DESIGN OF LOW POWER AND HIGH SPEED ADDER

DISSERTATION II

Submitted in partial fulfillment of the

Requirement for the award of the

Degree of

MASTER OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted By

S SHIVA KESHAV

Under the guidance of

Dr. RAVI SHANKAR MISHRA

(School of Electronics and Electrical Engineering)

Lovely Professional University

Phagwara, Punjab

April 2017

INDIA'S LARGEST UNIVERSITY*
LOVELY PROFESSIONAL UNIVERSITY
Transforming Education, Transforming India

TOPIC APPROVAL PERFORMA

School of Electronics and Electrical Engineering

Program : P175::M.Tech. (Electronics and Communication Engineering) [Full Time]

COURSE CODE : ECE521 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** EEERGD0257

Supervisor Name : Dr. Ravi Shankar Mishra **UID :** 19053 **Designation :** Associate Professor

Qualification : ________________________ **Research Experience :** ________________________

SR.NO.	**NAME OF STUDENT**	**REGISTRATION NO**	**BATCH**	**SECTION**	**CONTACT NUMBER**
1	Shakkari Shiva Keshav	11512441	2015	E1514	8699659202

SPECIALIZATION AREA : VLSI Design **Supervisor Signature:** ____________________

PROPOSED TOPIC : Design of Low power and high speed adder

Qualitative Assessment of Proposed Topic by PAC

Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	8.00
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.67
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	8.00
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.67
5	Social Applicability: Project work intends to solve a practical problem.	8.00
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	8.00

PAC Committee Members

PAC Member 1 Name: Anshul Mahajan	UID: 11495	Recommended (Y/N): Yes
PAC Member 2 Name: Dushyant Kumar Singh	UID: 13367	Recommended (Y/N): NA
PAC Member 3 Name: Cherry Bhargava	UID: 12047	Recommended (Y/N): Yes
PAC Member 4 Name: Anshul Mahajan	UID: 11495	Recommended (Y/N): Yes
DAA Nominee Name: Manie Kansal	UID: 15692	Recommended (Y/N): NA

Final Topic Approved by PAC: **Design of Low power and high speed adder**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 11211::Prof. Bhupinder Verma **Approval Date:** 08 Oct 2016

4/27/2017 2:18:59 PM

DECLARATION

I student of M. TECH under ECE Discipline at LOVELY PROFESSIONAL UNIVERSITY , Phagwara; hereby declare that the Dissertation-II report entitled on **“Design Low Power High Speed Adder”,** is an authentic record of my own work carried out as the requirements for the award of degree of Master of Technology on “Electronic and Communication” at Lovely Professional University, Phagwara; under the guidance of “**Dr. Ravi Shankar Mishra**”, an Associate professor of Deportment of Electronics and Communication Engineering, during January to April,2017.

S Shiva Keshav
Reg.no:11512441
Date:

It is certified that the above statement is correct to the best of my knowledge and belief.

Dr. Ravi Shankar Mishra

Associate Professor
Lovely Professional University
Phagwara-144411, Punjab.
Date**:**

CERTIFICATE

This is to certify that the Dissertation II entitled **"Design of Low power and high speed Adder"** is a record of bonafied work carried out by **S Shiva Keshav** under my guidance and supervision for the partial fulfillment of the degree of Master of Technology in Electronics and Communication Engineering during the academic session January 2017 – April 2017 at Lovely Professional University – Phagwara.

To the best of my knowledge, the results embodied in this Dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

Dissertation Supervisor
Dr. Ravi Shankar Mishra
Associate Professor
Department of ECE
Lovely Professional University
Phagwara
.

ACKNOWLEDGEMENT

I take this opportunity to acknowledge the co-operation, good will and both moral and technical support extended by several individuals out of which this Dissertation II has evolved. I always cherish my association with them.

I express my sincere and deepest regards to my supervisor **Dr. Ravi Shankar Mishra** for his valued guidance during this period of my Dissertation. This Dissertation work was enabled and sustained by his vision and ideas. His scholarly guidance and invaluable suggestions motivated me to complete my Dissertation work successfully.

I owe a great many thanks to **Dr. Ravi Shankar Mishra** for spending his valuable time. I considered my-self extremely fortunate to have this opportunity of associating with him.

I express my sincere thanks to the head of the department (ECE) and the members of Department of Electronics and Communication Engineering, Lovely Professional University for their cooperation.

I would like to express my deep gratitude to my parents. Their continuous love and support gave me strength for pursuing my dream. Last but not the least; I am thankful to my friends who have been a source of encouragement and inspiration throughout the duration of this Dissertation.

LIST OF CONTENTS

Title **Page No.**

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Adders are commonly found in the critical path of many building blocks of the microprocessors and also digital signal processing chips. Adders are essential for not only addition, but also essential for subtraction, multiplication, and division. Addition is one of the fundamental arithmetic operations. A fast and accurate operation of the digital system is mostly influenced by the performance of the respective adders. The most important constraints of the adder designs are the area, power, speed, timing, number of cells and delay.

As day by day digitalization is increasing in every application and so that the speed of processors are increasing. So, to fulfil the demand of digitalization we need adders which provide us accurate outputs with very less consumption of the constraints given above.

In this we have validate different types of adders like ripple carry adder, carry look ahead adder, carry select, carry skip and carry save on the bases of area, power and delay. The simulation and synthesis is done on industry standard tool Cadence on which we have worked on different technologies like 45nm, 90nm, 180nm.

Depending upon the parametric analysis these fast adders have utilized in Multiplier Accumulator Unit (MAC) in DSP application.

CHAPTER 1 INTRODUCTION

Adders are the vital components of the CPU (central processing unit) in today's Digital world. They are used in floating point calculations, ALUs, also in order to compute addresses of memory. In other applications like microprocessors and digital signal processors (DSP) architectures full - adders plays important role. The real revolution came into existence when reduction of operating voltage and continuous scaling of the transistor size has led to a predominant enhancement of the integrated circuits(IC). They play important role in the packaging of FPGA device and fast performing devices with low power consumption, high speed and smaller area. In Digital Signal Processing (DSP) and Central Processing Unit (CPU) adders are the most normally used arithmetic block, hence optimizing power is of most importance. Speed of a circuit increases rapidly with scaling technology to the depth of sub-micron and also power consumption increases significantly per chip with respect to increase in the density of the chip. Further, High-speed and low power are the two important factors that needs to be considered in realizing modern (VLSI) Very Large Scale Integration circuits. In case of circuits' design, the low-power adders with high-performance can be given at various different levels, such as in the process technology, the logic style, architecture and layout,.

The composition of large number of single-bit full adders becomes ripple carry adder. The architecture of the carry ripple adder circuit is easy. but speed of the circuit is slow because every adder start's operating only when previous output carry signal becomes ready. More complex than ripple carry adder is that which consumes high power but high speed in operating like the carry skip adder, carry look- ahead adder, carry select adder and carry increment adder.

What actually is an adder?

The adder is a circuit which performs the summation of two given inputs termed in digital electronics. In order to perform any operation this is the basic circuit. The adder's are not only used in the different parts of the processor but also used as the part of the ALU(s), where they is need to table lists, compute addresses, and many more.

The operation of an adder is carried out like : 0+0=0 0+1=1 1+0=1 1+1=10

Different types of adders:

They are of two type of adders:

- Full Adder
- Half Adder

1.1 Half Adder:

It is the one which performs summation of two single individual binary bits' x and y is an Half adder. It will generate two outputs, carry [C] and sum [S]. When two multiple bits are added the excess which goes into the next bit is the carry signal. For ex, in the above case when both inputs are entered as 1 then it resulted in 10 i.e., here 1 get's shifted to next bit the sum would be 0 and hence carry would be only 1. The pictured below is the normal half-adder design, in which for generating the sum we use xor gate and for generating carry we use and gate. The half adder will generate output of a carry and sum by adding two single digit input.

The half adder circuit is as follows:

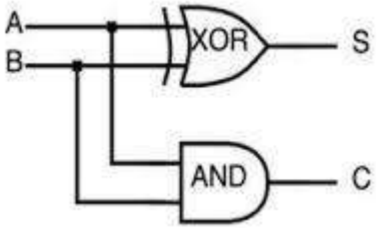


Fig 1. Half-Adder

Table 1. The truth table of the Half-Adder:

Input	X	0	0	1	1
	Y	0	1	0	1
Output	Sum	0	1	1	0
	Carry	0	0	0	1

1.2 Full Adder:

The one which sums the binary digits and notes the digits that are carried in and also out is known as full adder. In case of full adder the summation operation is done for three one-digit numbers, which are written as X, Y, and Cin. X and Y can be called as operands, and input Cin is a bit which is sent from previous digit. Which actually generates a two bits as output, which are sum and carry that are given as the Cout that is carry out and sum signal output as S.

The full adder circuit is as follows:

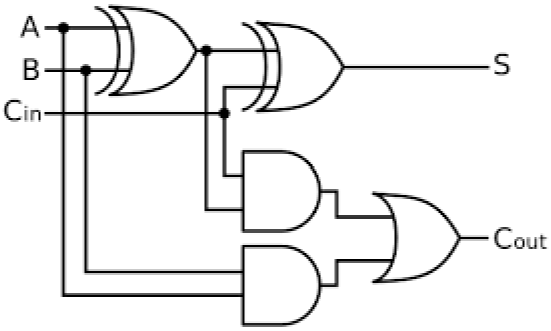


Fig 2. Full-Adder

It is also done with the help of two half adders to which A and B as input of one half adder, and output is the sum of it is given as an input to the next half adder, and Cin is provided as second input to same half adder and or the two carrys which results a Full Adder. The circuit using half adder is as shown below.

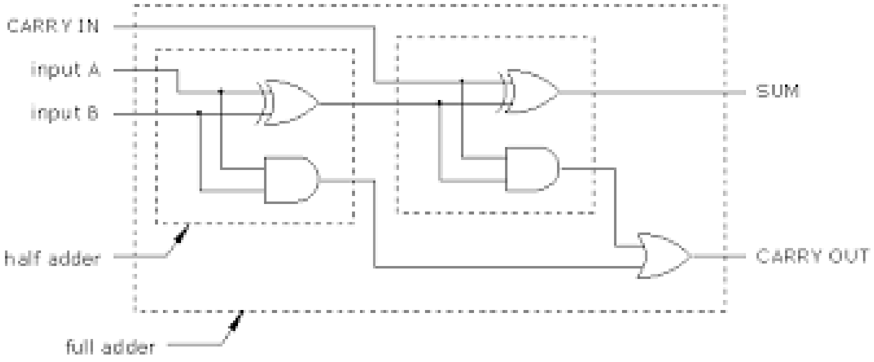


Fig 3. Full-Adder using Half-Adder

Table 2. The truth table for the full adder is:

Inputs	A	0	0	0	0	1	1	1	1
	B	0	0	1	1	0	0	1	1
	Carry in	0	1	0	1	0	1	0	1
Outputs	Sum	0	1	1	0	1	0	0	1
	Carry out	0	0	0	1	0	1	1	1

It usually a part of complex adders, that add 8, 16, 32, etc. binary digits. Many different types of complex adder's can be implemented by using these Full adders and Half adders.

Complex Adders are the one that are combination of full adders and half adders. Many complex adders are out in our digital world out of which some of them are given below:

- Ripple Carry Adder
- Carry Look Ahead Adder
- Carry Select Adder
- Carry Skip Adder
- Carry Save Adder

2.1 Carry Ripple Adder:

As word specifies this adder is the one that ripples the carry to next stage which is sent from the previous bit. It is combination of n full adders. So, the full adder output carry would be provided to the preceding full adder as input. Drawback of this adder is that it must wait for all the previous carry to generate the final carry. The circuit of the carry ripple adder as given below

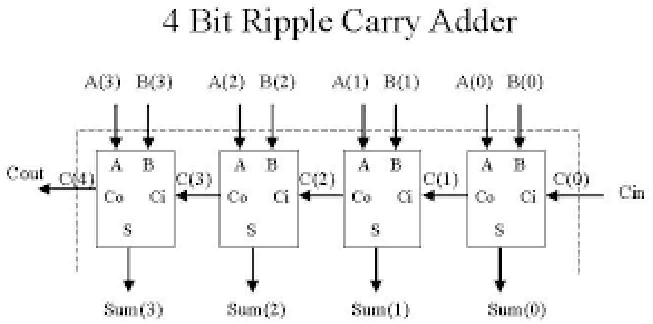


Fig 4. Carry Ripple Adder

Operation of carry ripple adder:

In the above example we are considering 4bit Ripple carry adder. In which we require 4 full adders. Let inputs are A, B then first bit of A and B are given to first full adder, and third input for the first full adder is Cin. Later carry and sum is generated. Further Carry out of first full adder is provided as input bit to second full adder and second bit of inputs as inputs to the second full adder then the outputs carry and sum have been processed and it goes on till n bits.

2.2 Carry look ahead adder:

It is another complex adder. Which uses concept of generate the carry and propagate the carry. However, the concept of operation of generating carry and propagating carry is as the following.

For carry generate we use G = A * B.

For carry propagate we use P = A xor B.

