Low Power Asynchronous Domino Logic Pipeline Design by Dual Rail Logic Gates

B. Pavithra, K. Priyanka, P. Venkateshwari
Ece, K. Ramakrishnan College of Technology
Trichy, India

Mr. R. .Ponnangan
Asst.Prof
Ece, K. Ramakrishnan College of Technology
Trichy, India

Abstract- **Asynchronous domino logic pipeline design is a latch less high throughput and low power design. In this design dual-rail domino gates are used to construct the stable critical data path and single-rail domino gates are used in non critical data paths. A 4 bit ripple carry adder is used to evaluate this pipeline design. This further saves a lot of power by reducing the overhead of logic circuits. The pipelined 8x8 bit multiplication by using 4x4 bit multiplication with full adder is used for evaluating the proposed pipeline method. Compared with a bundled-data asynchronous domino logic pipeline, now this can be implemented for higher order multipliers like 8x8.**

Keywords—Critical data path,dual-rail domino gate, single-rail domino gate

I. INTRODUCTION

DURING the last decade, there has been a revival in research on asynchronous technology. Along with the Continued CMOS technology scaling, VLSI systems become more and more complex. The physical design issues, such as global clock tree synthesis and top-level timing optimization, become serious problems. Even if technology scaling offers more integration possibilities, modularity and scalability are difficult to be realized at the physical level. Asynchronous design is considered as a promising solution for dealing with these issues that relate to the global clock, because it uses local handshake instead of externally supplied global clock The attractive properties are listed as follows

1) Low power consumption;
2) High operating speed;
3) No clock distribution and clock skew problems;
4) Better composability and modularity;
5) Less emission of electromagnetic noise;
6) Robustness toward variations in supply voltage, temper-ature, and fabrication process parameters.

In asynchronous design, the choice of handshake proto-cols affects the circuit implementation (area, speed, power, robustness, etc.). The four-phase bundled-data protocol and the four-phase dual-rail protocol are two popular protocols that are used in most practical asynchronous circuits. The four-phase bundled-data protocol design most closely resembles the design of synchronous circuits. Handshake circuits generate local clock pulses and use delay matching to indicate valid signal. It normally leads to the most efficient circuits due to the extensive use of timing assumptions. On the other hand, the four-phase dual-rail protocol design is implemented in an elaborate way that the handshake signal is combined with the dual-rail encoding of data. Handshake circuits are aware of the arrival of valid data by detecting the encoded handshake signal, which allows correct operation in the presence of arbitrary data path delays. This feature is very useful for dealing with data path delay variations in advanced VLSI systems, such as asynchronous field-programmable gate arrays (FPGAs) and system-on-chip .However, such attractive feature is realized at the expense of encoding and detection overheads. These overheads cause low circuit efficiency and restrict the application area of the four-phase dual-rail protocol design. Asynchronous domino logic pipeline is an interesting Pipeline style that can entirely avoid explicit storage elements Between stages by exploiting the implicit latching functionality of domino logic gates. The latch less feature provides the benefits of reduced critical delays, smaller silicon area, and lower power consumption.

