High Speed VLSI Implementation of a Finite Field Multiplier Using Adiabatic Logic

First Author[1], Second Author[2]

[1] *M. Chandrasekhara Rao*
(M-Tech II/II Embedded System)
Gudlavalleru Engineering College, Gudlavalleru.

[2] *E.Vargil Vijay*
Assistant professor in ECE Deportment
Gudlavalleru Engineering College, Gudlavalleru

ABSTRACT

In this paper, low power multiplier design using Finite field multiplier using backend design is I investigated. Adiabatic circuits are very low power circuits compared with CMOS logic circuits, provided the Power Clock Generators consumes less power and mutilate all low power advantages from the adiabatic logic by consuming large portion of the total power in the clock generation circuitry. Also clock routing is major challenge in the adiabatic, because of routing-delay between the gates. Compared with the conventional CMOS implementation, this design achieves energy savings from 50% to 74% for clock rates ranging from 100MHz to 300MHz.

Unlike most research involving finite field multipliers this work targets low power multiplier through the application of various power reduction techniques to different types of multipliers and comparing their power consumption among other factors, rather than comparing complexity measures such as gate count on area gate count is used as a starting point to choose potential architectures, namely , polynomial and normal basis architectures power reduction techniques employed are mainly concerned with architecture and logic level low power techniques, and now finite multiplier using adiabatic. They include supply voltage reduction. As well as in this paper I am concentrating on the heat dissipation & reducing the current using adiabatic logic.

Keywords *- Low power, Adiabatic, cmos*

Reed-Solomon codes are based on finite field arithmetic which involves dining closed binary operations over finite sets of elements. Unfortunately, a full review of finite fields is beyond the scope of this.

1 INTRODUCTION

Moore's law describes the requirement of the transistors for VLSI design, it gives the empirical observation that component density and performance of integrated circuits, doubles every year, which was then revised to doubling every two years. With the help of the scaling rules set by Dennard, smart optimization can be achieved by means of timely introduction of new processing techniques in device structures, and materials. To overcome the power and area requirements of the computational complexities, the dimensions of transistors are shrunk into the deep sub-micron region and predominantly handled by process engineering. Driven by tremendous advances in lithography, the 65nm process technology node featuring approximately 32nm transistors is in vogue right now in high volume production. Moreover the technology migration has become much costly for process the design in terms of its physical design. Developers are forced to bare the tool cost in order to achieve the low power requirements. The transistor cost versus lithographic tool cost is given in the silicon technology future road map, it is noted that transistor cost has decreased seven orders of magnitude whereas tool cost has increased. Thus, the alternate method or migration of process engineering is most invited. As a brief overview, we will start with the simplest example of a finite field which is the binary field consisting of the elements. Traditionally referred to as, the operations in this field are defined as integer addition and multiplication reduced modulo 2.We can create larger fields by extending into vector space leading to finite fields of size 2m.The field G is thus defined as a field with 2m elements each of which is a binary multiple.

Using this definition, we can group m bits of binary data and refer to it as an element of field G. This in turn allows us to apply the associated mathematical operations of the field to encode and decode data. For our purposes, we will limit our discussion to the finite field. This field consists of sixteen elements and two binary operations, addition and multiplication. There are two alternate (but equivalent) representations for the field elements. First, all nonzero elements represented as powers of a primitive field element (i.e. each nonzero element is of the form α _n for n = 0,1, . . . 14). Second, each element has an equivalent representation as a binary 4-tuple.While the representation has great mathematical convenience, digital hardware prefers the binary 4-tuple representation. These representations are illustrated in Table 1.

Table 1: Canonical representation of finite field.

Element	0	α^0	α^1	α^2	α^3	α^4	α^5	α^6
Representation	0000	0001	0010	0100	1000	0011	0110	1100
Element	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
Representation	1011	0101	1010	0111	1110	1111	1101	1001

2. POSITIVE FEEDBACK ADIABATIC LOGIC

Adiabatic logic family generates both positive and negative outputs. The two major differences with respect to ECRL are that the latch is made by two p MOSFETs and two n MOSFETs, rather than by only two p MOSFETs as in ECRL, and that the functional blocks are in parallel with the transmission p MOSFETs. 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 6. During the recovery phase, the loaded capacitance gives back energy to the power supply and the supplied energy decreases.Fig.1 shows Finite field elements from the Galois field GF(2^k) are represented as polynomials with binary valued coefficients, as such, multiplication in the field is defined modulo an irreducible polynomial of degree k-1one of the