The circuit diagram for carry look a-head adder is given below:

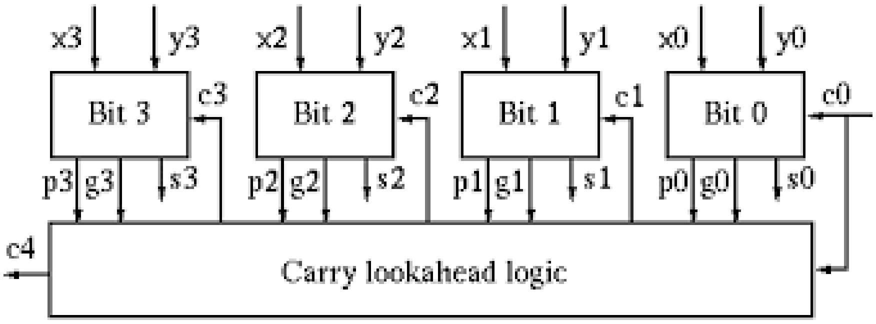


Fig 5. Carry Look Ahead Adder

Procedure of carry look ahead adder:

To generate carry propagate and carry generate, the addition digit is carried out then the summation would be generated else the least significant bit(LSB) carries and summation is propagated. If the expression is represented, as C_i is carry in digit of the bit *i*, then P_i is propogate, G_i is the generate bit of bit *i*,

$$C_{i+1} = G_i + (P_i * C_i)$$

The summation of every digit of the sequence, the Carry Look Ahead Logic shall provide either the digit pair will propagate the carry or generate the carry. Which will allow the operation to "pre-process" the summation of input digit pairs which provide the time of carry ahead. Further, when original summation is processed, then there would be no delay for the carry ripple effect. Above given is generalized 4-bit Carry Look Ahead block which is combination of 4-bit Carry ripple adder with the few adjustments:

For sample given, logic of carry generates (g) and carry propagates (p) equations are provided following. Out of which numbers in below equations are the input signals, initiating from 0 least significant bit to 3 most significant bit:

$C_1 = G_0 + (P_0 * C_0)$

$C_2 = G_1 + (P_1 * C_1)$

$C_3 = G_2 + (P_2 * C_2)$

$C_4 = G_3 + (P_3 * C_3)$

Further keeping C1 in C2, then C2 in C3 and C3 in C4 we get

$C_1 = G_0 + (P_0 * C_0)$

$C_2 = G_1 + (P_1 * G_0 + (P_1 * P_0 * C_0))$

$C_3 = G_2 + (P_2 * G_1 + (P_2 * P_1 * G_0 + (P_2 * P_1 * P_0 * C_0)))$

$C_4 = G_3 + (P_3 * G_2 + (P_3 * P_2 * G_1 + (P_3 * P_2 * P_1 * G_0 + (P_3 * P_2 * P_1 * P_0 * C_0))))$

2.3 Carry Select adder:

It is another complex adder which is a conditional sum adder. It is constructed with sharing of logic values commonly for sum generation. For sharing them commonly, a inverter with a xor gate are used to perform addition outputs which are given below figure. When input carry-in is be ready, then it can generate accurate sum depending on the input carry-in signal. Further for the output carry, It should use one and gate & one or gate to find out possible input carry values before generating. Finally when the carry-in bit would be ready, it can find out the accurate output carry out based on the input carry-in bit.

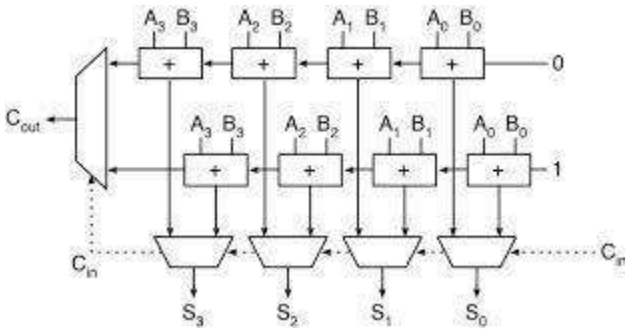


Fig 6. Carry Select Adder

Operation of Carry select Adder:

It uses a pair of ripple carry adders & a single multiplexer. Summation of the pair of n-digit numbers of it would be performed using the both adders in order to do the operation two times, once by taking the carry in (Cin) as one and the other time taking the carry in (Cin) as zero. When both the results are evaluated, then the correct carry out and also the actual sum is generated with the help of multiplexer once the original carry in is known.

2.4 Carry Skip adder:

It is also known as Carry Bypass Adder which is another type of complex adder. The operation of this adder has reduced the delay over ripple carry adder than the other adder. The enhancement of least delay is done using many carry skip adders combine to form a single carry skip-adder block. It has created to increase the operation through summing produced carry digit to the whole adder. The resultant circuit [1] is given in following figure for a 4bit adder. The input carry-in digit is provided as Ci and further the adder produces the output carry-out digit (Ci+4) by itself. Carry skip circuit utilizes a pair of logic gates. The (and) gate is utilized to use the carry-in then finally compares with the propagate signal group.

$$P_{(i,i+3)} = P_{i+3} * P_{i+2} * P_{i+1} * P_i$$

With help of above results, the generated output of (and) gate provided as input to (or) gate with Ci+4 to generate final Carry out.

$$\text{Carry} = C_{i+4} + P_{(i,i+3)} * C_i$$

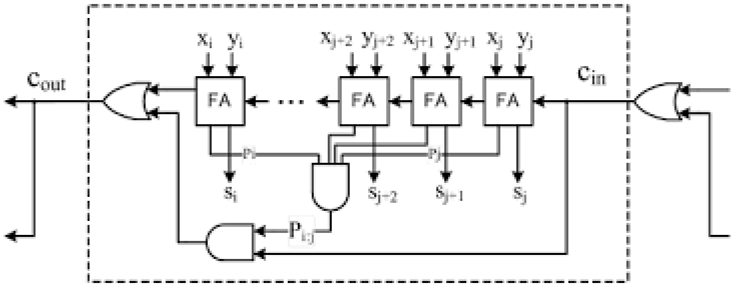


Fig 7. Carry Skip Adder

Operation of carry skip adder:

From the above circuit, when P(i, i+3) = 0, which results in output group carry out is controlled using (Ci+4) value. Further, when P(i,i+3)’ = 1, further input carry digit is Ci. = 1, then input carry-in block would be consequently sent to the following adders group. This adder’s named as “carry skip adder” due to the P(i,i+3)*Ci condition. If it is valid and then it skips the entire block.

2.5 Carry Save adder:

It is the one with low spread delay [4] (basic way), however it either of summing both input bits to an individual output sum, it sums the three input bits to a two output bits. At last, then its outputs are added to a conventional carry ripple adder or carry look ahead, which will generate the entire inputs output as sum.

While summing at least three bits together, the sequence of carry-save adders [2] is been ended by a individual carry-look ahead adder that gives much preferable propagation delays over the sequence of carry look a-head adder. Specifically, it’s spread delay is not influenced by the width of the bits being included.

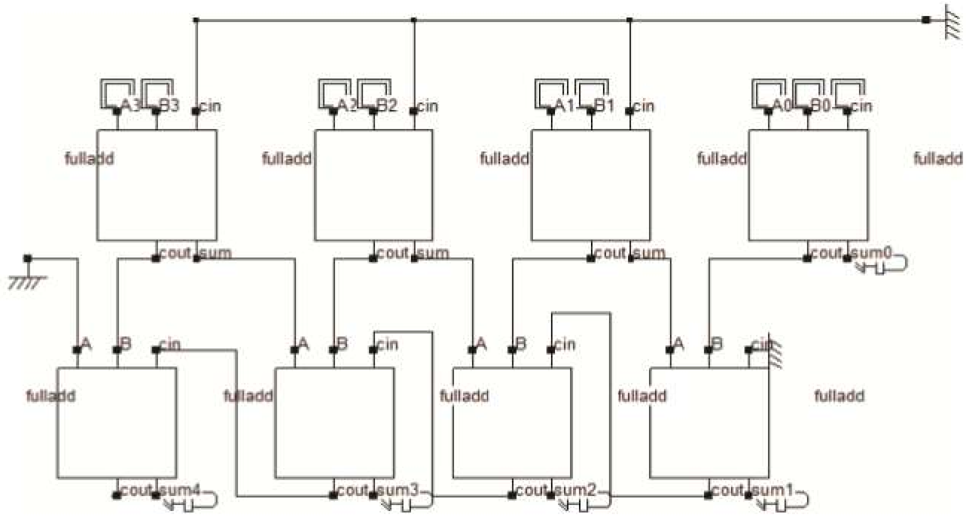


Fig 8. Carry Save Adder

Operation of carry save adder:

It is the complex adder which is completely combination of array of parallel full adder, with each of three input bits A, B, and cin which are loaded into each full adder's as inputs. Every full adder output, Sum is then associated with respective individual output bit, and then output of it (Cout) is associated with output of upcoming significant bit of the output second and also the most reduced one of output second is then provided by the Cin input of carry-save.

The chapter is focused on review of literature of different fast adder designs. To conclude this topic many journals, articles, conference papers have been studied. Some of them are described as below.

Pooja Kansliwal, Mahendra vucha, Rashmi Solanki, Prashant Gurjar (2011) [1]: This paper tells about the equipment usage of the different high speed adders. That are like full adder, carry-look a-head adder, carry_skip adder, carry ripple adder, carry select adder, these are integrated & recreated in the Xilinx-ISE 9.2i stage, whose output parameters caught like region and speed are thought for 16-bit and 8–bit adders .

Reena Rani, Laxmi Kanth Singh, Neelam Sharma (2009) [2]: In this paper numeric operations are performed with the help of a greater radix system, for example, Quaternary Signed Digit (QSD). They rely of Quaternary stamped digit structure. In QSD, every bit is addressed by a bit between 3 to -3. Pass on development, intensive operation on broad number of bits, for instance, 64, 128, and higher shall completed using unfaltering deferral, less diserse quality. FPGA instruments are used for Hardware implementation of these circuits. The arrangements mirrored using modalism programming and joined with the help of Leonardo Spectrum.

Nuno Roma, Tiago Dias, Leonel Sousa (2012) [3]: In this paper a point by point connection examination of the couple of fast adder structures for prevalent VLSI configuration is done. The appraisal of those structures is firstly finished in perspective of a clear gate check show range and input concede unit of time. The results gained with such model were then endorsed by using two totally uncommon real execution improvements, specifically CMOS consolidated circuits and Field Programmable Gate Arrays (FPGA). Test comes to exhibit that among the showed and evaluated topologies, the adder configuration in perspective of the radix-2 repetive association converter offered the slightest defer when realized with any of the considered advances. Regardless, it was in like manner the topology that required the most elevated measure of component. The presented results can be seen as a critical resource in the assurance of the most reasonable adder used to perform the operation in specific technology.

Jasbir Kaur, Lalith Sood (2015) [4]: In this paper, the execution of the different adders, for example, the Carry skip adder, the Carry increment adder, Ripple carry adder, the Carry look a-head adder, the Carry select adder , the Carry save adder are talked about, they are analyzed in the premise of their execution parameter's, for example speed, power circulation and area..

Akash Kumar, Deepika Sharma (2013) [5]: This paper tells about the comparative evaluation of the delay and speed of different varieties of adder just like the carry by-pass adder, the carry ripple adder, carry-look a-head adder, the carry select adder to generate high pace 32-bit MAC unit. The design is simple for carry ripple adder but it is well appropriate for just addition of the less width operand due to the postpone which is increasing linearly with width of operands. The carry bypass adder calls is hardly ever large than the area required by using the ripple carry adder. Postponement of the carry look a-head adder is much less as examine to other. A ripple carry adder would be slow than carry-look adder however carry look adder require comparingly large area. For high-speed multiplication and accumulation, we can use carry-look ahead adder for 32-bit MAC unit. In reality, the multiplier of carry-look a-head adder is two times of velocity of multiplier.

George Joseph, Anand N. Jayakrishnan , P. Johny S Raj (2012) [6]: In this paper usage of carry ripple adder, Kogge Stone-adder, carry select with Kogge Stone-adder have been performed in Virtex five FPGA device x5vlx20t and the use of Xilinx, generation delay have been calculated of the adder systems. The Kogge Stone adder is small delay than the alternative adder systems. As a result carry select of Kogge Stone adder good result, may be used as a high speed processors for speed carry community of adder circuit. Combining carry select to the Kogge Stone adder, may leads to a good and efficient adder for super speed processor which is carried and utilized for high velocity compute application.

R. P. P Singh, Praveen Kumar, Balwinder Singh (2009) [7]: In this paper, the overall performance analysis of the exceptional fast adders has been accomplished. The evaluation started on the premise of the following parameter performances i.e. Power consumption, area, and speed. However they'd presented a layout technique for the hybrid carry skip adders/carry look ahead. The usage of each fix and the variable block length has changed carry bypass adder. In conventional carry bypass adder, to generate carry for subsequent block each having ripple carry adder & skip logic has performed after every block. The velocity of depends upon at the carry forward from the preceding block to the subsequent block. In carry by-pass adder, we use the bring appearance in advance common sense in every block for generating the deliver to further blocks. Then changed carry by-pass adder’s with the fixed circuit require greater CLB’s due to the deliver appearance of Carry look ahead logic, whereas the circuit chematic, location would be carried out.

José Luís Güntzel, Jucemar Monteiro, Luciano Agostini (2011) [8]: This paper implemented the 2 different version for add-one carry select adder (A1CSA), that is enhanced under the bases of cell of Very Large Scale Integration layout flow. Those architectures, at the side of the CRA were synthesized for the 45nm technology TSMC fixed cell libraries used from Synopsys layout Compiler under Topographical mode along with other conventional adder architecture (CLA and CSA). (A1CSA and A1CSAH) are the two less energy dissipation add-one carry select adder appropriate for standard-cells synthesis. On common, 22.2% less amount of area compared to that of carry-select Adder hence synthesis effects has given that A1CSA is the smallest fast adder requiring, on. Also, they confirmed that an average, 10.8% quicker, 3.4% more strength-efficient over that of the carry-look ahead Adder for the A1CSAH , similar to the high-quality choice for the excessive efficiency, pace are addition.