II .EXISTING WORK

PS0 is a well-known implementation style of asynchronous domino logic pipeline based on dual-rail protocol [8]. It is an important foundation for most later proposed styles. Since our proposed pipeline is also based on PS0, we will begin by reviewing PS0 pipeline style, and then simply introducing two other advanced styles: 1) a timing-robust style called precharge half-buffer and 2) a high-throughput style called lookahead pipeline. Finally, we summary the delay assumptions of these pipelines and give our delay assumption in the proposed design. A.There are mainly two overheadproblems that prohibit the widespread use of PS0, the detection overhead in handshake control logic and the dual-rail encoding overhead in function block logic. A ripple carry adder, shownaintaining the Integrity of the SpecificationsThe detection overhead is caused by the full completion detectors that are used to deal with data path delay variations by detecting the entire data paths. The overhead greatly affects the pipeline speed and power consumption. The most serious problem is that the detection overhead is growing the width of data paths. In a 4-bit ripple carry adder, the width of datapaths is between 8 and 5 bits. The detection overheads of8–5-bit completion detectors might be acceptable in practicaldesign. However, in 32-bit ripple carry adder design, the widthof data paths is at least 33 bits. The overhead of 33-bitcompletion detector is so large that PS0 is hardly applicablein such situation. Even the detection time can be reduced bypartitioning wide data path into several data streams, the detection power is not reduced.The dual-rail encoding overhead is caused by dual-rail domino logic that is used for not only implementing logicfunction but also storing data between pipeline stages. Becausethere are no explicit storage elements (latches or registers),a lot of dual-rail domino buffers have to be added to levelizeeach stage. The added dual-rail domino buffers consume alot of silicon area and power. In a 4-bit ripple carry adder,18 dual-rail domino buffer gates are added, which almost cancel out the benefit of removing explicit storage elements. The pipeline is designed based on astable critical data path that is constructed using a special dual-rail logic. The critical data path transfers a data signal and encoded handshake signal. Noncritical data paths, composedof single-rail logic, only transfer data signal. A staticNORgate detects the dual-rail critical data path and generates atotal done signal for each pipeline stage. The outputs of NORgates are connected to the precharge ports of their previousstages.APCDP has the same protocol as PS0. The difference is that a total done signal is generated by detecting onlythe critical data path instead of the entire data paths. Such design method has two merits. First, the completion detector is simplified to a single NORgate, 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, APDCP has a small overhead in both handshake control logic and function block logic, which greatly improves the throughput and power consumption.

III. PROPOSED WORK

Based on the proposed construction method, design automation flow can be easily developed. First, the gate with the largest number of inputs in each stage and connection of gates between stages is identified. Then, the proper gates using SLGs or SLGLs are replaced according to the gate connection information and optimization methods. Another issue is the design automation for layout. Standard placement and routing (P&R) tools tend to reduce the work case delay in function blocks by optimizing circuit layout. Such optimization is not suitable for APCDP since it would degrade the robustness of the constructed critical data path. To avoid this problem, a specific P&R method has to be developed to increase the delay on the constructed path. The last issue is timing verification. There is almost no EDA support for verifying domino circuits because charge sharing.

Block Diagram:

The process is proposed in 4*4 multiplier using 8X8 multiplier, when it is blocks

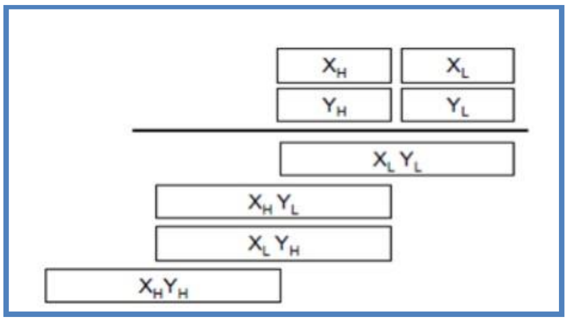


Fig (1)

In this case, the numbers x and y are first spilt into 2 parts as XH-XL and YH-YL. The products XL*YL, XH*YL, XL*YH and XH*YH are then calculated in parallel using 4 array multipliers. The products are then added to get the final product P15 – P0

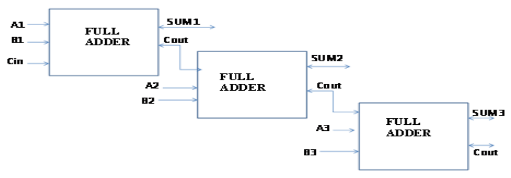


Fig (2)

SINGLE RAIL DATA PATHS:

Two distinct solutions are now proposed for gate-level pipelining of wide single-rail data paths. The first approach is again to use partitioning. In this case, the dual-rail completion detectors are replaced with single-rail completion generators and bundling signals. However, there is one important difference: in single-rail bundled data paths (unlike dual-rail), each data stream has its own distinct “request” signal. Therefore, the merging of multiple data streams now requires the explicit combining of the multiple request signals at the stage’s inputs. This merger is achieved by a simple modification of the stage completion generator: additional request signals are typically easily accommodated by adding extra transistors in series.