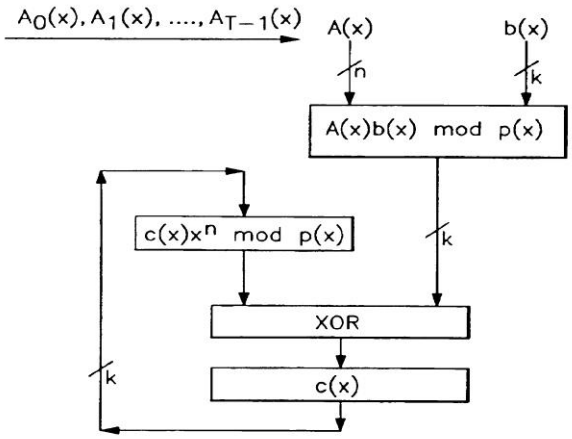

Figure.1. Block of finite field multiplier.

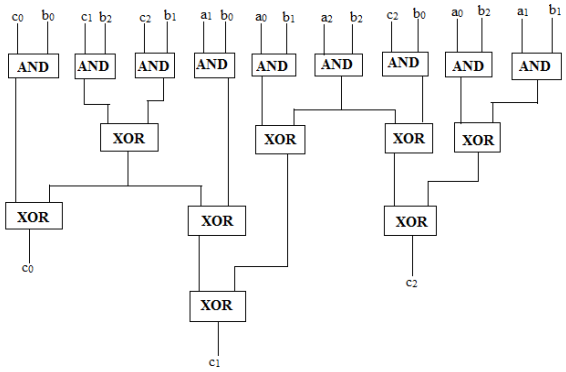

Figure.2. Block diagram of XOR and AND logic Gates.

Multiplicands is treated in blocks of polynomials of degree n-1 so that the multiplier operates over T cycles where k=nT. If K is not a composite number to start with , higher order terms are added, so that multipliers are now constructible even when k is prime since n<k, the construction of the needed multiplier circuits are much simpler. Designers are now provided with an opportunity of easily trading off circuit speed for circuit. Complexity in an orderly and structured fashion. Fig. 2 shows is a block diagram of a circuit for block circuit for multiplication accordance with the logical gates XOR and AND gates.

3. 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 power dissipation increasing with the clock frequency. The adiabatic technique prevents such losses: the charge does not owes from the supply voltage to the load capacitance and then to ground, but it owes 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 intertie (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, where the fundamental yield factor was the gate delay variation (in CMOS the power dissipations 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 and p-channel MOS transistors. Figure.3shows the AND and XOR function in PFAL, in this circuit one functional block is designed as a AND gate and another block is designed as a XOR gate, because these two logic gates are useful in finite field multiplier for Addition and Multiplication operations. In this circuit when the both inputs are high then outbar is low and out is vary (either low or high) depending upon clock. When both inputs are low then outbar is low and out is high depending up on clock. Either any one input is high and another input is low then outbar is high and out is low. Fig.4 shows the Simulation results of the AND and XOR function in PFAL, where A and B are two inputs, clk is clock, outbar is output results of XOR gate and out is output result of And and XOR function in PFAL. Fig.5 shows the Layout diagram for AND and XOR function in PFAL. If designed a circuit in Digital Schematic Editor Tool, this tool has a facility or option make verilog, when click on that option verilog file will be generated automatically of that circuit, this program will automatically saved in Micro wind layout tool, here we can see the AREA of that circuit in the form of Length and Breath, it will be shown in Layout form. When we run to this layout form then it will display the power results of that circuit, how much power it will consume, it shown in Fig.6. This power results are taken in voltage verses current.

3. MATHEMATICAL BACKGROUND

In this Section the mathematical background used for the design of the two architectures is presented. The basic GF(2k) field arithmetic is analyzed and a correspondence with binary logic operations is made, for GF(2k) field is described.