Kiran.M, Pavan Kumar.M.O.V (2013) [9]: In this paper it has defined that the comparative overall performance for fast implementing adders inclusive of Advanced CSLA, CSLA, PPA, Kogge stone structure, Ling adder on the excellent working frequencies. The results shows that parameters of power and area the Ling adder has a good deal better. There s a variation in the area and power which is reduced to 75% for 71MHz operating frequency. In 45nm technology these designs are performed using CADANCE layout tools . It's showed low power and low area for the Ling adder architecture designed. It is efficient and easy for VLSI hardware implementation.

Ramanath J Nayak, Ravikumar A Javali, Manjunath C Lakkannavar, Ashish M Mhetar (2014) [10]: In this paper after evaluating the timing consequences comparison for the carry save adder by carry ripple adder, carry look ahead adder for convey appearance in advance adder they have told that the performed design i.e. carry save adder with carry look a-head adder has set 27.5% quicker. Hence this CSA far utilized for Fast Fourier transform (FFT), Digital signal Processing (DSP) and also for different packages for which timing constraints have critical in which they have able to capable of meeting time necessities, progress in performance and also on the bases of a few energy and area.

S. Salivahanan, V. Kavinilavu, Samiappa Sakthikumaran, V. S. Kanchana Bhaaskaran, C. Vinoth and B. Brindha (2011) [11]: In this paper, they have proposed the design which is less complicated result for enhancing rate of carry select adder. The CSA has the drawback of having high chip location, that triumph over use of the generated 4-bit incrementor unit. The generated unit is likewise located to use much less energy. It may be used to speed the very last summation in parallel multiplier design and different architecture that makes use of adder design. The shape that is synthesized with the help of Synopsys front-end package deal the usage of SAED 90nm generation.

Rajwinder Kaur, Amit Grover (2015) [12] : In this paper, the prevailing carry select adder and its designing technique in the VLSI design has been defined. Even though those diverse designing approach which defined above are proved to make more talented carry select adder having less power utilization and less area than the opposite adders. More recent modification can focus on reaching extra advanced power-area-delay carry select adder for processor processing processor in very large scale integration layout.

Laxman Shanigarapu, Bhavana P. Shrivastava (2013) [13]: A uique method that has been derived in this paper to decrease the location, energy and delay for SQRT CSLA architecture. Which indicates layout for carry select adder implemented by the use of D-Latch and in comparison with normal CSA and modified CSA(BEC and without the usage of Multiplexer). Which are applied on Spartan XC3S500E FPGA device and the overall performance is verified. Area and power is calculated with the aid of the use of synopsys RTL tool. This paper having higher outcomes when as compared to CSA and changed techniques.

Aamir A. Farooqui, Vojin G. Oklobdzija ()[14]: They had proposed layout and implementation of MAC unit, which could carry out 32x32, 32x16, and pair of 16x16 multiplication, resulting a throughput of two, 1, and 1 cycle. In which multiplier, Booth encoding & 3-dimensional (TDM) method has been used, to result high speed multiplier. Unique circuit has generated to house all types of operands (sign or unsigned) also to deal with sign extension.

Changed booth algorithm combined with TDM and sign correction circuit effects for multiplier, with a delay (ppa) equal to six xor gates.

Abdelgawad (2013) [15]: They have designed a quick and less energy consuming multiply accumulate (MACC) Unit. Whose outcome shows that designed approach the other merging techniques. Final results indicates as the proposed 32-bit MAC unit reduces 5.5% of the location, 9% of the power, and delay by means of 13%. Finally, they like to emphasis the novelty and simplicity of that MAC unit to be utilized in DSP for WSNs future nodes.

David B. Roberts and Yuyun Liao (2002) [16]: In this paper, to satisfy the high requirement of almost high level DSP packages, a performance high and energy low 32-bit mac unit was designed. Which reaches 600 MHz in 1.3 V and 150 MHz at zero.7 V, further 800 MHz in 1.6 V. The excessive throughput rate is completed through utilizing a mixing length-encoding scheme & brand-new multiply accumulator structure which is more suitable DSP. The less energy intake is performed by the use of CPL at the critical data paths, static COMS on the relaxation of the MAC for numerous power reduction techniques.

Ohsang Kwon, Kevin Nowka, Earl E. Swartzlander, Jr. (2002) [17]: In this paper, a new rapid 5:3 compression approach is derived from a quick 2-bit adder cell. It has the delay of 2 _xor whilst a brand new logical decomposition is used. Further, its one-level dynamic CMOS circuit is proposed for exceedingly customized layout methodology. For the partial product reduction of a sixteen-bit with the aid of 16-bit MAC, the use of the brand new 5:3 compressor cell ends in 14.3% speed development in the form of xor delay. In distinctly custom designed dynamic CMOS circuit implementation, 11.7% speed development is determined with 8.1% much less power consumption in 0.225 μm bulk CMOS technology.

Adders are the most basic components of the CPU (central processing unit) in today's Digital world. They are used in floating point calculations, ALUs, also in order to compute addresses of memory. In other applications like microprocessors and digital signal processors (DSP) architectures full - adders plays important role.

As day by day digitalization is increasing in every application and so that the speed of processors are increasing. So, to fulfill the demand of digitalization we need adders which provide us accurate outputs with very less consumption of the constraints given below.

The accuracy of the digital devices is mostly determined by the operation of respective adders. The most reviewed constraints for designing the adder are area, power, speed, timing, number of cells and delay. So, Adders with optimized area, power efficient i.e., consumes very much less power, high speed and also performing the operation using the less number of cells.

4.1 Objectives of study:

- To design 4-bit and 8-bit Carry Select Adder, Carry Skip Adder and Carry Save Adder.
- To design Verilog code i.e., main module and test bench for all the three adders.
- To implement the code in Xilinx and Cadence NCsim and generate the waveform and RTL on different technologies
- To analysis the Timing, Area, Power on 45nm, 90nm, 180nm technologies.
- To compare the different parameters for different technologies.

4.2 Tools/ Software's used:

Xilinx, Cadence NCsim

4.3 Technologies used:

45nm, 90nm, 180nm.

CHAPTER 5 RESEARCH METHODOLOGY

5.1 PROBLEM FORMULATION:

As day by day digitalization is increasing in every application and so that the speed of processors are increasing. So, to fulfil the demand of digitalization we need adders which provide us accurate outputs with very less consumption of the constraints given below.

The high speed and the exact operation of the digital devices is mostly derived by the accuracy and speed of the respective adder. The most noted constraints for the adder designs are the area, power, speed, timing, number of cells and delay.

5.2 DESIGN APPROACH:

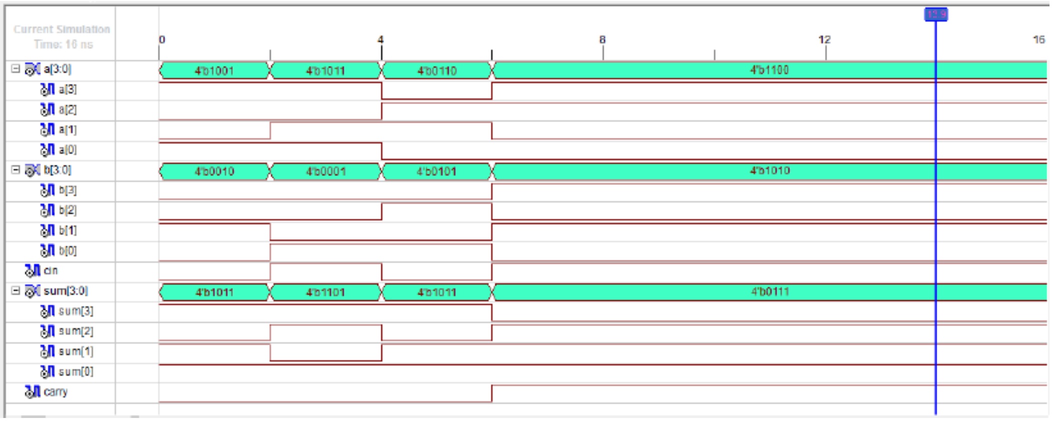
To design these fast adders, we have studied various papers. To implement them we need Verilog code i.e., main module and test bench. After getting the module for respective adders we tried to simulate the module in Xilinx and cadence NCsim tool, at this time we get the simulation result for different input combination. For synthesis of delay and power we use fast.lib and slow.lib where we check the delay and power at 45nm, 90nm, 180nm technologies. So, finally our main aim is to compare these techniques at the different technologies.

6.1 SIMULATION AND SYNTHESIS RESULTS OF 4BIT FAST ADDERS:

6.1.1.1 CARRY SELECT ADDER:

Carry select adder has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.

Fig 9. Output waveform of Carry select adder



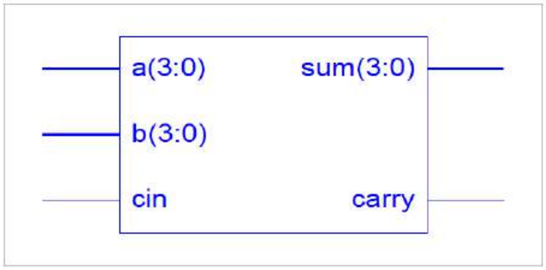


Fig 10. Carry select adder

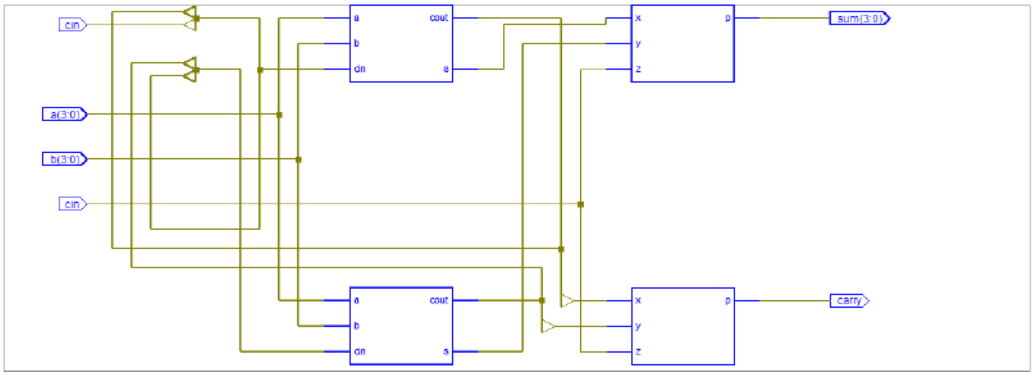


Fig 11. RTL view of carry select adder

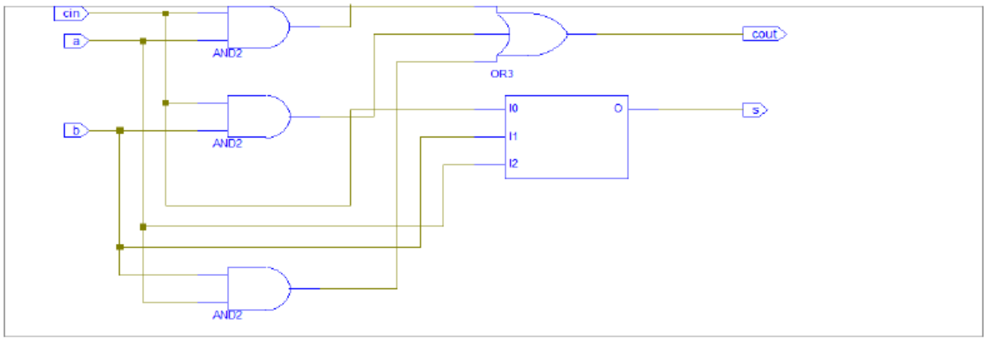


Fig 12. RTL view of full adder

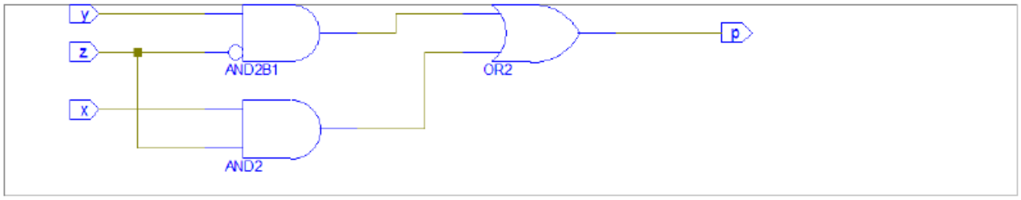


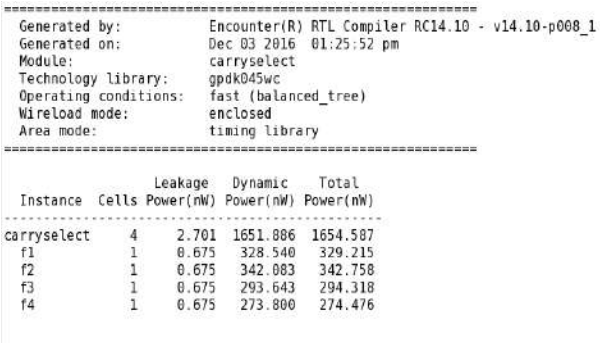
Fig 13. RTL view of multiplexer

6.1.1.2 POWER SYNTHESIS RESULT:

For the synthesis, here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

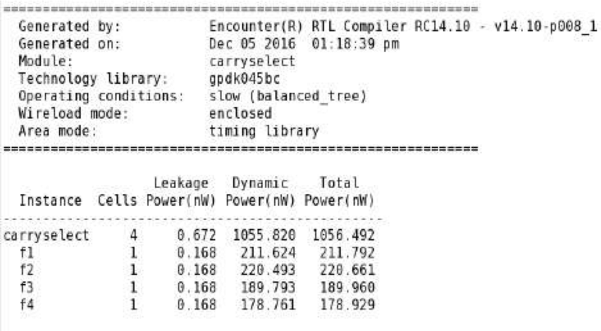
A. Using 45nm Technology

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 03 2016  01:25:52 pm
  Module:                  carryselect
  Technology library:      gpdk045wc
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                     Leakage   Dynamic    Total
  Instance   Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
carryselect     4     2.701  1651.886  1654.587
  f1            1     0.675   328.540   329.215
  f2            1     0.675   342.083   342.758
  f3            1     0.675   293.643   294.318
  f4            1     0.675   273.800   274.476
```

Slow.lib

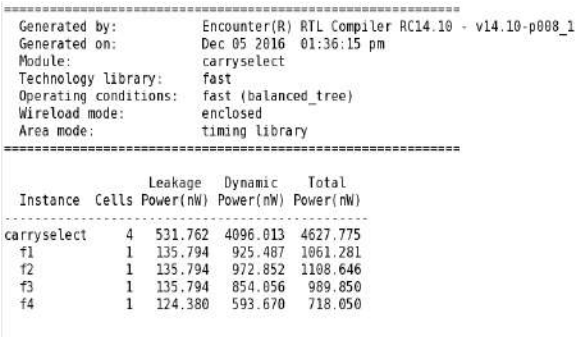


