

Asynchronous Domino Logic Pipeline Based ECRL

K. Shanmuga Priya¹, A. Indhumathi², R. Vignesh Chandrasekar³

^{1, 2, 3}Ranganathan Engineering College, Coimbatore, Tamil Nadu, India-641109

Email address: ¹shanmugapriya301@gmail.com

Abstract— This project presents a high-throughput and ultralow-power asynchronous domino logic pipeline design method, targeting to latch-free and extremely fine-grain or gate-level design. The data paths are composed of a mixture of dual-rail and single-rail domino gates. Dual-rail domino gates are limited to construct a stable critical data path. Based on this critical data path, the handshake circuits are greatly simplified, which offers the pipeline high throughput as well as low power consumption. Moreover, the stable critical data path enables the adoption of single-rail domino gates in the noncritical data paths. This further saves a lot of power by reducing the overhead of logic circuits. An 8×8 array style multiplier is used for evaluating the proposed pipeline method. Compared with a hybrid – rail asynchronous domino logic pipeline, the proposed ECRL pipeline consumes less power and also reduces no of transistor used.

Keywords— Asynchronous pipeline, critical data path, Dual- rail domino gate, single-rail domino gate, ECRL.

I. INTRODUCTION

The main objective of this research is to provide new low power solutions for Very Large Scale Integration (VLSI) designers. Especially, this work focuses on the reduction of the power dissipation, which is showing an ever-increasing growth with the scaling down of the technologies. Various techniques at the different levels of the design process have been implemented to reduce the power dissipation at the circuit, architectural and system level. Furthermore, the number of gates per chip area is constantly increasing, while the gate switching energy does not decrease at the same rate, so the power dissipation rises and heat removal becomes more difficult and expensive. Then, to limit the power dissipation, alternative solutions at each level of abstraction are proposed. The dynamic power requirement of CMOS circuits is rapidly becoming a major concern in the design of personal information systems and large computers. In this work, a new CMOS logic family called *ADIABATIC LOGIC*, based on the adiabatic switching principle is presented. The term adiabatic comes from thermodynamics, used to describe a process in which there is no exchange of heat with the environment. The adiabatic logic structure dramatically reduces the power dissipation. The adiabatic switching technique can achieve very low power dissipation, but at the expense of circuit complexity. Adiabatic logic offers a way to reuse the energy stored in the load capacitors rather than the traditional way of discharging the load capacitors to the ground and wasting this energy.

II. ADIABATIC

The term “adiabatic” describes a thermodynamic process in which there is no energy exchange with the environment,

and therefore no dissipation of energy or power occurs. But in VLSI, as the transfer of the electric charge takes place between the nodes of a circuit, various techniques can be applied to minimize this energy loss during this charge transfer process. Adiabatic technology is basically used to reduce the energy or power dissipation during the switching process and further reuse some of the energy by recycling it from the load capacitance. For recycling, the adiabatic circuits use the constant current source power supply and for reduce dissipation it uses the trapezoidal or sinusoidal power supply voltage. Adiabatic logic works on the principal of reversible logics. Reversible logic means when a system erases a bit of information, it dissipates heat and today’s computers erase a bit of information every time they perform a logic operation. These are called irreversible logics. And in contrast to it, the logic operations that do not erase information these are called reversible logics. Today most research has focused on building adiabatic logic out of CMOS. However, current CMOS technology, though fairly energy efficient compared to similar technologies, dissipate energy as heat, mostly when switching. To reduce this there are two fundamental rules:-

1. Never turn on a transistor when there is voltage difference between drain and source.
2. Never turn off a transistor that has current flowing through it.

Logic and the energy flows through the transistor in a controlled manner. As the second rule states that the transistor must not be turned off when there is current flowing through it, reason behind this is that the transistors are not perfect switches. The change from one state to another is directly proportional to the speed at which the gate voltage changes. Figure 1 shows the Charging and Discharging in CMOS System. In a CMOS inverter circuit during the negative half of the supply voltage the load capacitance gets charged through the functional block F (PMOS), in the above figure 1.

