordered Binary Decision Diagrams (OBDD) [18][19], or extensions thereto. A reversible buffer implemented using PFAL is shown in Fig. 1. During the preceding phase of the power-clock, one of the complementary inputs, "A_L" or "A_H" is asserted. The power-clock V_{PC} is then ramped up, causing the function to evaluate, and be presented on outputs "Z_L" and "Z_H". The recovery path is then evaluated on "P_L" and "P_H", and complete recovery can occur through these devices as the power-clock is ramped back down. It can be seen that without the recovery path facilitated by "P_L" and "P_H", that recovery would only be possible down to the threshold voltage of the cross-coupled pMOS devices. This is because when the outputs of the current stage go into recovery, the drive onto the forward inputs will have already been recovered, resulting in both paths being in a high-impedance state.

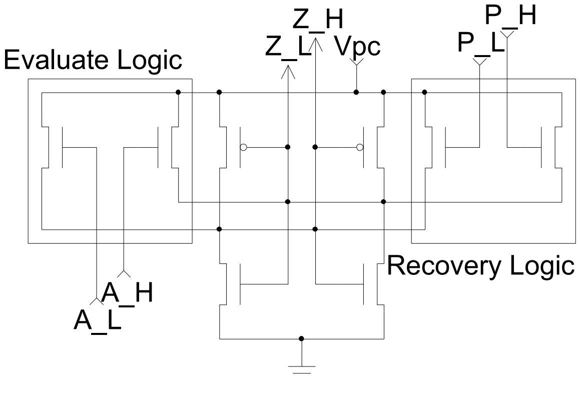

Figure 1. Reversible buffer implemented using PFAL.

IV. Reversible Computation

Following considerable work after the proposal of "Maxwell's Dæmon" [20]. It has been shown that energy dissipation during computation is caused not by the act of processing information, but by the loss of information when it is erased [21]. The consequence of this is that in an ideal situation, if information is processed in a reversible fashion and arbitrarily slowly, then this processing can occur using as little energy as is required. In reality, there will be some loses, which in this case will be due to electrical resistance.

For logic to be reversible, information must not be erased. This means that common functions like "AND" and "OR" cannot be directly implemented because with multiple inputs being reduced to a single output, there is clearly a loss of information. The common logic function with the most potential for reversibility is "XOR", although "XNOR" would be equally good. The relationship shown in (2) means that by preserving input A (or input B) as well as the result of A XOR B, a fully reversible system of gates can be created. This is known as the Feynman Gate and also as a Controlled-NOT (CN) gate.

$$P = A,\ Q = A \oplus B;\ A = P,\ B = P \oplus Q; \tag{2}$$

Unfortunately, the Feynman Gate is not a universal gate because on its own it cannot be used to create every possible logic function. However, the Toffoli Gate [22], which is its three-input variant, and is also known as the Controlled-Controlled-NOT (CCN), is a universal gate that can be used to implement any reversible function. The relationship shown in (3) details the operation of a Toffoli Gate.

$$P=A,\ Q=B,\ R=(A\bullet B)\oplus C;\ A=P,\ B=Q,\ C=(P\bullet Q)\oplus R; \tag{3}$$

There are various other reversible gates [23], which can be found by performing an appropriate literature search, but since the three-input Toffoli Gate detailed above is universal, these other gates will not be considered within the scope of this paper.

Using the design methodologies mentioned previously in Section III, it is possible to convert the functions described above in (3) into nMOS trees for implementation. The identical functional descriptions of inputs A, B, P and Q simply result in buffers. However, for inputs C and R, which are also identical, the result is an AND-XOR gate. Fig. 2 shows the decision tree used to implement this AND-XOR function.

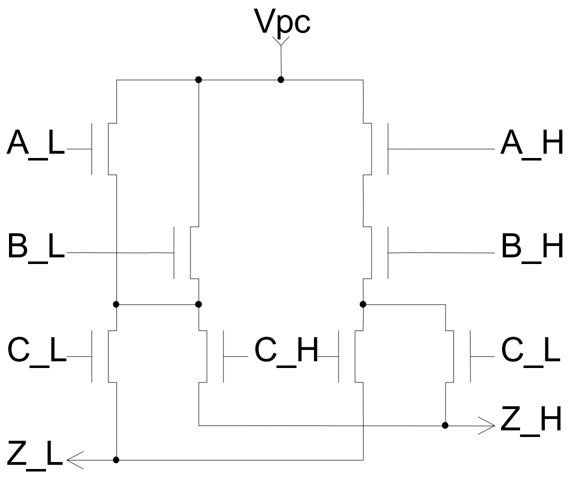

Figure 2. Decision Tree to implementing the AND-XOR function of a Toffoli Gate [22].