```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:18:39 pm
  Module:                  carryselect
  Technology library:      gpdk045bc
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                     Leakage   Dynamic    Total
  Instance   Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
carryselect     4     0.672  1055.820  1056.492
  f1            1     0.168   211.624   211.792
  f2            1     0.168   220.493   220.661
  f3            1     0.168   189.793   189.960
  f4            1     0.168   178.761   178.929
```

Fig 14. Power synthesis report for Carry select adder using 45nm

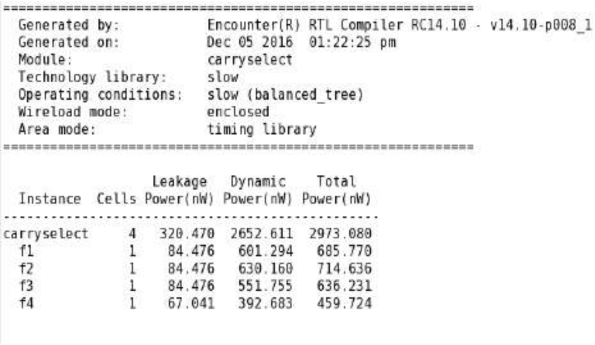
B. Using 90nm Technology

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:36:15 pm
  Module:                  carryselect
  Technology library:      fast
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                     Leakage   Dynamic    Total
  Instance   Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
carryselect     4   531.762  4096.013  4627.775
  f1            1   135.794   925.487  1061.281
  f2            1   135.794   972.852  1108.646
  f3            1   135.794   854.056   989.850
  f4            1   124.380   593.670   718.050
```

Slow.lib

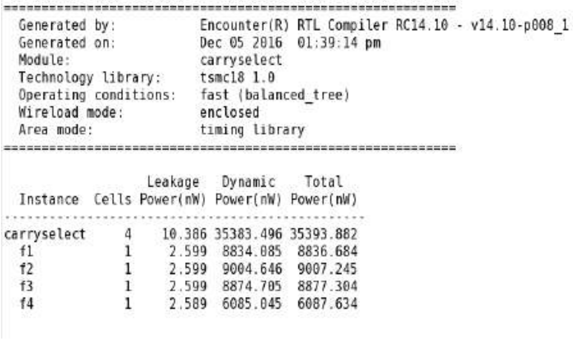


```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:22:25 pm
  Module:                  carryselect
  Technology library:      slow
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                     Leakage   Dynamic    Total
  Instance   Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
carryselect     4   320.470  2652.611  2973.080
  f1            1    84.476   601.294   685.770
  f2            1    84.476   630.160   714.636
  f3            1    84.476   551.755   636.231
  f4            1    67.041   392.683   459.724
```

Fig 15. Power synthesis report for Carry select adder using 90nm

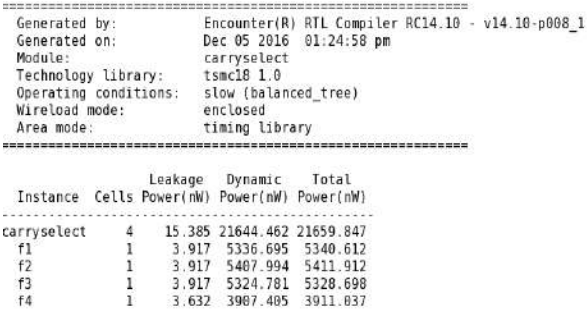
C. Using 180nm Technology:

Fast.lib:



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:39:14 pm
  Module:                 carryselect
  Technology library:     tsmc18 1.0
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                  Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryselect     4    10.386 35383.496 35393.882
  f1            1     2.599  8834.085  8836.684
  f2            1     2.599  9004.646  9007.245
  f3            1     2.599  8874.705  8877.304
  f4            1     2.589  6085.045  6087.634
```

Slow.lib:



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:24:58 pm
  Module:                 carryselect
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                  Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryselect     4    15.385 21644.462 21659.847
  f1            1     3.917  5336.695  5340.612
  f2            1     3.917  5407.994  5411.912
  f3            1     3.917  5324.781  5328.698
  f4            1     3.632  3907.405  3911.037
```

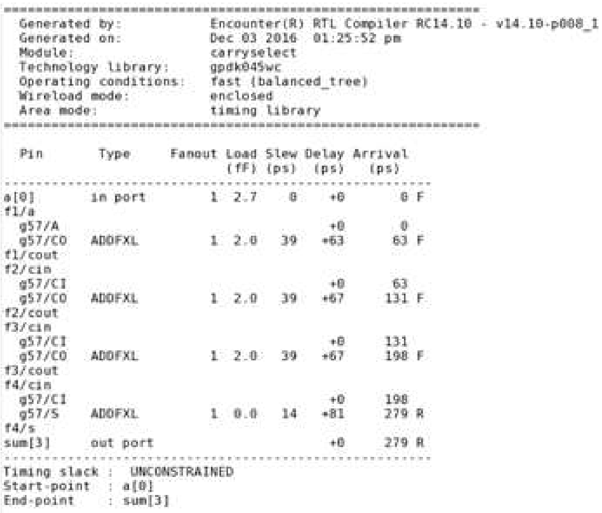
Fig 16. Power synthesis report for Carry select adder using 180nm

6.1.1.3 DELAY AND TIMING SYNTHESIS RESULTS:

Same as in case of power synthesis, here we are using here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

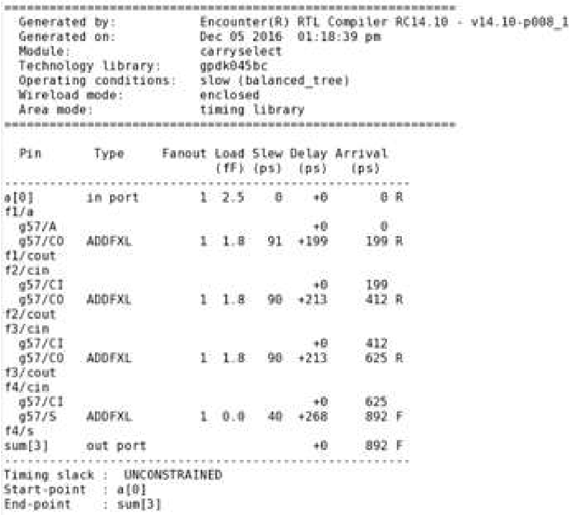
A. Using 45nm Technology

Fast.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 03 2016  01:25:52 pm
  Module:                 carryselect
  Technology library:     gpdk045wc
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-----------------------------------------------------
a[0]        in port       1  2.7    0    +0       0 F
f1/a
  g57/A                                  +0       0
  g57/CO    ADDFXL        1  2.0   39   +63      63 F
f1/cout
f2/cin
  g57/CI                                 +0      63
  g57/CO    ADDFXL        1  2.0   39   +67     131 F
f2/cout
f3/cin
  g57/CI                                 +0     131
  g57/CO    ADDFXL        1  2.0   39   +67     198 F
f3/cout
f4/cin
  g57/CI                                 +0     198
  g57/S     ADDFXL        1  0.0   14   +81     279 R
f4/s
sum[3]      out port                     +0     279 R
-----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[3]
```

Slow.lib

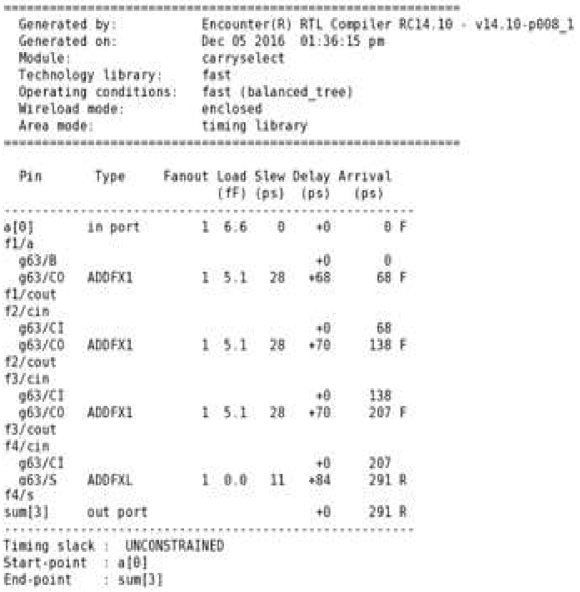


```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:18:39 pm
  Module:                 carryselect
  Technology library:     gpdk045bc
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-----------------------------------------------------
a[0]        in port       1  2.5    0    +0       0 R
f1/a
  g57/A                                  +0       0
  g57/CO    ADDFXL        1  1.8   91  +199     199 R
f1/cout
f2/cin
  g57/CI                                 +0     199
  g57/CO    ADDFXL        1  1.8   90  +213     412 R
f2/cout
f3/cin
  g57/CI                                 +0     412
  g57/CO    ADDFXL        1  1.8   90  +213     625 R
f3/cout
f4/cin
  g57/CI                                 +0     625
  g57/S     ADDFXL        1  0.0   40  +268     892 F
f4/s
sum[3]      out port                     +0     892 F
-----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[3]
```

Fig 17. Timing Synthesis Report for Carry select adder using 45nm

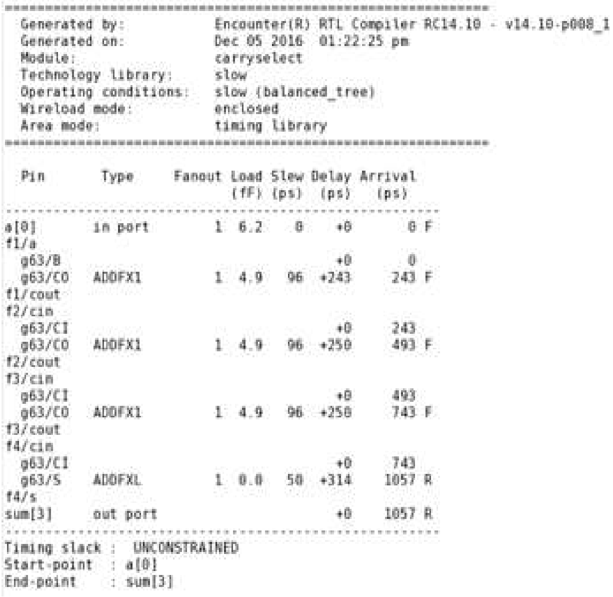
B. Using 90nm Technology

Fast.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:36:15 pm
  Module:                 carryselect
  Technology library:     fast
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
------------------------------------------------------
a[0]       in port        1  6.6    0    +0       0 F
f1/a
  g63/B                                  +0       0
  g63/CO   ADDFX1         1  5.1   28   +68      68 F
f1/cout
f2/cin
  g63/CI                                 +0      68
  g63/CO   ADDFX1         1  5.1   28   +70     138 F
f2/cout
f3/cin
  g63/CI                                 +0     138
  g63/CO   ADDFX1         1  5.1   28   +70     207 F
f3/cout
f4/cin
  g63/CI                                 +0     207
  g63/S    ADDFXL         1  0.0   11   +84     291 R
f4/s
sum[3]     out port                      +0     291 R
------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[3]
```

Slow.lib

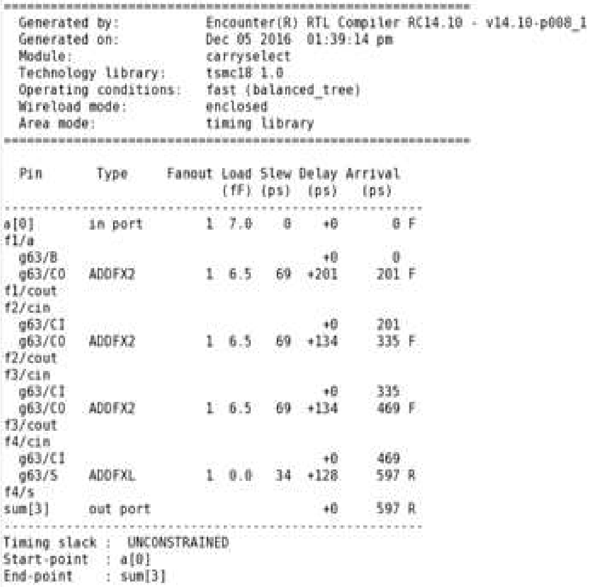


```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:22:25 pm
  Module:                 carryselect
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
------------------------------------------------------
a[0]       in port        1  6.2    0    +0       0 F
f1/a
  g63/B                                  +0       0
  g63/CO   ADDFX1         1  4.9   96  +243     243 F
f1/cout
f2/cin
  g63/CI                                 +0     243
  g63/CO   ADDFX1         1  4.9   96  +250     493 F
f2/cout
f3/cin
  g63/CI                                 +0     493
  g63/CO   ADDFX1         1  4.9   96  +250     743 F
f3/cout
f4/cin
  g63/CI                                 +0     743
  g63/S    ADDFXL         1  0.0   50  +314    1057 R
f4/s
sum[3]     out port                      +0    1057 R
------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[3]
```

