

## Adiabatic Logic Based Low Power Carry Select Adder for future Technologies

Mary Angel Asha Latha J(M.Tech) Prof. G.V.R.Sagar(M.Tech, P.hD)

**Abstract:** Adders are of fundamental importance in a wide variety of digital systems. Many fast adders exist, but adding fast using low area and power is still challenging. This paper presents a new bit block structure that computes propagate signals called “carry strength” in a ripple fashion. Several new adders based on the new carry select Adder structure are proposed. Comparison with well-known conventional adders demonstrates that the usage of carry-strength signals allows high-speed adders to be realised at significantly lower cost and consuming lower power than previously possible. As well as in this paper we are concentrating on the heat dissipation & we are reducing the current using adiabatic logic.

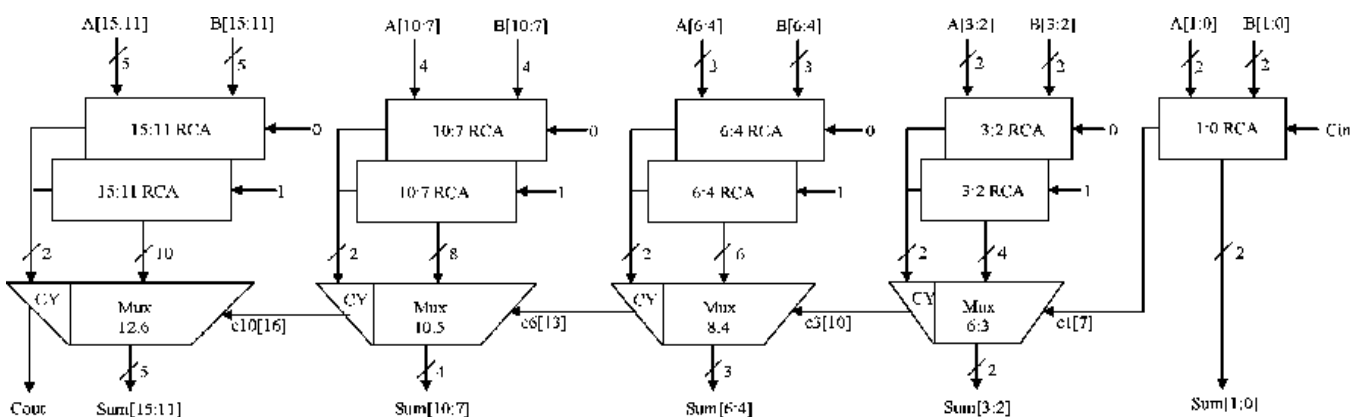
### I. INTRODUCTION

The importance of a fast, low-cost binary adder in a digital system is difficult to overestimate. Not only are adders used in every arithmetic operation, they are also needed for computing the physical address in virtually every memory fetch operation in most modern CPUs. Adders are also used in many other digital systems including telecommunications systems in places where a full-fledged CPU would be superfluous. Many styles of adders exist. Ripple adders are the smallest but also the slowest. More recently, carry-skip adders [1, 2, 3] are gaining popularity due to their high speed and relatively small size. Normally, in an N-bit carry-skip adder divided into a proper number of M-bit blocks [1, 4], a long-range carry signal starts at a generic block  $B_i$ , rippling through some bits in that block, then skips some blocks, and ends in a block  $B_j$ . If the carry does not end at the LSB of  $B_j$  then rippling occurs in that block and an additional delay is needed to compute the valid sum bits. Carry-look-ahead and carry-select adders [1] are very fast but far larger and consume much more power than ripple or carry-skip adders. Two of the fastest known addition circuits are the Lynch-Swartzlander’s [5] and Kantabutra’s [6]. hybrid carry-look-ahead adders. They are based on the usage of a carry tree that produces carries into appropriate bit positions without back propagation. In order to obtain the valid sum bits as soon as possible, in both Lynch-Swartzlander’s and Kantabutra’s adders the sum bits are computed by means of carry-select blocks, which are able to perform their operations in parallel with the carry-tree.