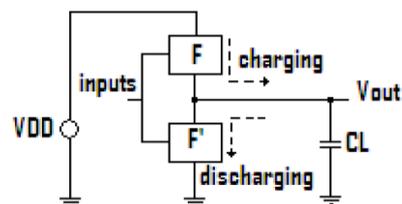


Fig. 1. Charging and discharging of CMOS system

There is some energy required to store this energy. Now 50% of this energy is dissipated as heat in the PMOS, hence the capacitance is charged by only 50% of the power supplied. Now during the positive half of the supply voltage the PMOS is switched off and the functional block FBAR (NMOS) is

switched on because of which the energy stored in the load capacitance is moved to the ground, and hence it represents the worst case of energy wastefulness. Figure 2 Charging and Discharging in Adiabatic system figure 2 shows the Charging and Discharging in Adiabatic System. In figure F functional block use PMOS transistor and FBar use NMOS transistor. Ease of Use

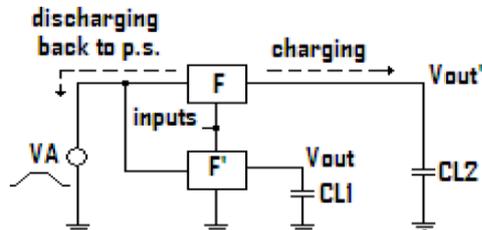


Fig. 2. Charging and discharging of adiabatic system

III. EFFICIENT CHARGE RECOVERY LOGIC

Figure 3 shows the Efficient Charge Recovery Logic (ECRL) and was proposed by Moon and Jeong. It uses two cross-coupled PMOS transistors and two NMOS transistors in the N- functional blocks for the ECRL adiabatic logic block.

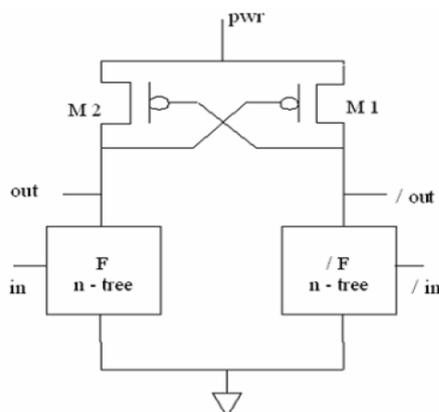


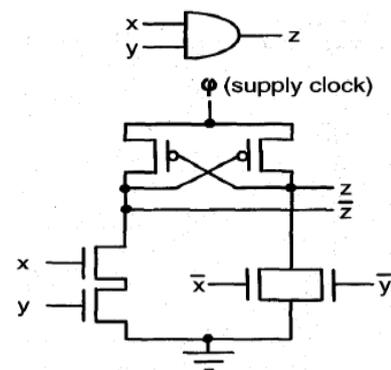
Fig. 3. Efficient charge recovery logic (ECRL)

In above figure pwr is the AC power supply is used for ECRL gates, so as to recover and reuse the supplied energy. Both out and /out are generated so that the power clock generator can always drive a constant load capacitance independent of the input signal. Full output swing is obtained because of the cross-coupled PMOS transistors in both precharge and recover phases. But due to the threshold voltage of the PMOS transistors, the circuits suffer from the non-adiabatic loss both in the precharge and recover phases. That is, to say, ECRL always pumps charge on the output with a full swing. However, as the voltage on the supply clock approaches to $|V_{tp}|$, the PMOS transistor gets turned off. [6] The ECRL circuits are operated in a pipelining style with the four-phase supply clocks. When the output is directly connected to the input of the next stage (which is a combinational logic), only one phase is enough for a logic value to propagate. However, when the output of a gate is fed back to the input, the supply clocks should be in phase. A latch is one of the simplest cases which have a feedback path.

The input signals propagate to the next stage in a single phase, and the input values are stored in four phases (1-clock) safely.

A. Working

Let us assume in is at high and inb is at low. At the beginning of a cycle, when the supply clock, pwr rises from zero to VDD, out remains at a ground level, because in turns on F- tree (NMOS logic tree). /out follows pwr through M1. When pwr reaches VDD, the outputs hold valid logic levels. These values are maintained during the hold phase and used as inputs for the evaluation of the next stage. After the hold phase, pwr falls down to a ground level, /out node returns its energy to pwr so that the delivered charge is recovered. Thus, the clock pwr acts as both a clock and power supply. For instance if we consider a circuit of a two input(x, y) AND gate with output(y).



So by substituting the value of y and at the positions of F and F/ n tree positions in the figure 1 respectively we get the AND circuit implementation using ECRL technology.