Fig 18. Timing Synthesis Report for Carry select adder using 90nm

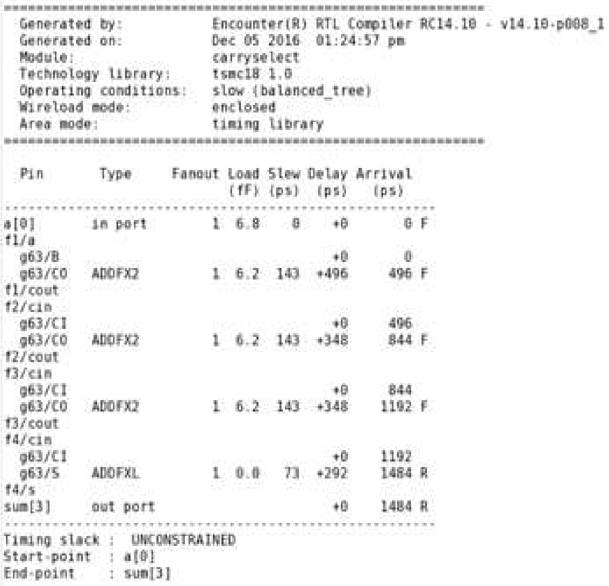
C. Using 180nm Technology:

Fast.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:39:14 pm
  Module:                 carryselect
  Technology library:     tsmc18 1.0
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
------------------------------------------------------
a[0]       in port        1  7.0    0    +0       0 F
f1/a
  g63/B                                  +0       0
  g63/CO   ADDFX2         1  6.5   69  +201     201 F
f1/cout
f2/cin
  g63/CI                                 +0     201
  g63/CO   ADDFX2         1  6.5   69  +134     335 F
f2/cout
f3/cin
  g63/CI                                 +0     335
  g63/CO   ADDFX2         1  6.5   69  +134     469 F
f3/cout
f4/cin
  g63/CI                                 +0     469
  g63/S    ADDFXL         1  0.0   34  +128     597 R
f4/s
sum[3]     out port                      +0     597 R
------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[3]
```

Slow.lib



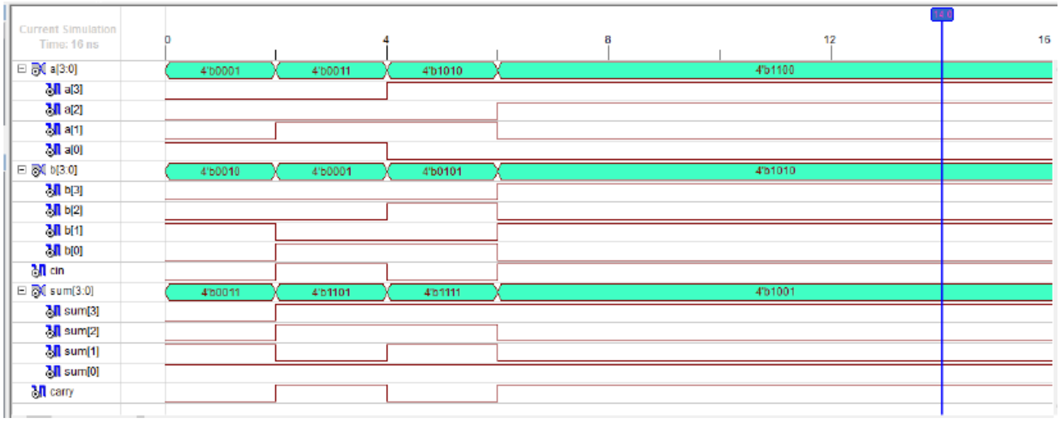
```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:24:57 pm
  Module:                 carryselect
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
------------------------------------------------------
a[0]       in port        1  6.8    0    +0       0 F
f1/a
  g63/B                                  +0       0
  g63/CO   ADDFX2         1  6.2  143  +496     496 F
f1/cout
f2/cin
  g63/CI                                 +0     496
  g63/CO   ADDFX2         1  6.2  143  +348     844 F
f2/cout
f3/cin
  g63/CI                                 +0     844
  g63/CO   ADDFX2         1  6.2  143  +348    1192 F
f3/cout
f4/cin
  g63/CI                                 +0    1192
  g63/S    ADDFXL         1  0.0   73  +292    1484 R
f4/s
sum[3]     out port                      +0    1484 R
------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[3]
```

Fig 19. Timing Synthesis Report for Carry select adder using 180nm

6.1.2.1 CARRY SKIP ADDER:

Carry skip adder has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.

Fig 20. Output waveform of Carry Skip Adder



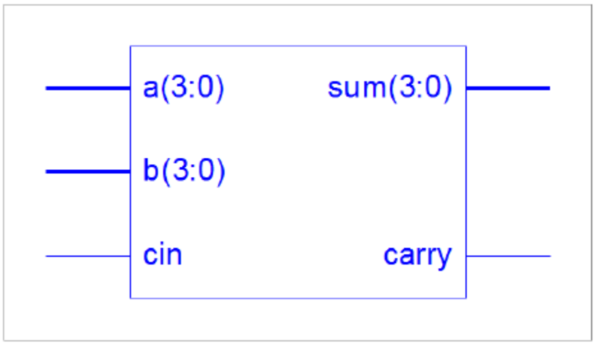


Fig 21. Carry skip adder

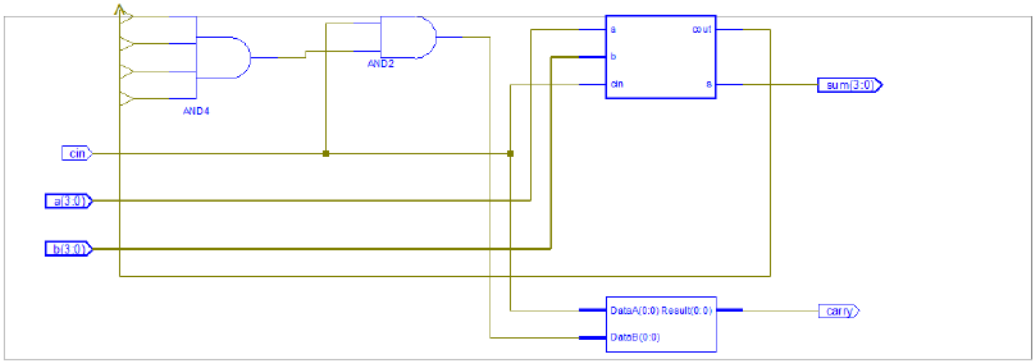


Fig 22. RTL view of carry select adder

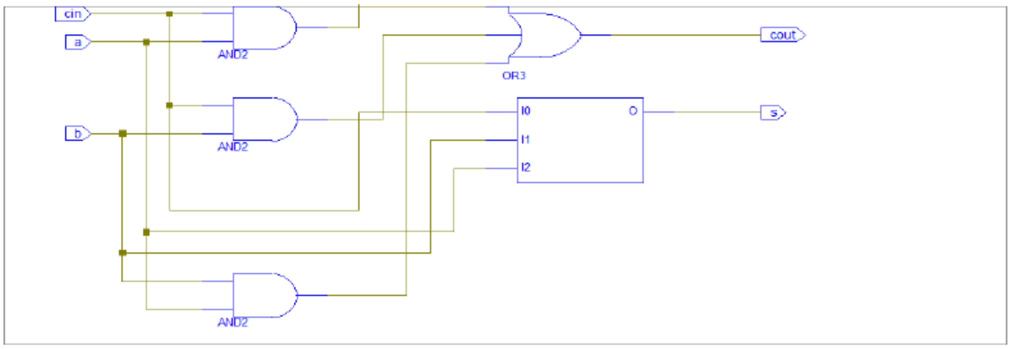


Fig 23. RTL view of full adder

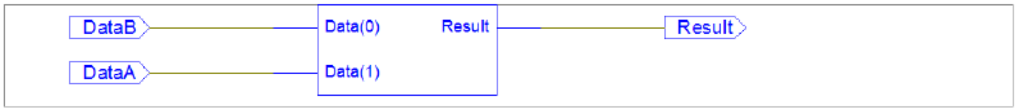


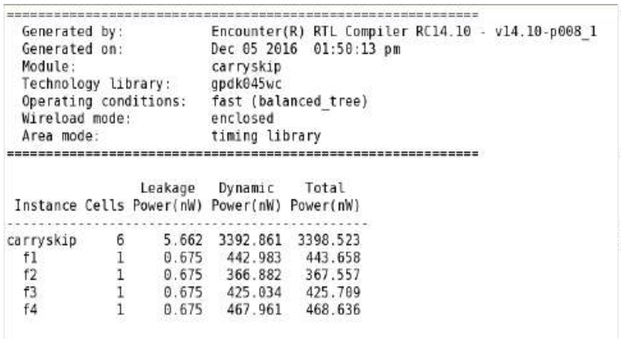
Fig 24. RTL view of OR gate

6.1.2.2 POWER SYNTHESIS RESULT:

For the synthesis, here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

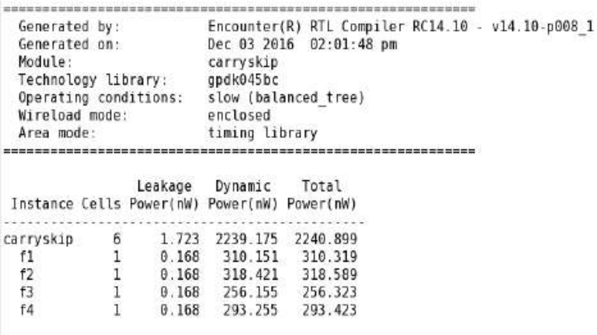
A. Using 45nm Technology

Fast.lib:



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:50:13 pm
  Module:                  carryskip
  Technology library:      gpdk045wc
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                   Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip     6     5.662  3392.861  3398.523
  f1          1     0.675   442.983   443.658
  f2          1     0.675   366.882   367.557
  f3          1     0.675   425.034   425.709
  f4          1     0.675   467.961   468.636
```

Slow.lib:

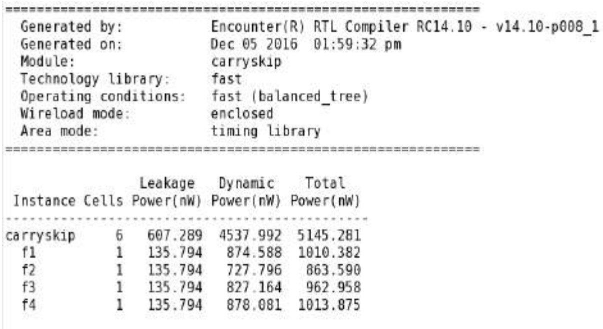


```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 03 2016  02:01:48 pm
  Module:                  carryskip
  Technology library:      gpdk045bc
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                   Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip     6     1.723  2239.175  2240.899
  f1          1     0.168   310.151   310.319
  f2          1     0.168   318.421   318.589
  f3          1     0.168   256.155   256.323
  f4          1     0.168   293.255   293.423
```

Fig 25. Power synthesis report for Carry skip adder using 45nm

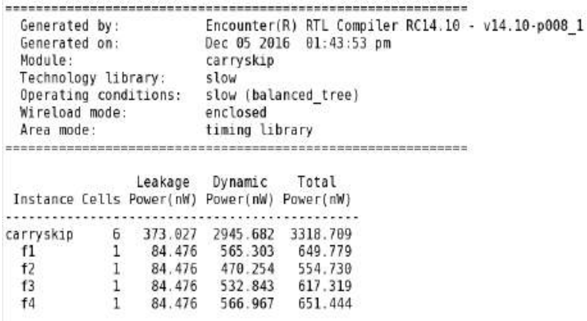
B. Using 90nm Technology

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:59:32 pm
  Module:                  carryskip
  Technology library:      fast
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                   Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip     6   607.289  4537.992  5145.281
  f1          1   135.794   874.588  1010.382
  f2          1   135.794   727.796   863.590
  f3          1   135.794   827.164   962.958
  f4          1   135.794   878.081  1013.875
```

Slow.lib

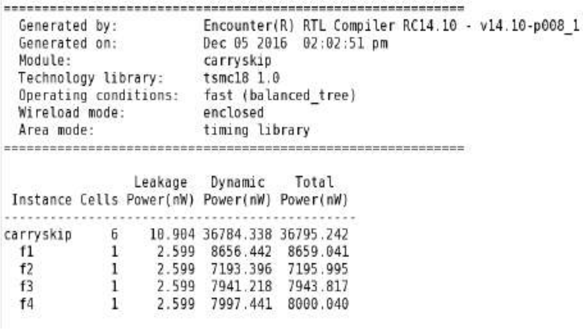


