

Design and Implementation of Low Power Efficient 8-bit Carry Look Ahead Adder using Adiabatic Technique

Sneha¹, Dwijesh B², Snehal Magadam³, K Sagar⁴, Hari Krishna Moorthy⁵

^{1, 2, 3, 4}UG-Final Year B.Tech Students, ⁵Associate professor, Department of Electronics and Communication Engineering, School of Engineering and Technology, Jain(Deemed-to-be University), Bangalore, Karnataka, India

Abstract: For high speed and portable equipment, energy efficiency is the most important factor. By adopting different design styles, the power consumption of the electronic devices can be reduced. In this paper by applying power minimization techniques at circuit levels the power consumption of any combinational logic circuits can be reduced. Adiabatic logic style is said to be an attractive solution for such low power electronic applications. Adiabatic describe the thermo dynamic process in which there is no energy exchange with the environment, and therefore very less dissipated energy loss. With this technique, dissipation in pull up network can be minimized and some energy stored at load capacitance can be recycled instead of heat dissipation. In our proposed design the efficiency of a fully adiabatic logic circuit for 8-bit carry look ahead adder will be implemented and power dissipation of adiabatic and conventional circuits is compared. Using CADENCE simulation, the parameters like Transistor gate sizing, maximum frequency of operation, the minimum voltage of operation are used to measure the performance of each circuit. We are using ECRL (Efficient Charge Recovery Logic) logic families which is compared with conventional cmos circuit.
Keywords: Adiabatic logic, 8-bit carry look ahead adder, cadence, ECRL, conventional cmos.

I. INTRODUCTION

With the development of VLSI technology power dissipation is increasing dramatically, low power devices are the essentiality of today. The world needs low power devices with faster speed. In this paper a type of adiabatic logic family ECRL is used. The word adiabatic tells that, thermodynamic process in which no energy exchange with outer environment which leads to no energy loss. But in VLSI, there is an always possibility of electric charge transfer between 2 nodes and because of this, various method can be used to reduce energy loss during charge transfer[5,6]. Fully adiabatic operation of a circuit is an ideal condition. It may be only achieved with very slow switching speed. In practical cases, energy dissipation with a charge transfer event is composed of an adiabatic component and a non-adiabatic component[5-6]. In conventional CMOS logic circuits (Fig.1), from 0 to V_{DD} transition of the output node, the total output energy CLv_{DD}^2 is drawn from power supply. At the end of transition, only $\frac{1}{2}CLv_{DD}^2$ energy is stored at the load capacitance. The half of drawn energy from power supply is dissipated in PMOS network (F). From V_{DD} to 0 transition of the output node, energy stored in the load capacitance is dissipated in the NMOS network (/F) [1]. Adiabatic logic circuits reduce the energy dissipation during switching process, and reuse the some of energy by recycling from the load capacitance [5, 6]. For recycling, the adiabatic circuits use the constant current source power supply and for reduce dissipation it uses the trapezoidal [7] or sinusoidal power supply voltage [8].

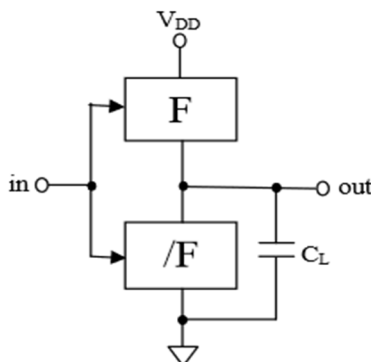


Fig.1 Conventional CMOS logic circuit with pull-up (F) and pull-down (/F) networks.