DUAL RAIL DOMINO GATE:

However, asynchronous domino logic pipeline has a common problem that dual-rail domino logic has to be used to create the domino data path. Single-rail domino logic cannot be used because it would break the domino data path hence only non-inverting logic can be implemented. As a result, the domino data path has a dual-rail encoding overhead that it consumes more silicon area and power consumption. Such overhead almost cancels out the area and power benefits provided by the latch less feature.

Another problem is overhead of handshake control logic. Conventional designs of asynchronous domino logic pipeline is based on four-phase dual-rail protocol rely on 704 A. domino data path to transfer the data and encoded handshake signal and it is used to detect the completion detectors and collect the handshake signal throughout the entire data paths.

COMPARISON TO SYNCHRONOUS PIPELINE TO BUNDLED DATA ASYNCHRONOUS PIPELINE:

In order to further show the benefits of APCCD, which is named as APCCDPS0 in previous section, we choose array style multiplier as a test case to compare to synchronous pipeline and bundled-data asynchronous pipeline.8×8 multipliers are respectively fine-grain pipelined using these three design methods. Synchronous pipeline with a se-quintal clock-gating [17]–[19] is also designed to com-pare to APCCD. Table 5 shows the comparison results. Be-cause dynamic logic in APCCD provides an implicit latch function, storage elements are all removed which respect.

The performance of power consumption (8×8multiplier, 2.86G data-set/s).It saves 278 flip-flops and 791 latches compared to synchronous pipeline And bundled data asynchronous line Synchronous pipeline with clock-gating design need extra flip-flops to implement the clock gating control circuit, which slightly deteriorates throughput and forward latency the performance of power consumption when all pipelined circuits work at 2.86G data-set/s.

The workload refers to the rate of the number of active-state cycles to the total number of cycles. In our case, the workload is calculated based on a period of consecutive data injection cycles (active-state cycles) following consecutive empty cycles, which is the workload is calculated as N/(N+M), where N is the number of consecutive data injection cycles admits the number of consecutive empty cycles. Previous works all mention that bundled-data asynchronous design shows better power performance than their synchronous counterparts.

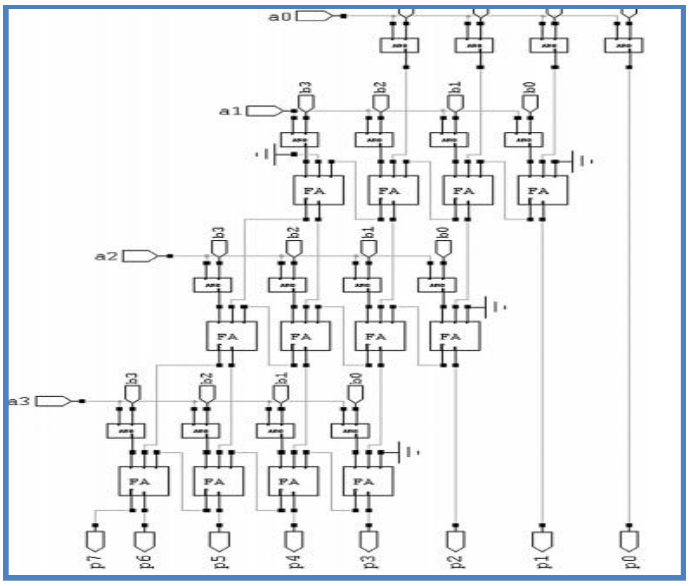
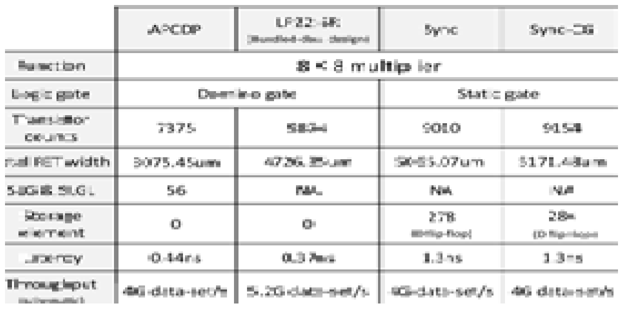