```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:43:53 pm
  Module:                  carryskip
  Technology library:      slow
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                   Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip     6   373.027  2945.682  3318.709
  f1          1    84.476   565.303   649.779
  f2          1    84.476   470.254   554.730
  f3          1    84.476   532.843   617.319
  f4          1    84.476   566.967   651.444
```

Fig 26. Power synthesis report for Carry skip adder using 90nm

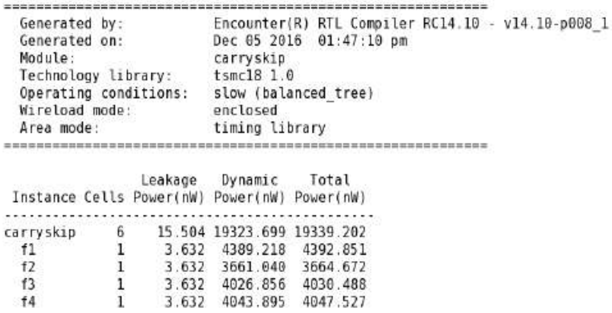
C. Using 180nm Technology:

Fast.lib:



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  02:02:51 pm
  Module:                  carryskip
  Technology library:      tsmc18 1.0
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                  Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip     6    10.904 36784.338 36795.242
  f1          1     2.599  8656.442  8659.041
  f2          1     2.599  7193.396  7195.995
  f3          1     2.599  7941.218  7943.817
  f4          1     2.599  7997.441  8000.040
```

Slow.lib:



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:47:10 pm
  Module:                  carryskip
  Technology library:      tsmc18 1.0
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                  Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip     6    15.504 19323.699 19339.202
  f1          1     3.632  4389.218  4392.851
  f2          1     3.632  3661.040  3664.672
  f3          1     3.632  4026.856  4030.488
  f4          1     3.632  4043.895  4047.527
```

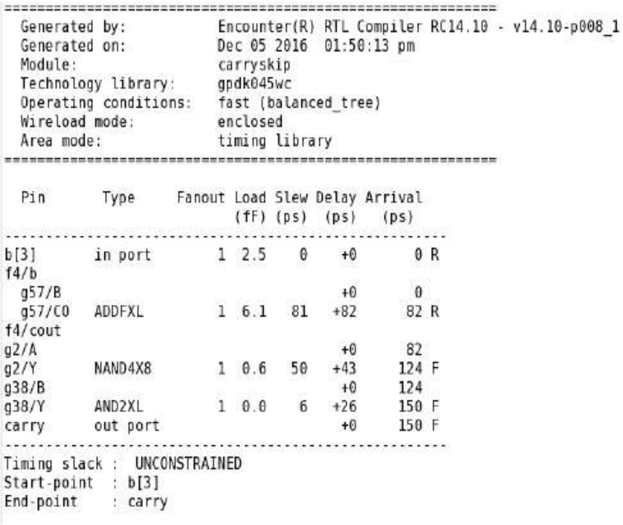
Fig 27. Power synthesis report for Carry skip adder using 180nm

6.1.2.3 DELAY AND TIMING SYNTHESIS RESULTS:

Same as in case of power synthesis, here we are using here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

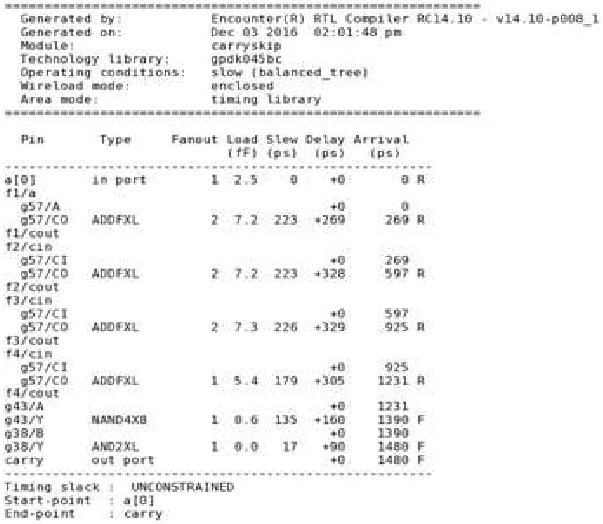
A. Using 45nm Technology

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:50:13 pm
  Module:                  carryskip
  Technology library:      gpdk045wc
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
---------------------------------------------------
b[3]      in port        1  2.5    0   +0       0 R
f4/b
  g57/B                                +0       0
  g57/CO  ADDFXL         1  6.1   81  +82      82 R
f4/cout
g2/A                                   +0      82
g2/Y      NAND4X8        1  0.6   50  +43     124 F
g38/B                                  +0     124
g38/Y     AND2XL         1  0.0    6  +26     150 F
carry     out port                     +0     150 F
---------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[3]
End-point    : carry
```

Slow.lib

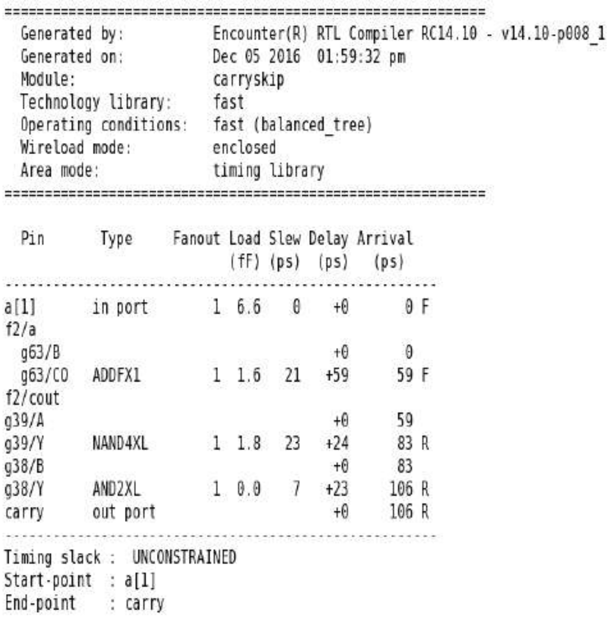


```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 03 2016  02:01:48 pm
  Module:                  carryskip
  Technology library:      gpdk045bc
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
---------------------------------------------------
a[0]      in port        1  2.5    0   +0       0 R
f1/a
  g57/A                                +0       0
  g57/CO  ADDFXL         2  7.2  223 +269     269 R
f1/cout
f2/cin
  g57/CI                               +0     269
  g57/CO  ADDFXL         2  7.2  223 +328     597 R
f2/cout
f3/cin
  g57/CI                               +0     597
  g57/CO  ADDFXL         2  7.3  226 +329     925 R
f3/cout
f4/cin
  g57/CI                               +0     925
  g57/CO  ADDFXL         1  5.4  179 +305    1231 R
f4/cout
g43/A                                  +0    1231
g43/Y     NAND4X8        1  0.6  135 +160    1390 F
g38/B                                  +0    1390
g38/Y     AND2XL         1  0.0   17  +90    1480 F
carry     out port                     +0    1480 F
---------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : carry
```

Fig 28. Timing Synthesis Report for Carry skip adder using 45nm

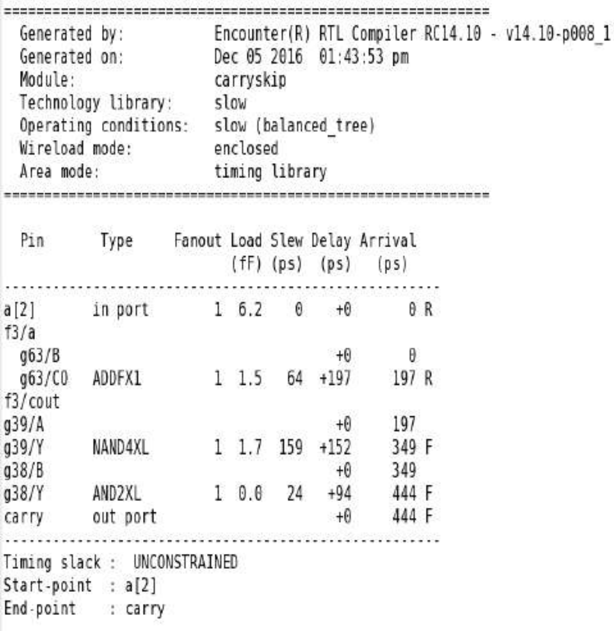
B. Using 90nm Technology

Fast.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:59:32 pm
  Module:                 carryskip
  Technology library:     fast
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]       in port       1  6.6    0    +0       0 F
f2/a
  g63/B                                 +0       0
  g63/CO   ADDFX1        1  1.6   21   +59      59 F
f2/cout
g39/A                                   +0      59
g39/Y      NAND4XL       1  1.8   23   +24      83 R
g38/B                                   +0      83
g38/Y      AND2XL        1  0.0    7   +23     106 R
carry      out port                     +0     106 R
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : carry
```

Slow.lib

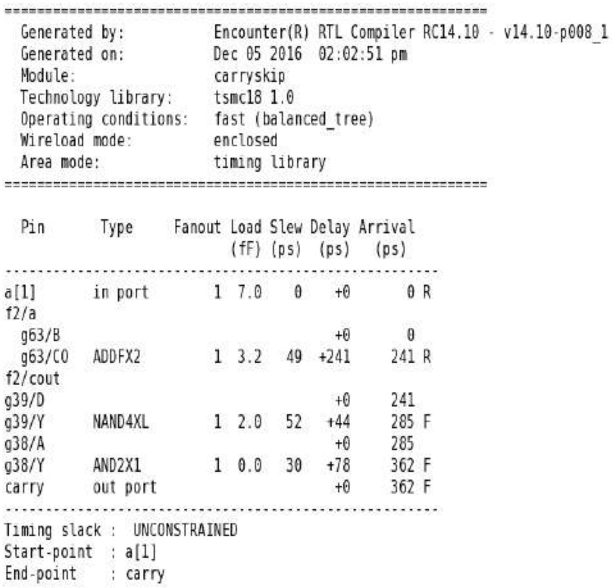


```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:43:53 pm
  Module:                 carryskip
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[2]       in port       1  6.2    0    +0       0 R
f3/a
  g63/B                                 +0       0
  g63/CO   ADDFX1        1  1.5   64  +197     197 R
f3/cout
g39/A                                   +0     197
g39/Y      NAND4XL       1  1.7  159  +152     349 F
g38/B                                   +0     349
g38/Y      AND2XL        1  0.0   24   +94     444 F
carry      out port                     +0     444 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[2]
End-point    : carry
```

Fig 29. Timing Synthesis Report for Carry skip adder using 90nm

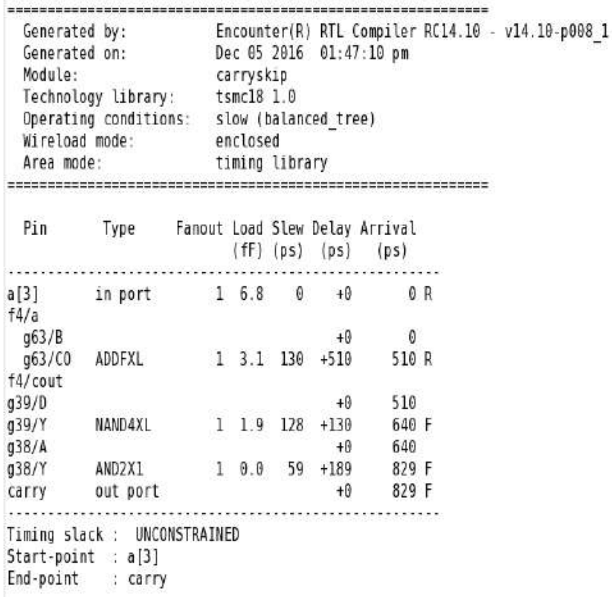
C. Using 180nm Technology:

Fast.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  02:02:51 pm
  Module:                 carryskip
  Technology library:     tsmc18 1.0
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]       in port       1  7.0    0    +0       0 R
f2/a
  g63/B                                 +0       0
  g63/CO   ADDFX2        1  3.2   49  +241     241 R
f2/cout
g39/D                                   +0     241
g39/Y      NAND4XL       1  2.0   52   +44     285 F
g38/A                                   +0     285
g38/Y      AND2X1        1  0.0   30   +78     362 F
carry      out port                     +0     362 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : carry
```

Slow.lib



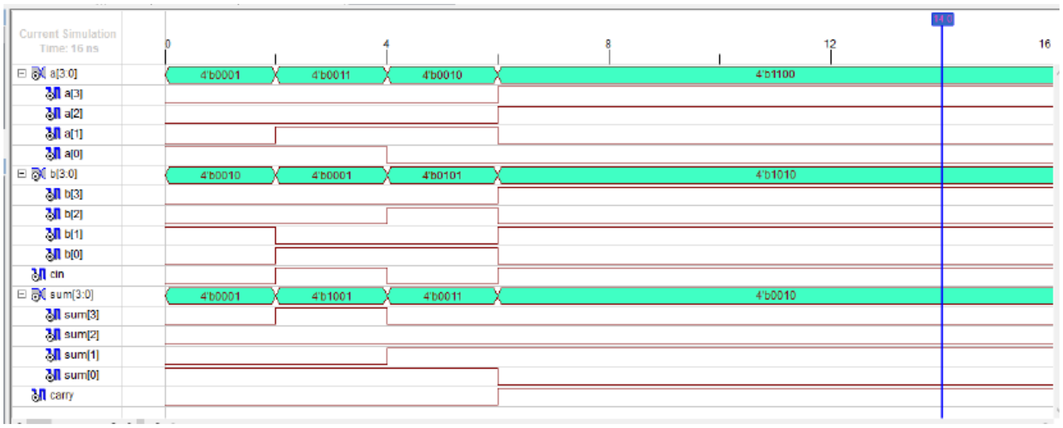
```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:47:10 pm
  Module:                 carryskip
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[3]       in port       1  6.8    0    +0       0 R
f4/a
  g63/B                                 +0       0
  g63/CO   ADDFXL        1  3.1  130  +510     510 R
f4/cout
g39/D                                   +0     510
g39/Y      NAND4XL       1  1.9  128  +130     640 F
g38/A                                   +0     640
g38/Y      AND2X1        1  0.0   59  +189     829 F
carry      out port                     +0     829 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[3]
End-point    : carry
```

Fig 30. Timing Synthesis Report for Carry skip adder using 180nm

6.1.3.1 CARRY SAVE ADDER:

Carry save adder has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.