II. NEED FOR LOW POWER

In the past, the major concerns were area, performance and cost. Power consideration was then the secondary concern. Now a days, power is the primary concern due to remarkable growth and success in the field of personal computing devices and wireless communication system which demand high speed computation with complex functionality with low power consumption. The motivation for reducing power consumption differs from application to application. In the class of micro powered battery-operated portable applications such as cell phones, the goal is to keep the battery life time and weight reasonable and packaging low cost. For high performance portable computers such as laptops, the goal is to reduce the power dissipation of the electronics portion of the system to a point which is about half of total power dissipation. [1]

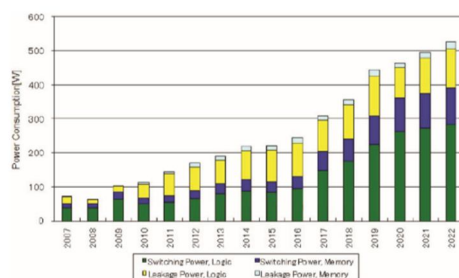


Fig.2 Increasing power consumption demand

A. Adiabatic Logic

In fig (1) constant current source (I) is used for charging of load capacitance (CL). But constant voltage source is used for charging the load capacitance in conventional CMOS logic. in PMOS n/w constant charging current corresponds to a linear voltage ramp and on-resistance is R. Initially we assume voltage across capacitor is zero [2].

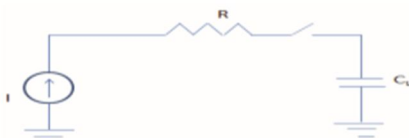


Fig.3 Adiabatic logic

IR is the voltage across switch

I^2R is P (t) in the switch

$(I^2R).T$ Energy (E) during charge E

Also $Q=C_L*V_{DD}$, $I=C_L*V_{DD}^2$

$E=(I^2R) T=(RC_L/T) C_L *V_{DD}^2$

Where, E is the energy dissipated during charging period, Q is the charge transferred to the load, C is the value of load capacitance, R is on-resistance of PMOS switch, T is charging time, and V is the value of load.

The newly proposed adiabatic circuit like Two phase clocked static CMOS logic, Complementary Energy Path Adiabatic Logic and quasi-static energy recovery logic chiefly suffer from the given failing i) Degradation of output amplitude. ii) Delay iii) Implex circuit Now a days we use DFAL to overcome the above limitations [3].

B. Steps of Adiabatic Circuit Design

General Rules Or Steps to Be followed for Adiabatic Circuit Design are as follows:

- 1) Replace each of the PMOS and NMOS devices in the pull-up and pull-down networks with T-gates.
- 2) Use expanded pull-up network to drive the true output.
- 3) Use expanded pull-down network to drive the complementary output.
- 4) The Pull up network is given the input where as the pull down network is feed will inverted input.
- 5) Both networks in the transformed circuit are used both to charge and discharge the load capacitance.
- 6) Replace DC Vdd by a pulsed power supply (Vpwr) with varying voltage to allow adiabatic operation. Today's adiabatic circuit employs MOSFET in place of T-gates.
- 7) In this work, out of the many adiabatic logic families following logic family is chosen
-Efficient Charge Recovery Logic (ECRL)

C. ECRL

Efficient Charge Recovery Logic (ECRL) is one of the adiabatic logic families and is useful for low energy systems. ECRL 2N-2P are identical and are based upon the standard CMOS family called Differential Cascade Voltage Switch Logic (DCVSL).

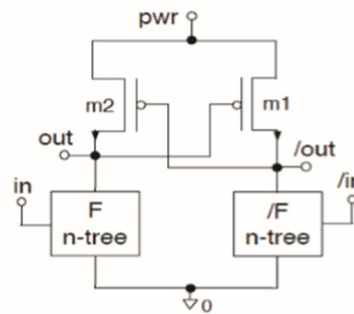


Fig. 4 Basic adiabatic circuit [5]

A pair of pull down NMOS Devices and a pair of cross-coupled PMOS devices are used in this structure to evaluate functions and to hold state respectively. It is only a quasi-adiabatic logic style because complete recovery of the power clock is not possible through the PMOS devices. ECRL is based around a pair of cross coupled PMOS transistors. The gate of each one is connected to the drain of the other and their source terminals are connected to the power clock. Complementary output signals are formed by these nodes. The function is evaluated by a series of pull down NMOS devices.