V. Results

The Toffoli Gate circuit was implemented as a front-end SPICE netlist using AMIS 0.35µm technology. This was done for both reversible and irreversible PFAL cells, and also for static CMOS. To allow a meaningful comparison, the irreversible design was created by removing the recovery paths from the devices. The design was tested by exercising it through two complete transition sequences. The chosen sequence exercises every possible transition of a three-bit value to another three-bit value, including degenerate cases where no transition occurs. This sequence requires a total of sixty-four transitions. The figures presented represent the arithmetic mean of the current supplied (over the sixty-four different possible transitions) to each pipeline stage leading to and from the Toffoli gate. These circuits were simulated under typical Process, Voltage and Temperature (PVT) conditions {tt, 3.3V, 25°C} using Mentor Graphic's Eldo simulator, with simulations tuned by including the ".OPTION TUNING=ACCURATE" card. The results were automatically calculated using ".EXTRACT" cards to measure the mean current drawn according to (4). This was performed on the second run-through of the transition sequence to eliminate measurement errors due to start-up.

$$I_{Mean} = \frac{1}{64}\int_{T_0}^{T_1} \mathrm{I}dt \qquad (4)$$

The simulations used eleven stages, each driven by an ideal voltage-source element. The Toffoli function is evaluated at stage six (highlighted in bold in Tables I & II). However, for the reversible system, the recovery path will need to compute this function based on the outputs of stage six to recover stage five (highlighted using italics in Tables I & II). Therefore, it may be useful to consider reversible gates as existing in between two adjacent power-clock signals. The final stage of the reversible system has its recovery inputs permanently tied to an inactive state. This accounts for its poor performance.

TABLE I. Mean current drawn in both reversible and irreversible PFAL Designs

Stage	**Mean Current (A)**			**Saving (%)**
	Reversible	***Irreversible***	***Saved***	
1	1.2917×10^{-15}	4.0169×10^{-15}	2.7252×10^{-15}	67.8
2	1.0621×10^{-15}	4.0478×10^{-15}	2.9857×10^{-15}	73.8
3	1.2866×10^{-15}	4.1259×10^{-15}	2.8393×10^{-15}	68.8
4	1.4049×10^{-15}	4.2877×10^{-15}	2.8828×10^{-15}	67.2
5	*3.1372×10^{-15}*	*4.6673×10^{-15}*	*1.5301×10^{-15}*	*32.8*
6	**6.1793×10^{-16}**	**5.4671×10^{-15}**	**4.8492×10^{-15}**	**88.7**
7	1.6027×10^{-15}	4.3141×10^{-15}	2.7114×10^{-15}	62.8
8	1.3997×10^{-15}	4.2985×10^{-15}	2.8988×10^{-15}	67.4
9	1.0849×10^{-15}	3.9443×10^{-15}	2.8594×10^{-15}	72.5
10	1.9549×10^{-15}	4.0166×10^{-15}	2.0617×10^{-15}	51.3
11	5.5129×10^{-15}	4.2376×10^{-15}	-1.2753×10^{-15}	-30.1

TABLE II. Mean current drawn in static CMOS and Reversible PFAL Designs

Stage	**Mean Current (A)**			**Saving (%)**
	Reversible	***Static CMOS***	***Saved***	
1	1.2917×10^{-15}	4.0186×10^{-14}	3.8894×10^{-14}	96.8
2	1.0621×10^{-15}	2.2993×10^{-14}	2.1931×10^{-14}	95.4
3	1.2866×10^{-15}	2.2875×10^{-14}	2.1588×10^{-14}	94.4
4	1.4049×10^{-15}	2.2847×10^{-14}	2.1442×10^{-14}	93.9
5	*3.1372×10^{-15}*	*3.0000×10^{-14}*	*2.6863×10^{-14}*	*89.5*
6	**6.1793×10^{-15}**	**4.0447×10^{-14}**	**3.9829×10^{-14}**	**98.5**
7	1.6027×10^{-15}	2.5807×10^{-14}	2.4204×10^{-14}	93.8
8	1.3997×10^{-15}	2.2881×10^{-14}	2.1481×10^{-14}	93.9
9	1.0849×10^{-15}	2.2868×10^{-14}	2.1783×10^{-14}	95.3
10	1.9549×10^{-15}	2.2845×10^{-14}	2.0890×10^{-14}	91.4
11	5.5129×10^{-15}	1.7876×10^{-14}	1.2363×10^{-14}	69.2

It can be seen from the data presented above that, on average, the reversible PFAL Toffoli Gate draws 11.3% of the current drawn by the irreversible PFAL AND-XOR gate, and that the reversible gate draws just 1.5% of the current that would be drawn by a static CMOS AND-XOR gate. The pipeline stage preceding the Toffoli Gate has the worst relative performance, but it must be remembered that for the reversible pipeline, this stage contains the complex recovery tree. If the combined values from the fifth and sixth stages are compared, then the combined saving is 62.9%.