Fig (3)
The fig 3 shows the 4 × 4 multiplier we have to give a pulse signal. After that we give inputs such as a0, a1, a2, and a3,b0, b1, b2 and b3.

Table3.1

Evalution resuts of 8 X 8 array style multipliers



	APCDP	LP22-SR (Bundled-data design)	Sync	Sync-CG
Function	8 × 8 multiplier			
Logic gate	Domino gate		Static gate	
Transistor counts	7375	5834	9010	9154
Total FET width	3075.45um	4726.35um	5055.07um	5171.48um
[illegible]	56	NA	NA	NA
Storage element	0	0	278 (flip-flop)	284 (D flip-flops)
Latency	0.44ns	0.37ns	1.3ns	1.3ns
Throughput [illegible]	4G data-set/s	5.2G data-set/s	4G data-set/s	4G data-set/s

Every output data-set is triggered by an ack signal from the receiver. When the supply voltage is changed from 1.2 V to 0.75 V, all computation results have been verified to be correct. To a certain degree, it demonstrates the robustness of APCCD. The simulation results show that the post-layout multiplier works well at 2.16 GHz. A unique feature of many asynchronous dynamic pipelines is that they are latchless. Thus, with proper sequencing of control operations, an asynchronous pipeline can exploit the implicit latching functionality of dynamic gates and entirely avoid explicit storage elements between stages. Achieving similar latchless operation in a synchronous implementation typically would require using complex multiphase clocking. This asynchronous feature provides the benefits of reduced critical delays, smaller chip area, and lower power consumption, thereby minimizing some of the key overheads of fine-grained pipelining.

IV.SOFTWARE DESCRIPTION

MICROWIND:

Microwind is a tool for designing and simulating circuits at layout level. The tool features full editing facilities (copy, cut, past, duplicate, move), various views (MOS characteristics, 2D cross section, 3D process viewer), and an analog simulator. The microwind software allows the designer to simulate and design an integrated circuit at physical description level. Born in Toulouse (France). Micro wind is an innovative CMOS design tool for educational market.

Microwind is developed as comprehensive package on windows platform to enable students to learn smart design methods and techniques with more practice. With inbuilt layout editing tools, mix-signal simulator, MOS characteristic viewer and more, it allows students to learn complete design process with ease. Microwind unifies schematic entry, pattern based simulator, SPICE extraction of schematic, Verilog extractor, layout compilation, on layout mix-signal circuit simulation, cross sectional & 3D viewer, netlist extraction, BSIM4 tutorial on MOS devices and sign-off correlation to deliver unmatched design performance and productivity. With its approach for CMOS design education, Microwind has gained lot followers worldwide. Universities across the globe are using Microwind tool for budding engineers to teach CMOS concepts with ease. Paving their path for more skilled softwares to be used at later stage of their course work.

TOOLS FROM MICROWIND:

- Microwind
- DSCH
- Microwind3 Editor
- Microwind 2D viewer
- Microwind 3D viewer
- Microwind analog simulator
- Microwind tutorial on MOS devices

View of Silicon Atoms Using softwares for microwind and DSCH

Microwind window:

MICROWIND supports entire front-end to back-end design flow. For front-end designing, we have DSCH (digital schematic editor) which posses in-built pattern based simulator for digital circuits.