IV. PROPOSED WORK

A. Asynchronous Circuit

An asynchronous circuit, or self-timed circuit, is a sequential digital logic circuit which is not governed by a clock circuit or global clock signal. Instead they often use signals that indicate completion of instructions and operations, specified by simple data transfer protocols. This type is contrasted with a synchronous circuit in which changes to the signal values in the circuit are triggered by repetitive pulses called a clock signal. Most digital devices today use synchronous circuits. However asynchronous circuits have the potential to be faster, and may also have advantages in lower power consumption, lower electromagnetic interference, and better modularity in large systems. Asynchronous circuits are an active area of research in digital logic design. The advantages are

1. Achieve average case performance
2. Consume power only when needed
3. Provide easy modular composition
4. Do not require clock alignment at interfaces
5. Metastability has time to resolve
6. Avoid clock distribution problems
7. Exploit concurrency more gracefully

- 8. Provide intellectual Challenge
- 9. Exhibit intrinsic elegance

B. Asynchronous Pipeline Based on Constructed Critical Data Path

Figure 4 shows the block diagram of the proposed system. The pipeline is designed based on a stable critical data path that is constructed using special dual-rail logic. The critical data path transfers a data signal and an encoded handshake signal. Noncritical data paths, composed of single-rail logic, only transfer data signal. A static NOR gate detects the dual-rail critical data path and generates a total done signal for each pipeline stage. The outputs of NOR gates are connected to the precharge ports of their previous stages.

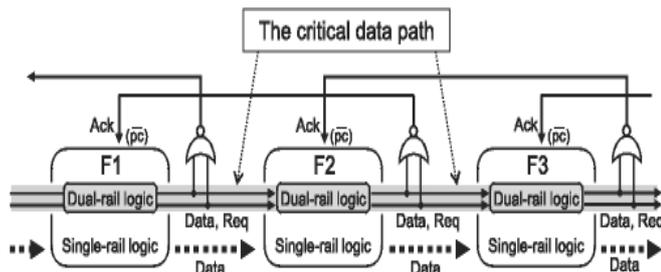


Fig. 4. Block diagram of ADPE

ADPE has the same protocol as PS0. The difference is that a total done signal is generated by detecting only the critical data path instead of the entire data paths. Such design method has two merits. First, the completion detector is simplified to a single NOR gate, and the detection overhead is not growing with the data path width. Second, the overhead of function block logic is reduced by applying single-rail logic in noncritical data paths. As a result, ADPE has a small overhead in both handshake control logic and function block logic, which greatly improves the throughput and power consumption.

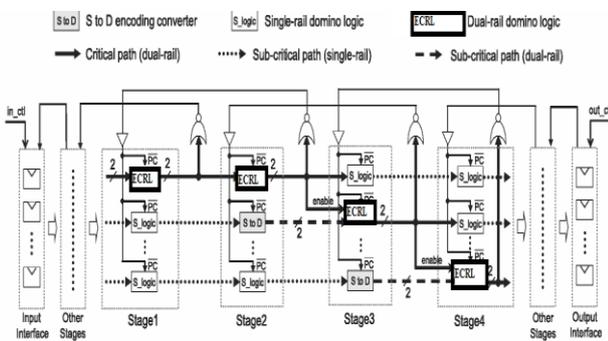


Fig. 5. Structure of ADPE

Finding a stable critical data path in function blocks is very important in the proposed design. The problem is that it is difficult to get a stable critical data path using traditional logic gates. Traditional logic gates have the gate-delay data-dependence problem. The gate delay is dependent on input data patterns. Adding delay elements is an intuitive way to construct a stable critical data path. However, this method

needs complex timing analysis and would cause huge overhead of delay elements. This paper introduces an efficient solution that uses ECRLs to construct the critical data path.

Encoding conversion: Since the completion detector detects only the constructed critical data path, the noncritical data paths do not have to transfer encoded handshake signal anymore. The logic overhead in the noncritical data paths can be reduced using single-rail domino gates instead of dual-rail domino gates. However, single-rail domino gate and dual-rail domino gate use different encoding schemes. It has encoding compatibility problem when a single-rail domino gate connects to a dual-rail domino gate. Encoding converter needs to be designed to solve the problem.