This paper presents two new families of adders, both based on a new bit carry Select & adiabatic structure that computes propagate signals called “carry-strength” in a ripple fashion. The first family of adders is a family of new carry-select adders that are significantly faster than traditional carry-select adders while not much larger. The second family of adders is a family of hybrid lookahead adders similar to those presented in [5, 6] but significantly smaller and still comparable in speed.

In our new type of carry-select adder, the new block structure eliminates the delay due to the rippling at the end of the life of a long-range carry signal. The main idea is, that for each bit position  $k$  in a block  $B_j$  we compute whether the carry-in to position  $k$  comes from the carry-in to block  $B_j$ , or whether this carry is internally generated in block  $B_j$ . To this purpose we will use a new type of bit block, in which we will compute propagate signals that start at the LSB of the block and end at every bit position. We find it helpful to call the complements of these “carry-strength” signals, because they indicate for each bit position whether the carry-in to that position originates within the same bit block.

In basic arithmetic computation, adder is still plays an important role though many people focus on more complex computation such as multiplier, divider, cordic circuits. Although several algorithms and architectures are implemented in literature, there is not an general architecture for measuring performance equally. Much architecture is tested under different conditions which possibly result in variant performance even implemented with the same algorithm.



**Fig1: Conventional Carry Select Adder using RCA**

CLA is proved to have good performance using in high speed adder, so in many papers this architecture are used commonly. STCLA – Spanning Tree Using CLA uses a tree of 4-bit Manchester Carry-Lookahead chains (MCC) to generate carry for different bit position. RCLCSA – Recursive CLA/CSA Adder uses the same conception as STCLA except the lengths of its

carry chains are variant, not fixed. HSAC – High Speed Adder Using CLA uses Ling’s adder which solves the transition of carry propagation delay.

Adder using different implementation is the most critical issue. For example, STCLA and RCLCSA use dynamic CMOS while HSAC uses static CMOS. Here, we want to implement a general architecture for measuring this three different algorithm which means we can use both dynamic CMOS and static CMOS to implement these algorithms for equal comparison. At last, I will offer my new architecture improved from the original paper.

Let me talk about the original implementation. It’s based on the Adiabatic adder. But it takes advantage of the characteristics of CMOS circuit. Generally, we don’t use “bar”(inverted) as we conduct every equation . But in reality, “bar” is automated added at the output of logic circuits. So, they use this special characteristic to reduce the carry propagation time

## II. POSITIVE FEEDBACK ADIABATIC LOGIC

The structure of PFAL logic is shown. Two n-trees realize the logic functions. This logic family also generates both positive and negative outputs. The two major differences with respect to ECRL are that the latch is made by two pMOSFETs and two nMOSFETs, rather than by only two pMOSFETs as in ECRL, and that the functional blocks are in parallel with the transmission pMOSFETs. Thus the equivalent resistance is smaller when the capacitance needs to be charged. The ratio between the energy needed in a cycle and the dissipated one can be seen in figure 4. During the recovery phase, the loaded capacitance gives back energy to the power supply and the supplied energy decreases.