Fig 31. Output waveform of carry save adder



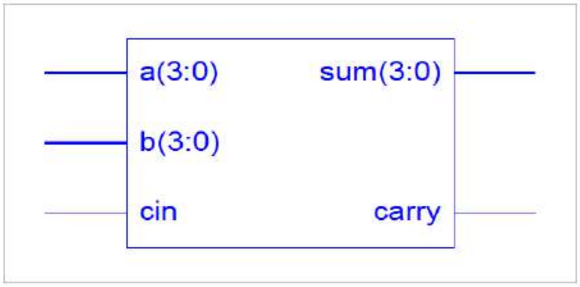


Fig 32. Carry save adder

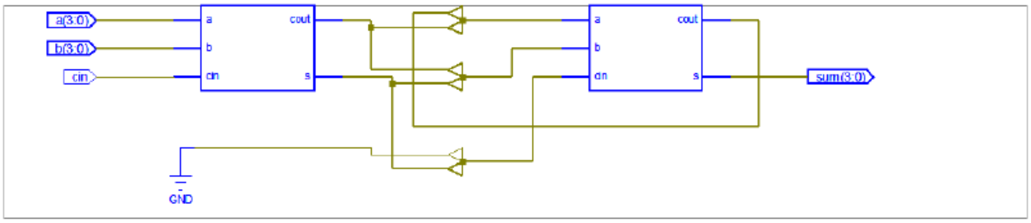


Fig 33. RTL view of carry save adder

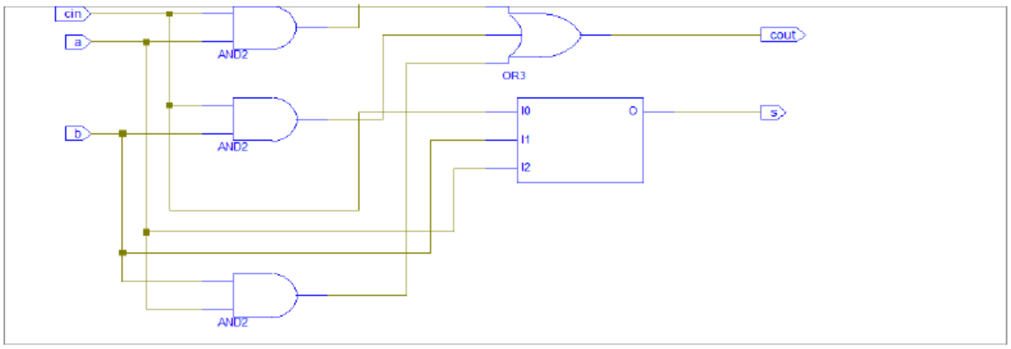


Fig 34. RTL view of full adder

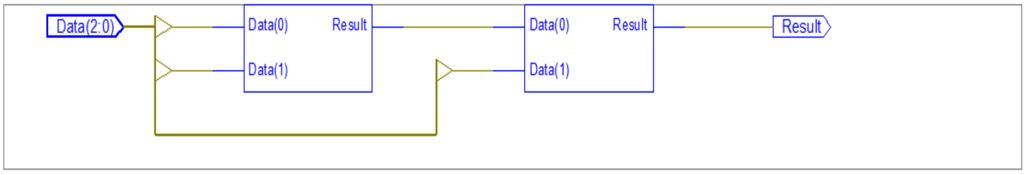


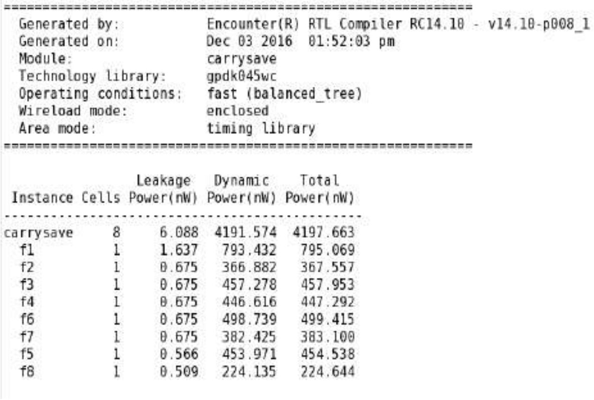
Fig 35. RTL view of XOR Gate

6.1.3.2 POWER SYNTHESIS RESULT:

For the synthesis, here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

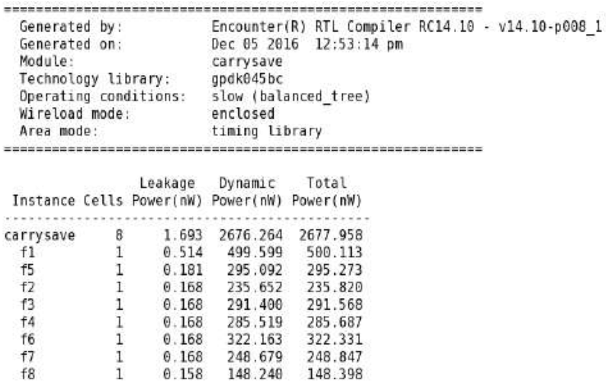
A. Using 45nm Technology

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 03 2016  01:52:03 pm
  Module:                  carrysave
  Technology library:      gpdk045wc
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                  Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carrysave     8     6.088  4191.574  4197.663
  f1          1     1.637   793.432   795.069
  f2          1     0.675   366.882   367.557
  f3          1     0.675   457.278   457.953
  f4          1     0.675   446.616   447.292
  f6          1     0.675   498.739   499.415
  f7          1     0.675   382.425   383.100
  f5          1     0.566   453.971   454.538
  f8          1     0.509   224.135   224.644
```

Slow.lib

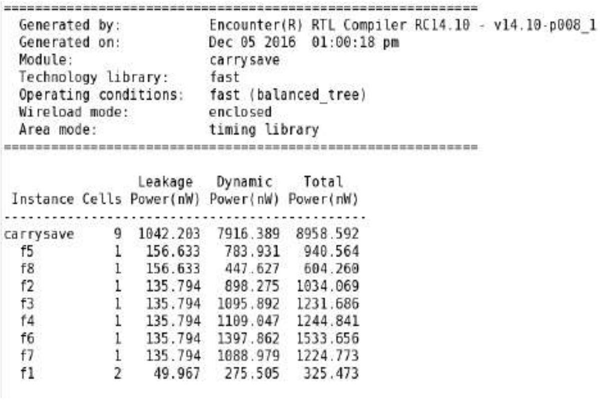


```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  12:53:14 pm
  Module:                  carrysave
  Technology library:      gpdk045bc
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                  Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carrysave     8     1.693  2676.264  2677.958
  f1          1     0.514   499.599   500.113
  f5          1     0.181   295.092   295.273
  f2          1     0.168   235.652   235.820
  f3          1     0.168   291.400   291.568
  f4          1     0.168   285.519   285.687
  f6          1     0.168   322.163   322.331
  f7          1     0.168   248.679   248.847
  f8          1     0.158   148.240   148.398
```

Fig 36. Power synthesis report for Carry save adder using 45nm

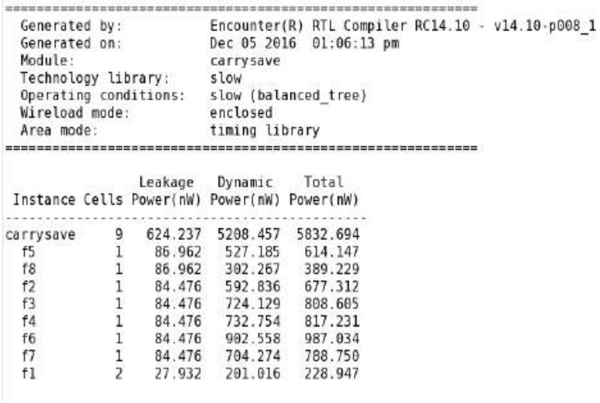
B. Using 90nm Technology

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:00:18 pm
  Module:                  carrysave
  Technology library:      fast
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                  Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carrysave     9  1042.203  7916.389  8958.592
  f5          1   156.633   783.931   940.564
  f8          1   156.633   447.627   604.260
  f2          1   135.794   898.275  1034.069
  f3          1   135.794  1095.892  1231.686
  f4          1   135.794  1109.047  1244.841
  f6          1   135.794  1397.862  1533.656
  f7          1   135.794  1088.979  1224.773
  f1          2    49.967   275.505   325.473
```

Slow.lib

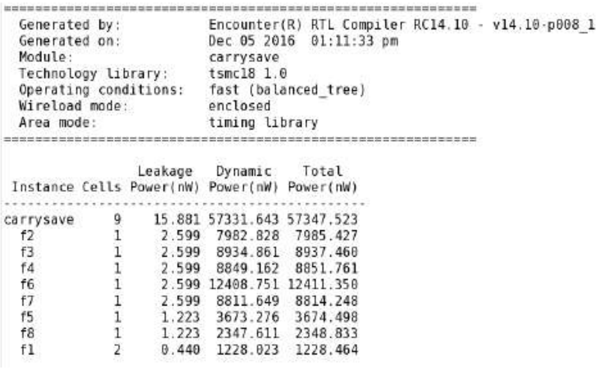


```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:06:13 pm
  Module:                  carrysave
  Technology library:      slow
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                  Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carrysave     9   624.237  5208.457  5832.694
  f5          1    86.962   527.185   614.147
  f8          1    86.962   302.267   389.229
  f2          1    84.476   592.836   677.312
  f3          1    84.476   724.129   808.605
  f4          1    84.476   732.754   817.231
  f6          1    84.476   902.558   987.034
  f7          1    84.476   704.274   788.750
  f1          2    27.932   201.016   228.947
```

Fig 37. Power synthesis report for Carry save adder using 90nm

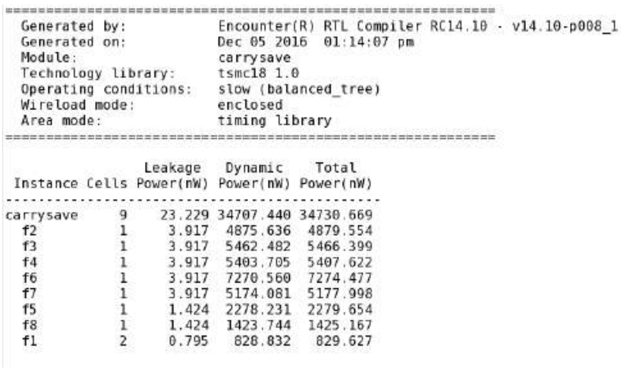
C. Using 180nm Technology:

Fast.lib:



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:11:33 pm
  Module:                 carrysave
  Technology library:     tsmc18 1.0
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                   Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
----------------------------------------------
carrysave     9    15.881 57331.643 57347.523
  f2          1     2.599  7982.828  7985.427
  f3          1     2.599  8934.861  8937.460
  f4          1     2.599  8849.162  8851.761
  f6          1     2.599 12408.751 12411.350
  f7          1     2.599  8811.649  8814.248
  f5          1     1.223  3673.276  3674.498
  f8          1     1.223  2347.611  2348.833
  f1          2     0.440  1228.023  1228.464
```

Slow.lib:



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  01:14:07 pm
  Module:                 carrysave
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                   Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
----------------------------------------------
carrysave     9    23.229 34707.440 34730.669
  f2          1     3.917  4875.636  4879.554
  f3          1     3.917  5462.482  5466.399
  f4          1     3.917  5403.705  5407.622
  f6          1     3.917  7270.560  7274.477
  f7          1     3.917  5174.081  5177.998
  f5          1     1.424  2278.231  2279.654
  f8          1     1.424  1423.744  1425.167
  f1          2     0.795   828.832   829.627
```

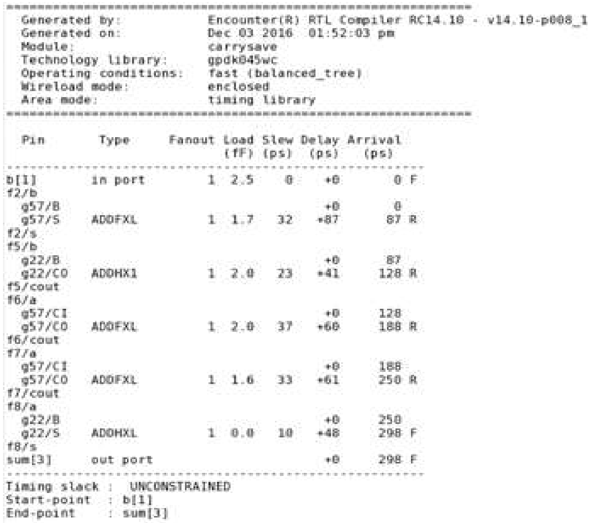
Fig 38. Power synthesis report for Carry save adder using 180nm

6.1.3.3 DELAY AND TIMING SYNTHESIS RESULTS:

Same as in case of power synthesis, here we are using here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