V. PROPOSED WORK

A. Asynchronous Pipeline Based on Constructed Critical Data Path

Figure 6 shows the block diagram of the proposed asynchronous pipeline (AHDE). The pipeline is designed based on a stable critical data path that is constructed using special dual-rail logic. The critical data path transfers a data signal and an encoded handshake signal. Noncritical data paths, composed of single-rail logic, only transfer data signal. A static NOR gate detects the dual-rail critical data path and generates a total done signal for each pipeline stage. The outputs of NOR gates are connected to the precharge gates of their previous stages.

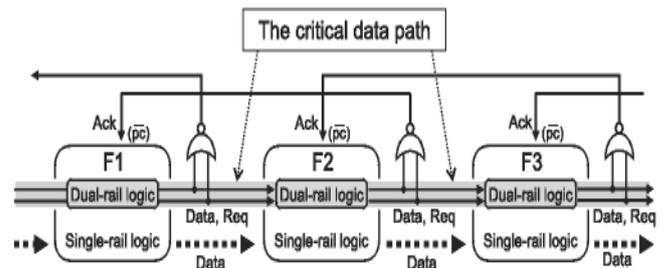


Fig. 6. Block diagram of AHDE

AHDE has the same protocol as PS0. The difference is that a total done signal is generated by detecting only the critical data path instead of the entire data paths. Such design method has two merits. First, the completion detector is simplified to a single NOR gate, and the detection overhead is not growing with the data path width. Second, the overhead of function block logic is reduced by applying single-rail logic in noncritical data paths. As a result, AHDE has a small overhead in both handshake control logic and function block logic, which greatly improves the throughput and power consumption.

Efficient charge recovery logic

Figure 7 shows the Efficient Charge Recovery Logic (ECRL) and was proposed by Moon and Jeong. It uses two cross-coupled PMOS transistors and two NMOS transistors in the N- functional blocks for the ECRL adiabatic logic block. Figure 7 ECRL Logic Circuit in figure pwr is the AC power supply is used for ECRL gates, so as to recover and reuse the supplied energy. Both out and /out are generated so that the

power clock generator can always drive a constant load capacitance independent of the input signal. Full output swing is obtained because of the cross-coupled PMOS transistors in both precharge and recover phases. But due to the threshold voltage of the PMOS transistors, the circuits suffer from the non-adiabatic loss both in the precharge and recover phases. That is, to say, ECRL always pumps charge on the output with a full swing. However, as the voltage on the supply clock approaches to $|V_{tp}|$, the PMOS transistor gets turned off. The ECRL circuits are operated in a pipelining style with the four-phase supply clocks. When the output is directly connected to the input of the next stage (which is a combinational logic), only one phase is enough for a logic value to propagate. However, when the output of a gate is fed back to the input, the supply clocks should be in phase. A latch is one of the simplest cases which have a feedback path. The input signals propagate to the next stage in a single phase, and the input values are stored in four phases (1-clock) safely.

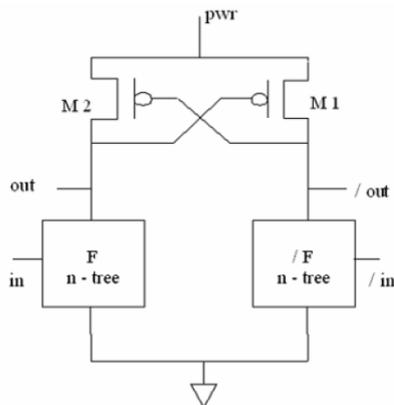


Fig. 7. ECRL

Working

Let us assume in is at high and inb is at low. At the beginning of a cycle, when the supply clock, pwr rises from zero to VDD, out remains at a ground level, because in turns on F- tree (NMOS logic tree). /out follows pwr through M1. When pwr reaches VDD, the outputs hold valid logic levels. These values are maintained during the hold phase and used as inputs for the evaluation of the next stage. After the hold phase, pwr falls down to a ground level, /out node returns its energy to pwr so that the delivered charge is recovered. Thus, the clock pwr acts as both a clock and power supply. For instance if we consider a circuit of a two input(x, y) AND gate with output(z).

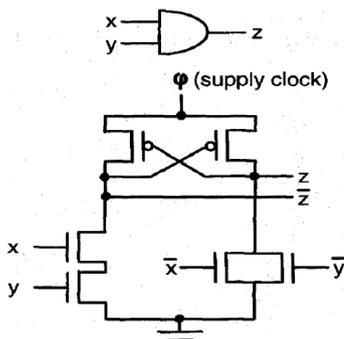


Fig. 8. ECRL AND gate

B. Structure of AHDE

Figure 9 shows the structure of AHDE. The solid arrow represents a constructed critical data path (dual-rail data path), the dotted arrow represents the noncritical data paths (single-rail data paths), and the dashed arrow represents the output of single-rail to dual-rail encoding converter.