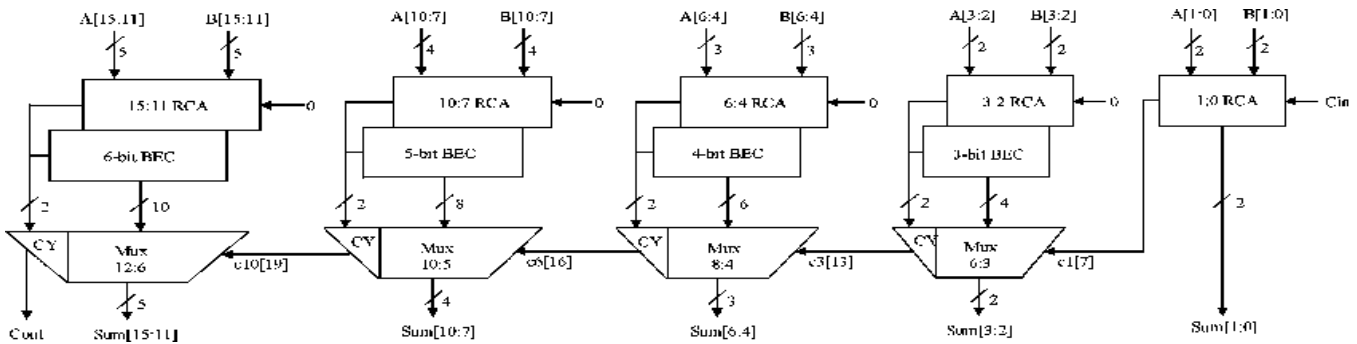
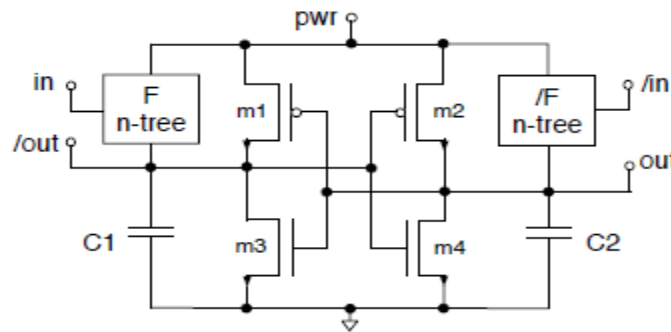


Fig1: Modified Carry select adder using BEC General schematic for PFAL family



## III. POWER DISSIPATION IN ADIABATIC LOGIC GATES

A limiting factor for the exponentially increasing integration of microelectronics is represented by the power dissipation. Though CMOS technology provides circuits with very low static power dissipation, during the switching operation currents are generated, due to the discharge of load capacitances, that cause a power dissipation increasing with the clock frequency. The adiabatic technique prevents such losses: the charge does not flow from the supply voltage to the load capacitance and then to ground, but it flows back to a trapezoidal or sinusoidal supply voltage and can be reused. Just losses due to the resistance of the switches needed for the logic operation still occur. In order to keep these losses small, the clock frequency has to be much lower than the technological limit. In the literature, a multitude of adiabatic logic families are proposed. Each different implementation shows some particular advantages, but there are also some basic drawbacks for these circuits. The goal of this paper is to compare different adiabatic logic families and to investigate their robustness against technological parameter variations. For this purpose three adiabatic logic families are evaluated and the impact of parameter variations on the power dissipation is determined. Both inter-die (and global) and intra-die (or local) parameter variations of different components in the same sub-circuit are considered. The most important factor is the threshold voltage variation, especially for sub-micrometer processes with reduced supply voltage. This was also found for low voltage CMOS circuits, cf., where the fundamental yield factor was the gate delay variation (in CMOS the power dissipation is not significantly dependent on the threshold voltage). For adiabatic circuits the timing conditions are not critical, because the clock frequency is particularly low, and therefore the outputs can always follow the clocked supply voltage. Here the yield critical requirement is the power dissipation that has a very low nominal value. Hence it exhibits large relative deviations due to parameter variations that can lead to the violation of the specifications.

The general PFAL gate consists of a two cross coupled inverters and two functional blocks F and /F (complement of F) driven by normal and complemented inputs which realizes both normal and complemented outputs. Both the functional blocks implemented with n channel MOS transistors. The equations used to implement PFAL adder and the corresponding sum and carry implementations.