1) The designing of circuit is done by following steps:

- a) Make the truth table for the desired circuit, combine the case for 1's and 0's separately and make Boolean equation for both.
- b) Draw the cross-coupled PMOS-structure.
- c) Replace N-tree structure below out terminal with circuit for Boolean equation of 0 and N-tree below/out with circuit for Boolean equation of one

2) Working of the circuit is as follows

- 1) For the cases of 0's the F N tree will work and we will get 0 potential at out. This 0 will make transistor m1 ON and the /out terminal will follow the clock.
- 2) For the case of 1s the /F N tree will work and we will get 0 potential at /out terminal. This 0 will make transistor m2 ON and the out terminal will follow the clock.

D. AND Gate Design Using Adiabatic Logic

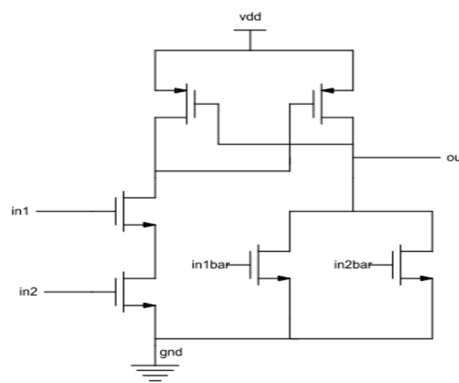


Fig. 5 Schematic of AND gate using adiabatic logic

The NAND gate circuit may be used as the starting point for the creation of an AND gate. All that needs to be added is another stage of transistors to invert the output signal. In NAND gate, if both of the A and B inputs are high, then both the NMOS transistors (bottom half of the diagram) will conduct, neither of the PMOS transistors (top half) will conduct, and a conductive path will be established between the output and Vss (ground), bringing the output low. If either of the A or B inputs is low, one of the NMOS transistors will not conduct, one of the PMOS transistors will, and a conductive path will be established between the output and Vdd (voltage source), bringing the output high.

E. XOR Gate Design Using Adiabatic Logic

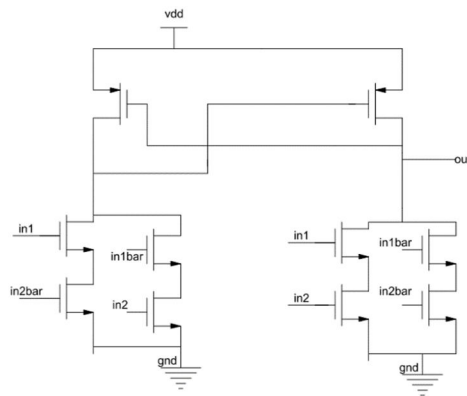


Fig. 6 Schematic of XOR gate using adiabatic logic

The XOR gate is a digital logic gate that implements an exclusive OR; that is, a true output (1) results if one, and only one, of the inputs to the gate is true (1). If both inputs are false (0) or both are true (1), a false output (0) results. A way to remember XOR is "one or the other but not both". XOR represents the inequality function, i.e., the output is HIGH (1) if the inputs are not alike otherwise the output is LOW (0). XOR can also be viewed as addition modulo 2. As a result, XOR gates are used to implement binary addition in computers. The XOR logic gate can be used as a one bit added that adds any two bits together to output one bit. For example, if we add 1 plus 1 in binary, we expect a two-bit answer, 10 (i.e. 2 in decimal). Since the trailing sum bit in this output is achieved with XOR, the preceding carry bit is calculated with an AND gate. This is the main principle in half adders and the combined AND-XOR circuit may be chained together in order to add ever longer binary numbers.