4. SIMULATION RESULTS

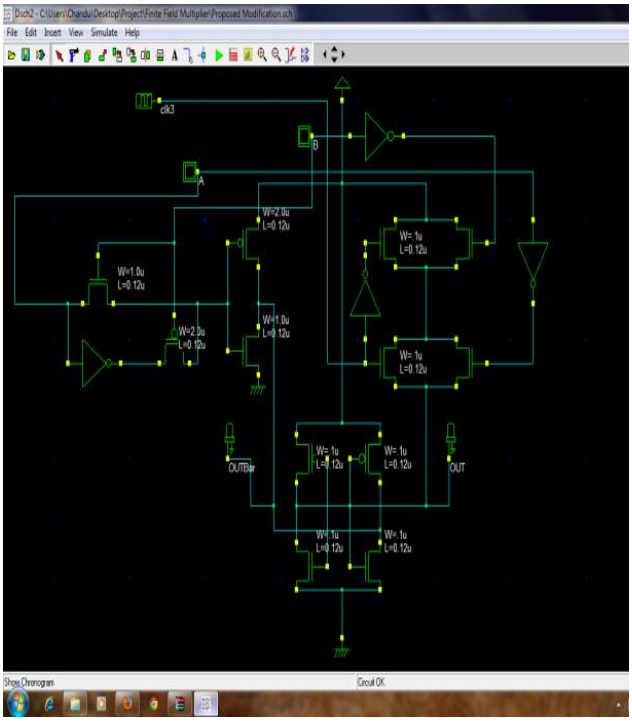

Figure.3. AND and XOR function in adiabatic logic.

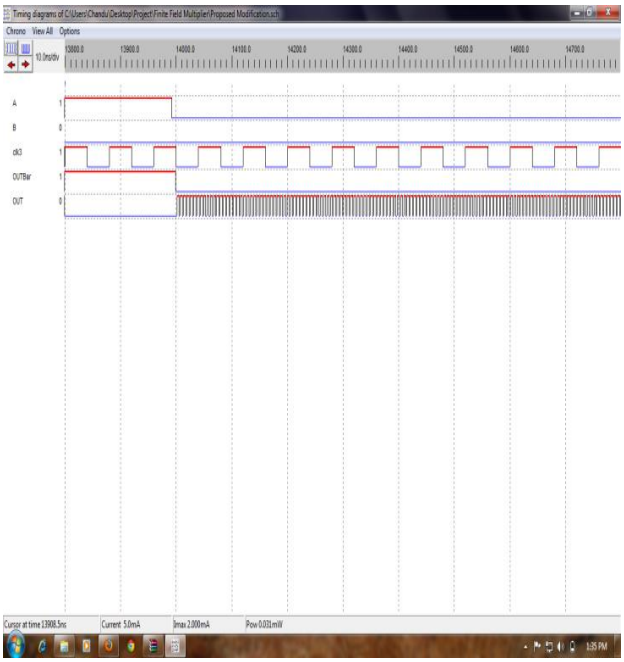

Figure.4. Result of AND and XOR function in Adiabatic logic.

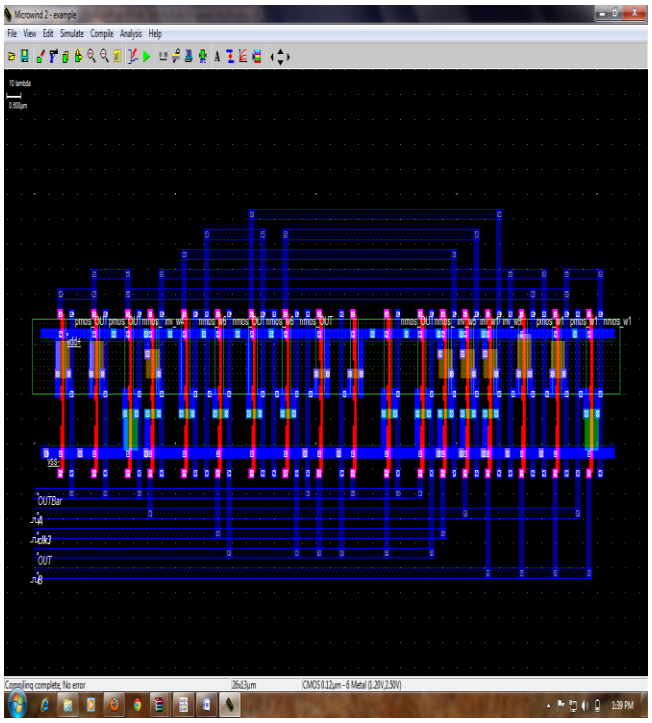

Figure.5. Layout diagram for AND and XOR function in Adiabatic logic.