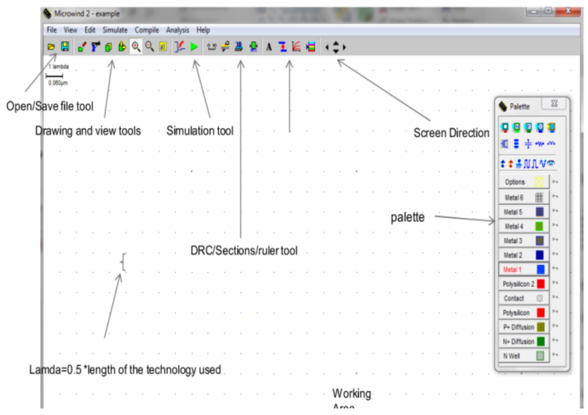


Fig (4)

User can also build analog circuits and convert them into SPICE files and use 3rd party simulators like WinSpice or pSPICE. DSCH can convert the digital circuits into Verilog file which can be further synthesized for FPGA/CPLD devices of any vendor. The same Verilog file can be compiled for layout conversion in MICROWIND.

The back-end design of circuits is supported by MICROWIND. User can design digital circuits and compile here using Verilog file. MICROWIND automatically generates a error free CMOS layout. Although this place-route is not optimized enough as we do not indulge in complex place & route algorithms. User can also create CMOS layout of their own using compile one line Verilog syntax or custom build the layouts by manual drawing.

Advantage:

- Produce system with longer and effective operational life.
- Produce system that more closely meet user needs and requirements.
- Produce system with excellent documentation.
- Produce system that need less system support.
- Produce more flexibility.

Disadvantage:

- Produce initial system that are more expensive to build and maintain
- Require more extensive and accurate definition of users needs and requirements
- May be difficult to customize
- Require training of maintenance staff.
- May be difficult to use with existing system
.

V. RESULT & IMPLEMENTATION

Full adder:

The full-adder circuit adds three one-bit binary numbers (A, B, Cin) and outputs two one-bit binary numbers, a sum (S) and a carry (Count).

Ripple carry adder:

It is possible to create logical circuit using multiple full adders to add N (pre case 16) bit numbers. Each full adder inputs a Cin (Carry input) which is the Cout (Carry output) of previous adder. This kind of adder is ripple carry adder since each carry bit ripples to the next full adder.

4 bit multiplier using AND logic circuits:

AND logic circuit:

CMOS is also sometimes referred to as complementary-symmetry metal–oxide–semiconductor (or COS-MOS). The words "complementary-symmetry" refer to the fact that the typical digital design style with CMOS uses complementary and symmetrical pairs of p-type and n-type metal oxide semiconductor field effect transistors (MOSFETs) for logic function.

The layout design and output waveform of 2 input AND gate is obtained after designing as respectively. From the output waveform of CMOS AND gate the waveform the truth table .i.e. for any of the low input of AND gate output is low.