F. CLA Design Using Adiabatic Logic

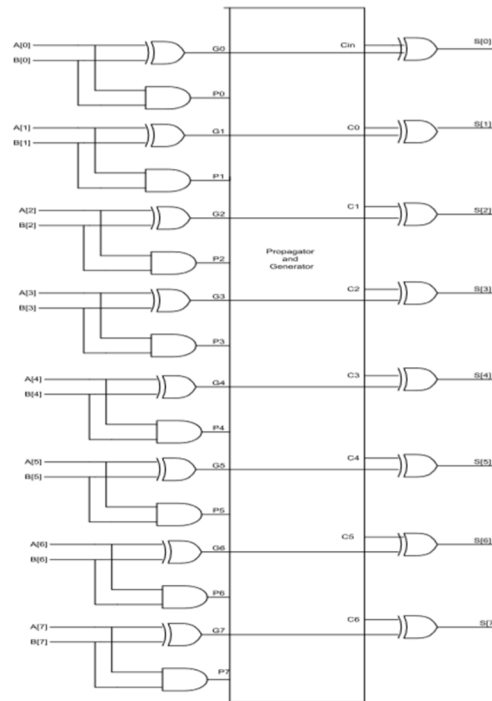


Fig. 7 Schematic of carry look ahead adder adiabatic logic

The carry-look-ahead adder is one way to speed up the carry computation. The carry-look-ahead adder breaks the carry computation into two steps, starting with the computation of two intermediate values. If the adder has two inputs a_i and b_i , then P_i and G_i can be written as follows.

$$P_i = a_i \oplus b_i$$

$$G_i = a_i * b_i$$

Using c_i carry input, the sum and carry output can be written as follows

$$S_i = c_i \oplus (a_i \oplus b_i)$$

$$C_{i+1} = G_i + P_i C_i$$

1) Carry look ahead adder depends on two things that have been explained as follows.

- a) Calculating, for each digit position, whether that position is going to propagate a carry if one comes in from the right.
- b) Combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right.
- 2) Supposing that groups of 4 digits are chosen. Then the sequence of events goes something like this:
 - a) All 1-bit adders calculate their results. Simultaneously, the look ahead units perform their calculations.
 - b) Suppose that a carry arises in a particular group. Within at most 3 gate delays, that carry will emerge at the left-hand end of the group and start propagating through the group to its left
 - c) If that carry is going to propagate all the way through the next group, the look ahead unit will already have deduced this. Accordingly, before the carry emerges from the next group the look ahead unit is immediately (within 1 gate delay) able to tell the next group to the left that it is going to receive a carry - and, at the same time, to tell the next look ahead unit to the left that a carry is on its way. The net effect is that the carries start by propagating slowly through each 4-bit group, just as in a ripple-carry system, but then move 4 times as fast, leaping from one look ahead carry unit to the next. Finally, within each group that receives a carry, the carry propagates slowly within the digits in that group. The more bits in a group, the more complex the look ahead carry logic becomes, and the more time is spent on the "slow roads" in each group rather than on the "fast road" between the groups (provided by the look ahead carry logic). On the other hand, the fewer bits there are in a group, the more groups have to be traversed to get from one end of a number to the other, and the less acceleration is obtained as a result. Deciding the group size to be governed by look ahead carry logic requires a detailed analysis of gate and propagation and this is in fact usually done.

Each look ahead carry unit already produces a signal saying "if a carry comes in from the right, I will propagate it to the left", and those signals can be combined so that each group of (let us say) four look ahead carry units becomes part of a "super group" governing a total of 16 bits of the numbers being added. The "super group" look ahead carry logic will be able to say whether a carry entering the super group will be propagated all the way through, it is able to propagate carries from right to left 16 times as fast as a naive ripple carry. With this kind of two-level implementation, a carry may first propagate through the "slow road" of individual adders, then, on reaching the left-hand end of its group, propagate through the "fast road" of 4-bit look ahead carry logic, then, on reaching the left-hand end of its super group, propagate through the "superfast road" of 16-bit look ahead carry logic. Again, the group sizes to be chosen depend on the exact details of how fast signals propagate within logic gates and from one logic gate to another. For very large numbers (hundreds or even thousands of bits) look ahead carry logic does not become any more complex, because more layers of super groups and super super groups can be added as necessary. The increase in the number of gates is also moderate: if all the group sizes are 4, one would end up with one third as many look ahead carry units as there are adders. However, the "slow roads" on the way to the faster levels begin to impose a drag on the whole system (for instance, a 256-bit adder could have up to 24 gate delays in its carry processing), and the mere physical transmission of signals from one end of a long number to the other begins to be a problem. At these sizes carry-save adders are preferable, since they spend no time on carry propagation at all. Carry look ahead logic uses the concepts of generating and propagating carries. Although in the context of a carry look ahead adder, it is most natural to think of generating and propagating in the context of binary addition, the concepts can be used more generally than this. In the descriptions below, the word digit can be replaced by bit when referring to binary addition.