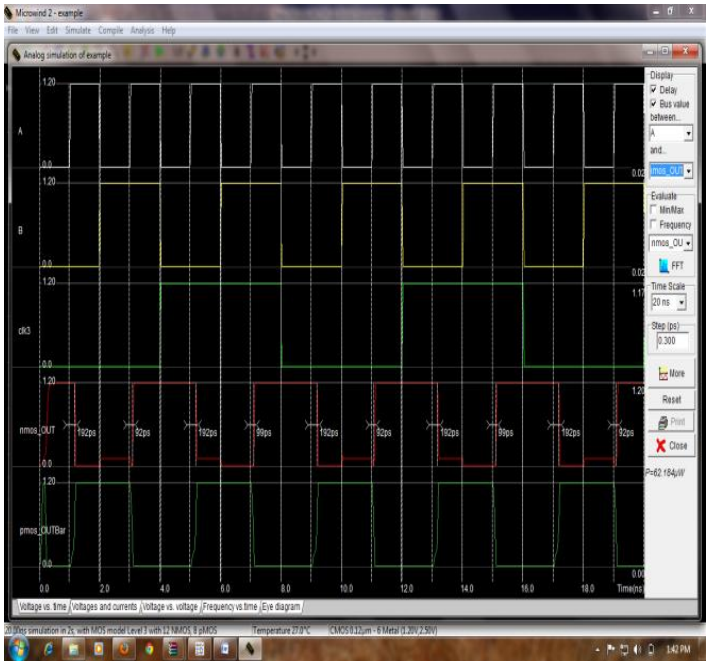

Figure.6. Power Results of AND and XOR function in adiabatic logic.

6. 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 After reading some papers, I realize that improving multiplier is very difficult now because of the adiabatic. If we want to get higher performance we must reduce the complexity in transistor level

REFERENCES

[1] A. H. Namin, H. Wu and M. Ahmadi," *High Speed VLSI Implementation of a Finite Field Multiplier Using Redundant Representation*, IEEE Trans. May 2009.

[2] A. H. Namin, H. Wu and M. Ahmadi, *A New Finite Field Multiplier Using Redundant Representation*, IEEE Trans. on Computers, Vol.57, no.5, May 2008.

[3] B. Ansari And H. Wu,: "*Efficient Finite Field Processor For GF (2^163) and its VLSI Implementation*", Fourth International Conference on Information Technology (ITNG), pp.1021-1026, April 2007

[4] W. Tang, H. Wu, And M. Ahmadi, "*VLSI Implementation of bit Parallel Word–Serial Multiplier In GF (2^233)*, Proceeding Of The 3rd International IEEE-NEWCAS Conference, pp. 399-402, June 2005.

[5] A. Reyhani-Masoleh and M. A. Hasan, *Low Complexity Word-Level Sequential normal basis multipliers*, IEEE Trans. Computers, Vol. 54, no.2 ,pp.98 – 110, Feb 2005.

[6] H. Wu, M. A. Hasn, I. F. Blake, Shuhong Gao , *Finite Field Multiplier Using Redundant Representation*, IEEE Trans. Computers, Vol.51, no.11, pp. 1306 – 1316(2002).

[7] Certicom Corp., *Current Public-Key Cryptographic System*, White Paper, http://www.certicom.com(2000).

[8] KOREN, I.: "Computer arithmetic algorithms", Prentice-Hall, 1993

[9] KANTABUTRA, V.: "Designing optimum one-level carry-skip adders", *IEEE Trans. on Comp.*, 1993, Vol. 42, n°6, pp.759-764.

[10] 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.

[11] NAGENDRA, C., IRWIN, M.J., OWENS, R.M.: "Area-time-power tradeoffs in parallel adders", *IEEE Trans. CAS-II*, 43, (10), pp. 689-702.

[12] T. LYNCH, E.E. SWARTZLANDER, "A spanning-tree carry-look-ahead adder", *IEEE Trans. on Comp.*, Vol. 41, n°8, Aug. 1992.

[13] V. KANTABUTRA, "A recursive carry-look-ahead/carry-select hybrid adder", *IEEE Trans. on Comp.*, Vol. 42, n°12, Dec. 1993.

[14] 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/

[15] A. Tyagi, " A reduced-area scheme for carry-select adders" *IEEE Trans. on Comp.*, Vol. 42, n°10, Oct. 1993.