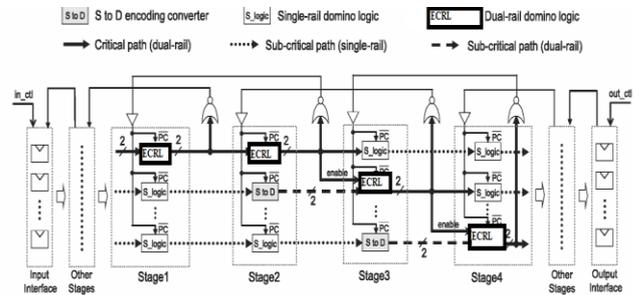


Fig. 9. Structure of AHDE

In each pipeline stage, a static NOR gate is used as 1-bit completion detector to generate a total done signal for the entire data paths by detecting the constructed critical data path. Driving buffers deliver each total done signal to the precharge/evaluation control port of the previous stage. Since the completion detector only detects the constructed critical data path, the noncritical data paths do not have to transfer encoded handshake signal anymore. Therefore, single-rail domino gates are used in the noncritical data path to save logic overhead. Encoding converter is used to bridge the connection between single-rail domino gate and dual-rail domino gate.

Encoding conversion

Since the completion detector detects only the constructed critical datapath, the noncritical data paths do not have to transfer encoded handshake signal anymore. The logic overhead in the noncritical data paths can be reduced using single-rail domino gates instead of dual-rail domino gates. However, single-rail domino gate and dual-rail domino gate use different encoding schemes. It has encoding compatibility problem when a single-rail domino gate connects to a dual-rail domino gate. Encoding converter needs to be designed to solve the problem.

Output waveform

A. Existing System

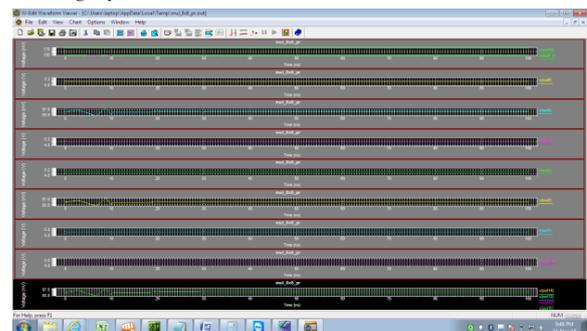


Fig. 10. AHPCDP output waveform

VI. CONCLUSION

This project introduced a novel design method of asynchronous domino logic pipeline. The pipeline is realized based on a constructed critical data path. The design method greatly reduces the overhead of handshake control logic as well as function block logic, which not only increases the pipeline throughput but also decreases the power consumption and no of transistor used. The evaluation results show that the proposed design has better performance than a hybrid-rail asynchronous domino logic pipeline design