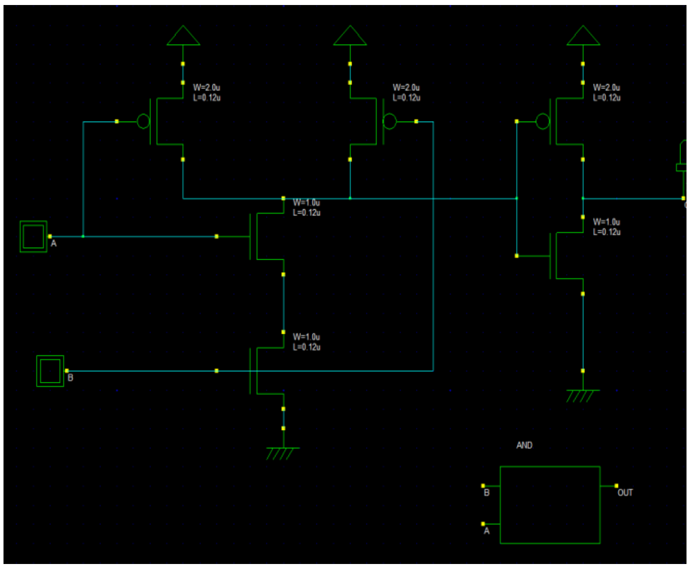
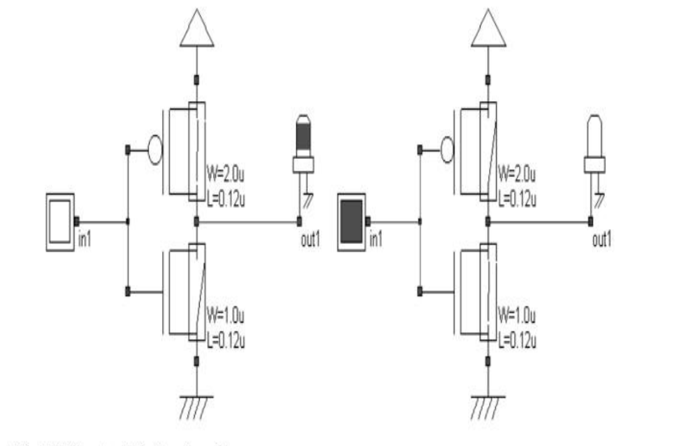


Fig (5): AND LOGIC CIRCUIT

The CMOS inverter:

The CMOS inverter design is detailed in the figure below. Here the p-channel MOS and the n-channel MOS transistors function as switches. When the input signal is logic 0 (Fig. 4-3 left), the nMOS is switched off while PMOS passes VDD through the output. When the input signal is logic 1 (Fig. 4-3 right), the pMOS is switched off while the nMOS passes VSS to the output.



': The MOS Inverter (File CmosInv.sch)

Fig (5)
Full adder circuit design:

Full adder:

The full-adder circuit adds three one-bit binary numbers (A ,B, Cin) and outputs two one-bit binary numbers, a sum (S) and carry (Count). The FULL ADDER (FA for short) circuit can be represented in a way that hides its innerworkings: The FULL ADDER can then be assembled into a cascade of full adders to add two binary numbers. For example the diagram below shows how one could add two 4-bit binary numbers X3X2X1X0 and Y3Y2Y1Y0 to obtain the sum S3S2S1S0 with a final carry-out C4.

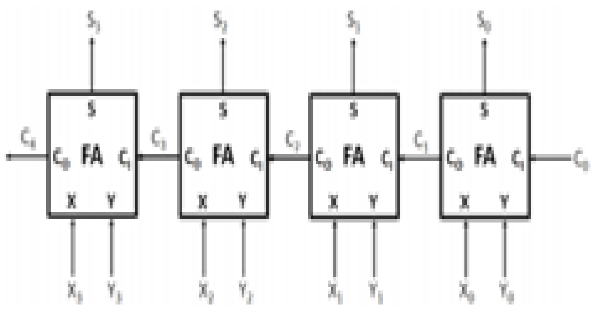
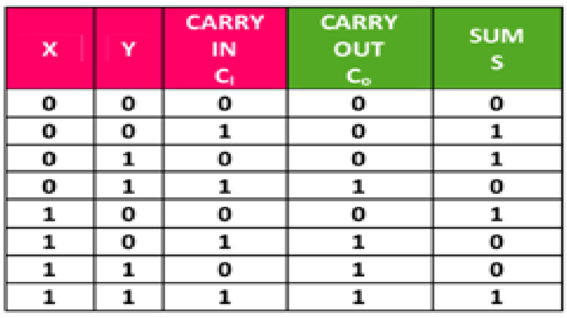


Fig (6)

This circuit is commonly called a ripple-carry adder since the carry bit ripples from one FA to the next FA on its left. This rippling cause this adder design to be somewhat slow when compared to other more complex designs. The initial carry-in can be hard-wired to zero, making it unnecessary for a half adder on the far right.