The logical organization of conventional and adiabatic adders is constructed by the replication of 2 and 4, 4bit blocks for %bit and 16-bit adder, respectively. Each 4bit block may be viewed as consisting of a carry unit, a sum generation unit, and a sum selection unit. (In practice, the three parts are of course not necessarily so distinctly separated.) The carries and both types of sum bits are produced using lookahead functions as much as possible. The detailed logic design of this adder can be found in [10]. The adiabatic adder results after the substitution of the conventional CMOS adder's blocks with the corresponding adiabatic. Regarding the delay for an n-bit adiabatic carry select adder, which is constructed by mbit blocks ( $m < n$ ), we obtain:

where  $2t$ , is the delay from the computation of the partial sum  $P$ , and  $G_{iand}$ ,  $N(t+2t_{inv}7$  with  $N=n/m$ , the delay of carry propagation through the m-bit blocks. The design of this adder involved re-thinking of the circuit according to the principle of the adiabatic switching and no changes were held in the above equations. Also, to best of our knowledge a similar adiabatic conditional sum adder hasn't been introduced until now. Finally, following similar substitutions, for the conditional sum adder whose structure resembles that of carry select adder, we can result in another low power adiabatic adder.

The schematic and simulated waveform of the carry select adder. The energy stored at output can be retrieved by the reversing the current source direction during discharging process. Hence adiabatic switching technique offers the less energy dissipation in PMOS network and reuses the stored energy in the output load capacitance by reversing the current source direction.