REFERENCES

- [1] A. J. Martin and M. Nystrom, "Asynchronous techniques for system-on-chip design," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1089–1120, 2006.
- [2] B. Yasoda and S. Kaleem Basha, "Performance analysis of energy efficient and charge recovery adiabatic techniques for low power design," *IOSR Journal of Engineering (IOSRJEN)*, vol. 3, issue 6, pp. 14-21, 2013.
- [3] H. Saxena, Akansha, and V. Chaudhary, "Low power adiabatic logic circuits analysis," *International Journal of Advance Research in Science and Engineering*, vol. 2, issue 5, pp. 84-93, 2013.
- [4] J. S. Denker, "A review of adiabatic computing," in *Proceedings of the Symposium on Low Power Electronics*, pp. 94-97, 1994.
- [5] J. Spars and S. Furber, *Principles of Asynchronous Circuit Design: A Systems Perspective*, Boston, MA, USA: Kluwer, 2001.
- [6] J. M. Rabey and M. Pendram, *Low Power Design Methodologies:5-7*, 5th edition, Kluwer Academic Publishers, 2002
- [7] M. Pedram, "Power minimization in IC design: Principles and applications," *ACM Transactions on Design Automation of Electronic Systems*, vol. 1, issue 1, pp. 3-56, 1996.
- [8] M. Singh and S. M. Nowick, "The design of high-performance dynamic asynchronous pipelines: Lookahead style," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 11, pp. 1256–1269, 2007.
- [9] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd edition, New York: Addison - Wesley, 1993.
- [10] P. D. Khandekar, S. Subbaraman, and M. Patil, "Low power digital design using energy-recovery adiabatic logic," *International Journal of Engineering Research and Industrial Applications*, vol. 1, no. III, pp. 199-208, 2008.
- [11] S. M. Nowick and M. Singh, "High-performance asynchronous pipelines an overview," *IEEE Design & Test of Computers*, vol. 28, no. 5, pp. 8–22, 2011.
- [12] S. Samanta, "Adiabatic Computing a contemporary review," 4th *International Conference on Computer and Devices for Communication: Codec 09*, Kolkata, 2009.
- [13] T. Indermauer and M. Horowitz, "Evaluation of charge recovery circuits and adiabatic switching for low power design," *Technical Digest IEEE Symposium Low Power Electronics*, San Diego, pp. 102-103, 2002.
- [14] Y. Moon and D. K. Jeong, "An efficient charge recovery logic circuit," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 04, pp. 514-522, 1996.
- [15] Z. Xia, S. Ishihara, M. Hariyama, and M. Kameyama, "Design of high-performance asynchronous pipeline using synchronizing logic gates," *IEICE Transactions on Electronics*, vol. E95-C, no. 8, pp. 1434–1443, 2012.
- [16] Z. Xia, S. Ishihara, M. Hariyama, and M. Kameyama, "Dual-rail/single rail hybrid logic design for high-performance asynchronous circuit," *IEEE International Symposium on Circuits and Systems*, pp. 3017–3020, 2012.

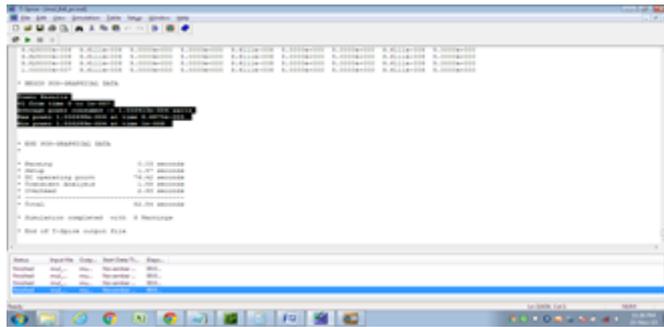


Fig. 11. AHPCDP power result

B. Proposed System

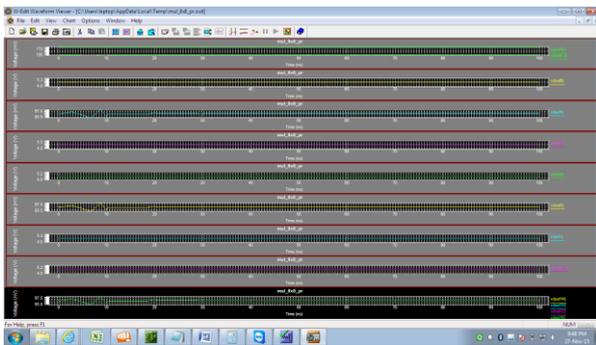


Fig. 12. AHDE output waveform

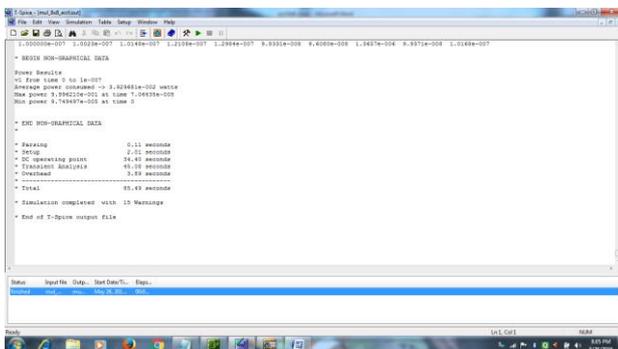


Fig. 13. AHDE power result

C. Comparison Table

TABLE I. Comparison table

	Exist ing 2X2	Propo sed 2X2	Exist ing 4X4	Propo sed 4X4	Exist ing 8X8	Propo sed 8X8
Power (mW)	18.9	16.8	25.9	23.5	41.5	39.2
Transistor count	122	101	634	606	1372	1312