Truth table: 3.2 for full adder



X	Y	CARRY IN C_i	CARRY OUT C_o	SUM S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

It is a mathematical operation that at its simplest is an abbreviated process of adding an integer to itself a specified number of times. The operation of multiplication is rather simple in digital electronics. It has its origin from the classical algorithm for the product of two binary numbers.

This algorithm uses addition and shift left operations to calculate the product of two numbers .Based upon the above procedure, we can deduce an algorithm for any kind of multiplication. We can check at the initial stage also that whether the product will be positive or negative or after getting the whole result, MSB of the results tells the sign of the product. Digital multiplication is a series of bit shifts and bit additions, where two numbers, the multiplicand and the multiplier are combined into the result. Considering the bit representations of the multiplicandX0X1 X2……Xn-1 and the multiplierY0Y1Y2….Yn-1, in order to form the product, up to n shifted copies of the multiplicand is to be added for unsigned multiplication. The entire process consists of three steps, partial product generation, partial product reduction and final addition

Example:

Even though the result comes from the combination of all operations, a certain amount of independence exists between each operation, considering the addition on a particular column. All the bits in a column must be added together along with the carry in bits coming from the previous column.

Carry save addition implies that addition in separate columns can be performed independently without concerning about carry from previous column. There are several ways to implement addition of partial products in the trapezoidal array. The waveform may be vary according to pulse signal we give i.e. in ns .here we take two pulse signal such as 500ns and 100ns.

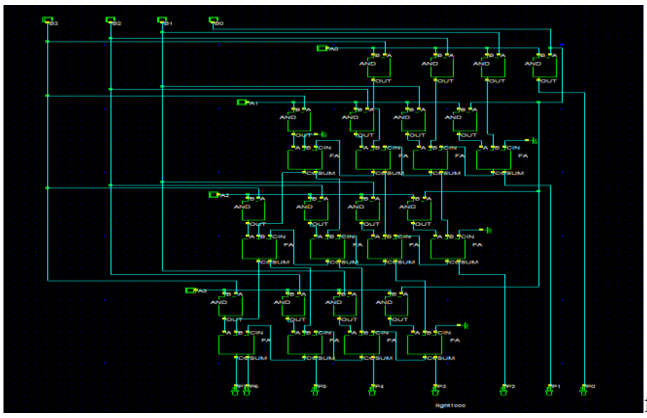
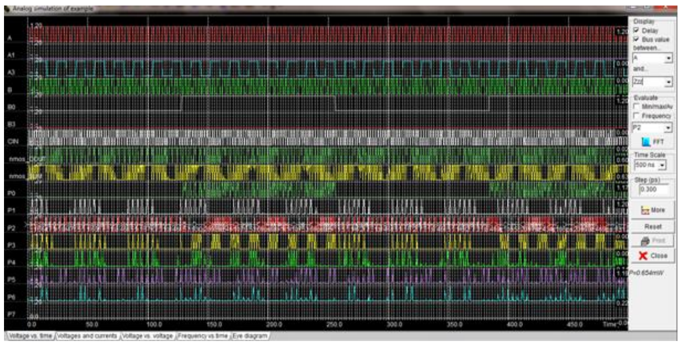
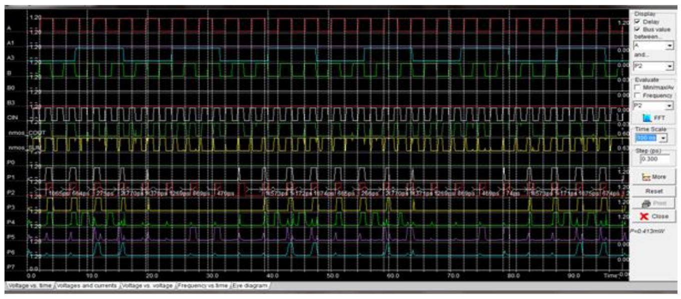


fig (7)
Circuit diagram for 4×4 multiplier

SIMULATION AND RESULT:



Simulation results of 4X4 multiplier by using 500ns.