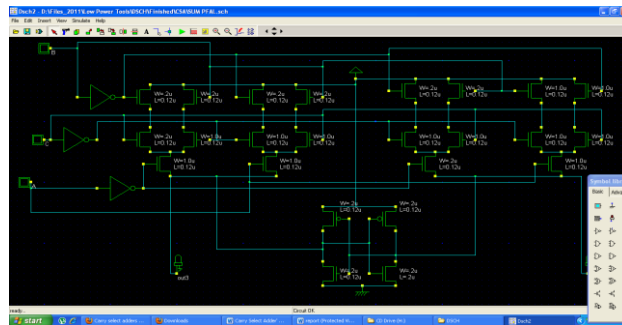


Fig 3: PFAL Sum Block

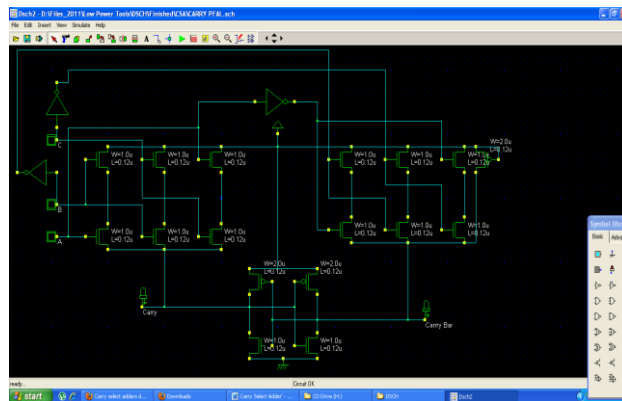


Fig4: PFAL Carry Block

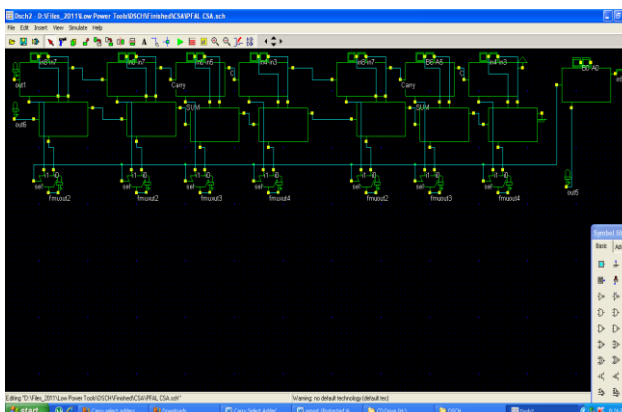
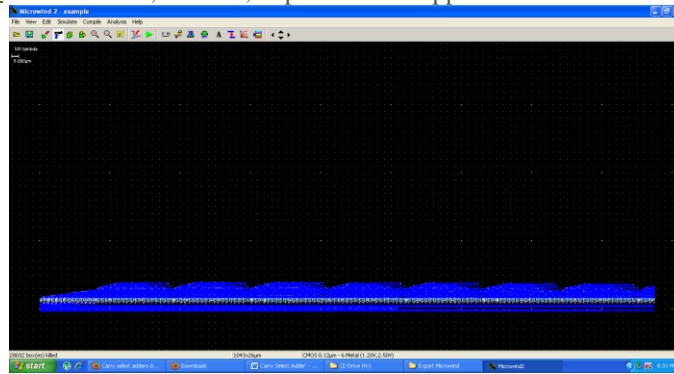
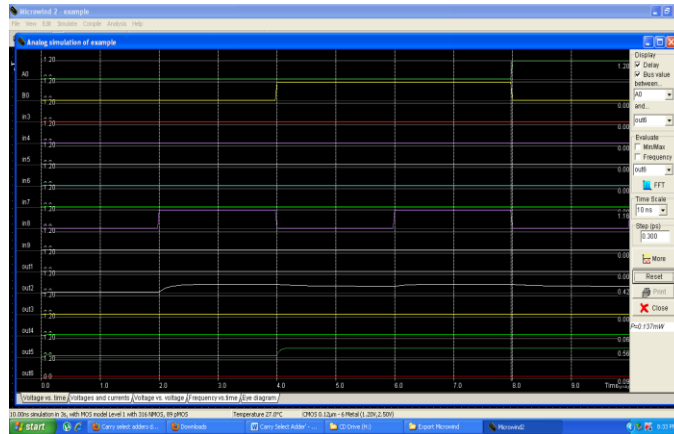


Fig5: Proposed Adiabatic CSA



**Fig6: Proposed PFAL CSA Layout with area**



**Fig7: Proposed Circuit Power Results**

#### IV. CONCLUSION

The new implementation is based on the original architecture, so it can be used in both static CMOS and dynamic CMOS circuits. And through my architecture, I can reduce power and area consumption but sacrifice some timing (which can be neglected). By this implementation, I prove that the new architecture is really better than the traditional HSAC. After reading some papers, I realize that improving adder is very difficult now because of the transistor level. If we want to get higher performance we must reduce the complexity in transistor level.

#### REFERENCES

- [1] KOREN, I.: "Computer arithmetic algorithms", Prentice-Hall, 1993
- [2] KANTABUTRA, V.: "Designing optimum one-level carry-skip adders", *IEEE Trans. on Comp.*, 1993, Vol. 42, n°6, pp.759-764.
- [3] CHAN, P.K., SCHLAG, M.D.F., THOMBORSON, C.D., OKLOBDZIJA, V.G.: "Delay optimization of carry-skip adders and block carry-look-ahead adders", *Proc. of Int'l Symposium on Computer Arithmetic*, 1991, pp.154-164.
- [4] NAGENDRA, C., IRWIN, M.J., OWENS, R.M.: "Area-time-power tradeoffs in parallel adders", *IEEE Trans. CAS-II*, 43, (10), pp. 689-702.
- [5] T. LYNCH, E.E. SWARTZLANDER, "A spanning-tree carry-look-ahead adder", *IEEE Trans. on Comp.*, Vol. 41, n°8, Aug. 1992.
- [6] V. KANTABUTRA, "A recursive carry-look-ahead/carry-select hybrid adder", *IEEE Trans. on Comp.*, Vol. 42, n°12, Dec. 1993.
- [7] R. Zimmermann and H. Kaeslin, "Cell-Based multilevel Carry-Increment Adders with Minimal AT- and PT-Products, unpublished manuscript. <http://www.iis.ee.ethz.ch/~zimmi/>
- [8] Tyagi, "A reduced-area scheme for carry-select adders" *IEEE Trans. on Comp.*, Vol. 42, n°10, Oct. 1993.