The addition of two 1-digit inputs A and B is said to generate if the addition will always carry, regardless of whether there is an input carry (equivalently, regardless of whether any less significant digits in the sum carry). For example, in the decimal addition $52 + 67$, the addition of the tens digits 5 and 6 generates because the result carries to the hundreds digit regardless of whether the ones digit carries (in the example, the ones digit does not carry ($2+7=9$)). carry-look ahead adder for a number of reasons based on a balance between size and speed. The fastest design implements a parallel structure, where each bit is calculated simultaneously. We were unable to utilize this design because of our chip real estate limitations. On the other hand, a serial adder is characterized by one of the simplest transistor structures, but is excruciatingly slow. We then examined the carry look ahead adder, which is faster than the serial adder because the carry bit is calculated parallel to the XOR operation on the inputs bits. We had the option of going with a dynamic carry structure or a static structure.

Table.I Power comparison between adiabatic and cmos power dissipation

Supply voltage	Logic gates	CMOS Power dissipation (w)	Adiabatic Power dissipation (w)
1V	AND	1.045E ⁻⁶	996.3E ⁻⁹
1V	XOR	335.0E ⁻⁹	300.8E ⁻⁹
1V	CLA	14.15E ⁻⁶	12.67E ⁻⁶

III. CONCLUSION

In this research work, the carry look adder was designed using adiabatic and CMOS logic style. Both logic families are thoroughly investigated with respect to various parameter such as power, delay, power speed product and area with variation of Transistor gate sizing, frequency of operation and supply voltage. It has been observed and validated that ECRL technique dissipates less power and energy when compared with conventional method. The circuit implementation and simulation are carried out using Cadence EDA tool with gpdk 90nm technology. Hence forth adiabatic logic is preferred for Low power circuit design.

REFERENCES

- [1] Massoud Pedran Departments of EE-Systems, University of Southern California Los Angeles,"Design Technologies For Low PoweVLSI,to appear in Encyclopedia of Computer Science and Technology,1995.
- [2] Atul Kumar Maury & GagneshKumar, EnergyEfficient Adiabatic Logic for Low Power VLSI Applications, 2011 International Conference on Communication Systems and Network Technologies. K. Elissa, "Title of paper if known," unpublished.
- [3] Shipraupadhay,R.A Mishra, RKNagari, and SPSingh, Diodefree Adiabatic logic circuits, hindawani published corporations ISRN Electronics Volume2013, ArticleID 673601.
- [4] W. C. Athas, L.J. Svensson, J.G. Koller, N. Tzartzanis, and E. Chou, "Low power digital systems based on adiabatic-switching principles," IEEE Trans. VLSI Systems, vol. 2, no. 4, pp. 398-407, Dec. 1994.
- [5] J. S. Denker, "A review of adiabatic computing," in IEEE Symp. on Low Power Electronics, pp. 94-97, 1994.
- [6] J. G. Koller and W. C. Athas, "Adiabatic switching, low energy computing, and the physics of storing and erasing information," IEEE Press, in Pmc. Workshop on Physics and Computation, PhysCmp '92. oct. 1992.
- [7] T. Gabara, "Pulsed Power Supply CMOS," Technical Digest IEEE Symposium Low Power Electronics, San Diego, pp. 98- 99, Oct. 1994.
- [8] B. Voss and M. Glesner, "A low power sinusoidal clock," In Proc. of the International Symposium on Circuits and Systems, pp.108-111, May2001