Simulation results of 4x4 multiplier by using 100ns.

The waveform may be vary according to pulse signal we give i.e. in ns .here we take two pulse signal such as 500ns and 100ns.

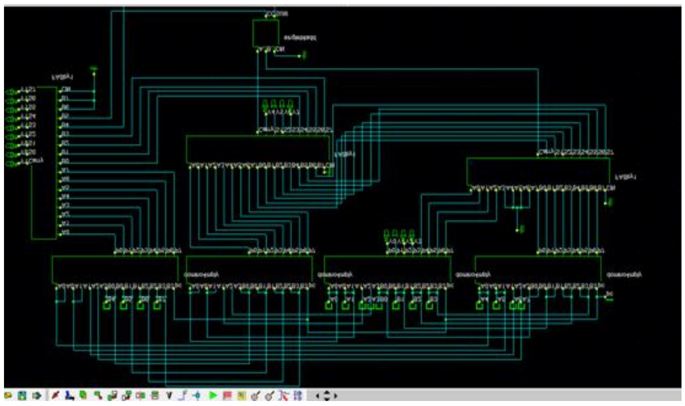
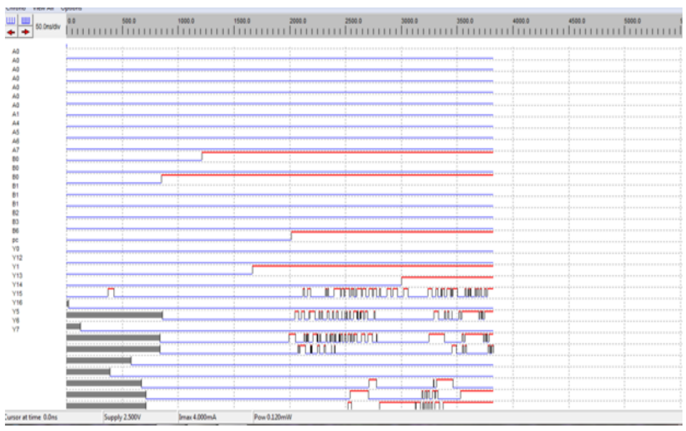
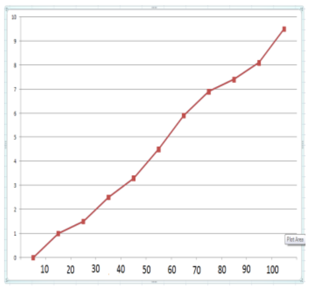
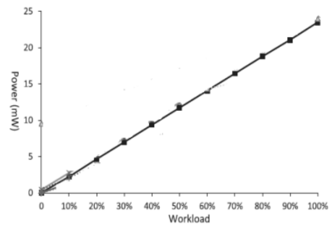


Fig (8) Circuit diagram for 8x8 multiplier



Fig(9) Waveform for 8x8 Multiplier Circuit

CONCLUSION:

Power consumption, speed, accuracy, hardware requirements, chip area are some the grave concerns in the VLSI industry today .Reduction in the power consumption and delay of a multiplier circuitry is expected to cause a revolution in the field of electronics and communication as these circuits are widely used in every digital and analog system including computers, calculators and other consumer electronics commodities. In this paper the proposed 8x8 multiplier design using CMOS logic shows up to 70% power efficiency at 50MHz and 65% power efficiency at 100MHz in multiplier design using CMOS logic. Thus any system that incorporates this novel design of multiplier will help to curb the power consumption and area of the systems.