VI. Conclusions

In this paper, it has been shown that PFAL has the potential to be used to implement arbitrary reversible logic functions. It has also been shown that by making PFAL fully reversible, considerably reduced power consumption can be obtained. The potential to create reversible pipelines using *Asynchrobatic* logic has been alluded to and has clear potential for further work.

ACKNOWLEDGMENT

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

REFERENCES

[1] Dickinson A.G. & Denker J.S., "Adiabatic Dynamic Logic", Proc. CICC 1994, pp. 282-285.

[2] Moon Y. & Jeong D-K., "An Efficient Charge Recovery Logic Circuit", IEEE J.SSC, 31(4):514-522, 1996.

[3] Kramer A., Denker J.S., Flower B. & Moroney J., "2nd Order Adiabatic Computation with 2N-2P and 2N-2N2P Logic Circuits", Proc. ISLPED 1995, pp. 191-196.

[4] Yeh, C.C., Lou J.H. & Kuo J.B., "1.5V CMOS full-swing energy efficient logic (EEL) circuit suitable for low-voltage and low-power VLSI applications", Elec. Lett., 33(16):1375-1376, 31st July 1997.

[5] Vetuli A., Pascoli S.D. & Reyneri L.M., "Positive Feedback in Adiabatic Logic", Elec. Lett., 32(20):1867-1869, 26th Sept. 1996.

[6] Amirante E., Fischer J. & Schmitt-Landsiedel D., "Verlustleistungsschwankungen in adiabatischen Schaltungen", ITG-Fachbericht 162 "Mikroelektronik für die Informations-technik", Darmstadt, Germany, 20-21 Nov. 2000, pp. 133-138, (Title translates to: "Variations of the Power Dissipation in Adiabatic Circuits").

[7] Amirante E., "Adiabatic Logic in Sub-Quartermcron CMOS Technologies", Shaker Verlag, 2004, ISBN 3-8322-3487-X.

[8] Blotti A. & Saletti R., "Ultra low-power adiabatic circuit semi-custom design", IEEE T. VLSI, 12(11):1248-1253, 2004.

[9] Fischer J., Teichmann P. & Schmitt-Landsiedel D., "Scaling trends in adiabatic logic", Proc. 2nd Conf. on Computing Frontiers, pp. 427-434, Italy, 2005.

[10] Teichmann P., Fischer J., Chouard F.R., & Schmitt-Landsiedel D., "Design of Ultra-Low-Power Arithmetic Structures in Adiabatic Logic", ISIC 2007, pp. 365-368.

[11] Varga L., Kovács F. & Hosszú G., "An Efficient Adiabatic Charge-Recovery Logic", Proc. IEEE SouthEastCon 2001, pp.17-20.

[12] Heller L., Griffin W., Davis J. & Thoma, N., "Cascode voltage switch logic: A differential CMOS logic family", ISSCC Dig. Tech. Papers, 1984, pp. 16-17.

[13] Khazamipour A. & Radecka K., "Adiabatic Implementation of Reversible Logic", MWSCAS 2005, pp. 291-294.

[14] Athas W.C., Svensson L.J., Koller J.G. & Chou E., "Low Power Digital Systemsbased on Adiabatic-Switching Principles", IEEE T. VLSI 2(4):398-407, 1994.

[15] Willingham D.J. & Kale I., "Asynchronous, quasi-Adiabatic (*Asynchrobatic*) Logic for Low-Power very Wide Data Width Applications", Proc. ISCAS 2004, vol. II, pp. 257-260.

[16] Willingham D.J. & Kale I., "An *Asynchrobatic*, radix-four, carry look-ahead adder", Proc. PRIME 2008, pp. 105-108.

[17] Chu K.M. & Pulfrey D.L., "Design Procedures for Differential Cascode Voltage Switch Logic Circuits", IEEE J-SSC, 21(6):1082-1087, 1986.

[18] Karoubalis T., Alexiou G.Ph. and Kanopoulos N., "Optimal synthesis of differential cascode voltage switch (DCVS) logic circuits using ordered binary decision diagrams (OBDDs)", Proc Euro-DAC 1995, pp. 282-287.

[19] Bryant R.E., "Graph-Based Algorithms for Boolean Function Manipulation", IEEE T. Comp, 5(8):667-691, 1986.

[20] Leff H.S. & Rex A.F. (eds.), "Maxwell's Demon: Entropy, Information, Computing", IoP Publishing, 1990, ISBN 0750300574.

[21] Bennett C.H., "Logical Reversibility of Computation", IBM J. R&D, 17(6):525-532, 1973.

[22] Feynman R.P.; Hey T. & Allen R.W. (eds.) "Feynman Lectures on Computation", Westview Press, 2000, p. 190, ISBN 0738202967.

[23] Fredkin E. & Toffoli T., "Conservative Logic", Int. J. Theoret. Phys., 21(3/4):219-253, 1982.