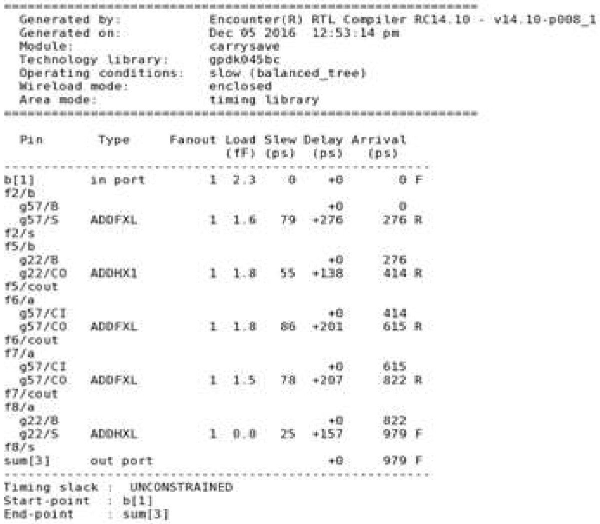
A. Using 45nm Technology

Fast.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 03 2016  01:52:03 pm
  Module:                 carrysave
  Technology library:     gpdk045wc
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin        Type     Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
----------------------------------------------------
b[1]        in port        1  2.5    0    +0       0 F
f2/b
  g57/B                               +0       0
  g57/S     ADDFXL         1  1.7   32   +87      87 R
f2/s
f5/b
  g22/B                               +0      87
  g22/CO    ADDHX1         1  2.0   23   +41     128 R
f5/cout
f6/a
  g57/CI                              +0     128
  g57/CO    ADDFXL         1  2.0   37   +60     188 R
f6/cout
f7/a
  g57/CI                              +0     188
  g57/CO    ADDFXL         1  1.6   33   +61     250 R
f7/cout
f8/a
  g22/B                               +0     250
  g22/S     ADDHXL         1  0.0   10   +48     298 F
f8/s
sum[3]      out port                  +0     298 F
----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[1]
End-point    : sum[3]
```

Slow.lib

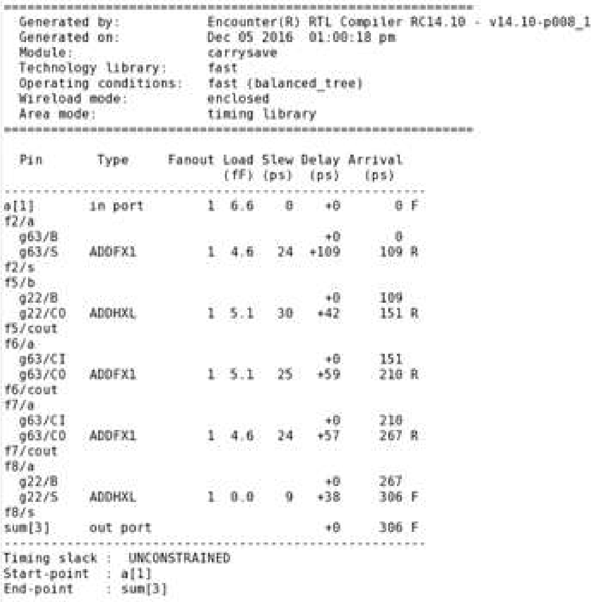


```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Dec 05 2016  12:53:14 pm
  Module:                 carrysave
  Technology library:     gpdk045bc
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin        Type     Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
----------------------------------------------------
b[1]        in port        1  2.3    0    +0       0 F
f2/b
  g57/B                               +0       0
  g57/S     ADDFXL         1  1.6   79  +276     276 R
f2/s
f5/b
  g22/B                               +0     276
  g22/CO    ADDHX1         1  1.8   55  +138     414 R
f5/cout
f6/a
  g57/CI                              +0     414
  g57/CO    ADDFXL         1  1.8   86  +201     615 R
f6/cout
f7/a
  g57/CI                              +0     615
  g57/CO    ADDFXL         1  1.5   78  +207     822 R
f7/cout
f8/a
  g22/B                               +0     822
  g22/S     ADDHXL         1  0.0   25  +157     979 F
f8/s
sum[3]      out port                  +0     979 F
----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[1]
End-point    : sum[3]
```

Fig 39. Timing Synthesis Report for Carry save adder using 45nm

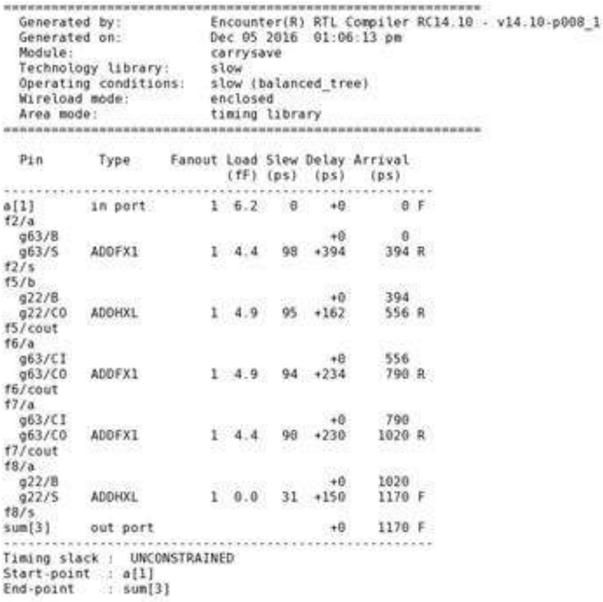
B. Using 90nm Technology

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:00:18 pm
  Module:                  carrysave
  Technology library:      fast
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]       in port        1  6.6    0    +0       0 F
f2/a
  g63/B                                  +0       0
  g63/S    ADDFX1         1  4.6   24  +109     109 R
f2/s
f5/b
  g22/B                                  +0     109
  g22/CO   ADDHXL         1  5.1   30   +42     151 R
f5/cout
f6/a
  g63/CI                                 +0     151
  g63/CO   ADDFX1         1  5.1   25   +59     210 R
f6/cout
f7/a
  g63/CI                                 +0     210
  g63/CO   ADDFX1         1  4.6   24   +57     267 R
f7/cout
f8/a
  g22/B                                  +0     267
  g22/S    ADDHXL         1  0.0    9   +38     306 F
f8/s
sum[3]     out port                      +0     306 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : sum[3]
```

Slow.lib

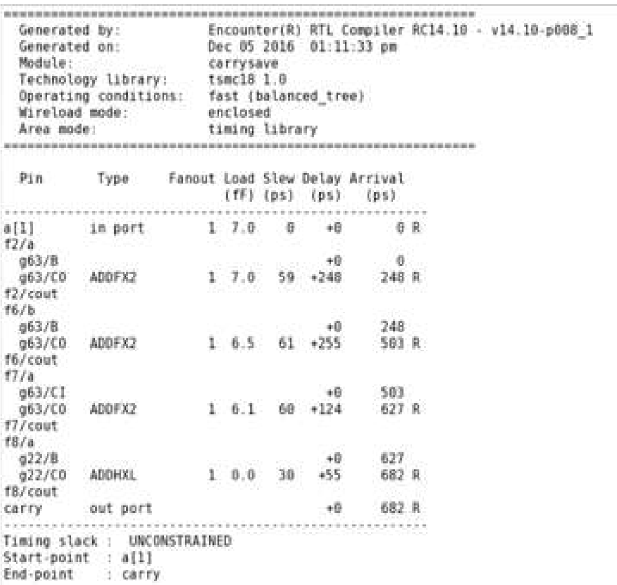


```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:06:13 pm
  Module:                  carrysave
  Technology library:      slow
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]       in port        1  6.2    0    +0       0 F
f2/a
  g63/B                                  +0       0
  g63/S    ADDFX1         1  4.4   98  +394     394 R
f2/s
f5/b
  g22/B                                  +0     394
  g22/CO   ADDHXL         1  4.9   95  +162     556 R
f5/cout
f6/a
  g63/CI                                 +0     556
  g63/CO   ADDFX1         1  4.9   94  +234     790 R
f6/cout
f7/a
  g63/CI                                 +0     790
  g63/CO   ADDFX1         1  4.4   90  +230    1020 R
f7/cout
f8/a
  g22/B                                  +0    1020
  g22/S    ADDHXL         1  0.0   31  +150    1170 F
f8/s
sum[3]     out port                      +0    1170 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : sum[3]
```

Fig 40. Timing Synthesis Report for Carry save adder using 90nm

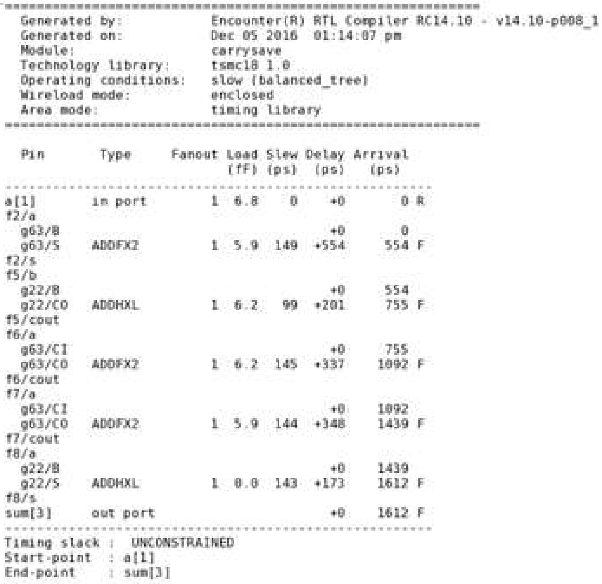
C. Using 180nm Technology:

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:11:33 pm
  Module:                  carrysave
  Technology library:      tsmc18 1.0
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]       in port        1  7.0    0    +0       0 R
f2/a
  g63/B                                  +0       0
  g63/CO   ADDFX2         1  7.0   59  +248     248 R
f2/cout
f6/b
  g63/B                                  +0     248
  g63/CO   ADDFX2         1  6.5   61  +255     503 R
f6/cout
f7/a
  g63/CI                                 +0     503
  g63/CO   ADDFX2         1  6.1   60  +124     627 R
f7/cout
f8/a
  g22/B                                  +0     627
  g22/CO   ADDHXL         1  0.0   30   +55     682 R
f8/cout
carry      out port                      +0     682 R
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : carry
```

Slow.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Dec 05 2016  01:14:07 pm
  Module:                  carrysave
  Technology library:      tsmc18 1.0
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]       in port        1  6.8    0    +0       0 R
f2/a
  g63/B                                  +0       0
  g63/S    ADDFX2         1  5.9  149  +554     554 F
f2/s
f5/b
  g22/B                                  +0     554
  g22/CO   ADDHXL         1  6.2   99  +201     755 F
f5/cout
f6/a
  g63/CI                                 +0     755
  g63/CO   ADDFX2         1  6.2  145  +337    1092 F
f6/cout
f7/a
  g63/CI                                 +0    1092
  g63/CO   ADDFX2         1  5.9  144  +348    1439 F
f7/cout
f8/a
  g22/B                                  +0    1439
  g22/S    ADDHXL         1  0.0  143  +173    1612 F
f8/s
sum[3]     out port                      +0    1612 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : sum[3]
```

Fig 41. Timing Synthesis Report for Carry save adder using 180nm

6.1.4 PERFORMANCE EVALUATION:

Table 3. Comparison between the parameters of Carry Select adder, Carry Skip adder, Carry Save adder.

Fast Adders	Technology used	Type	Cells	Total Power (uW)	Total Delay(ps)
Carry Select Adder	45nm	Fast	4	1654.587	279
		Slow	4	1056.492	892
	90nm	Fast	4	4627.775	291
		Slow	4	2976.080	1057
	180nm	Fast	4	35393.882	597
		Slow	4	21659.847	1484
Carry Skip Adder	45nm	Fast	6	3398.523	150
		Slow	6	2240.899	1480
	90nm	Fast	6	5145.281	106
		Slow	6	3318.709	444
	180nm	Fast	6	36795.242	362
		Slow	6	19339.202	829
Carry Save Adder	45nm	Fast	8	4197.663	298
		Slow	8	2677.958	979
	90nm	Fast	9	8958.592	306
		Slow	9	5832.694	1170
	180nm	Fast	9	57347.523	682
		Slow	9	34730.669	1612

Comparison of Total No. of Cells:

Fast

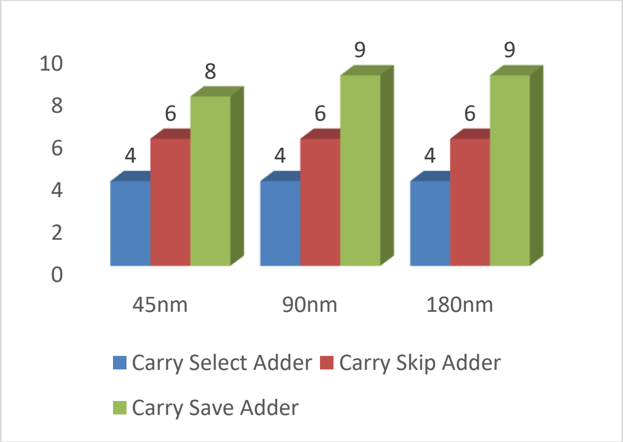


Fig 42. Comparison of Total No. of Cells Graph

Slow

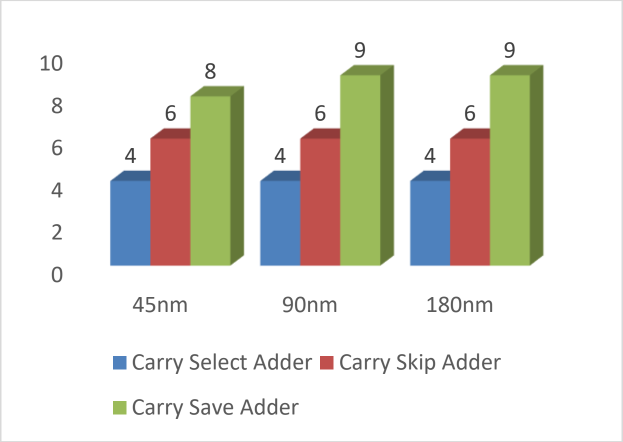


Fig 43. Comparison of Total No. of Cells Graph

Comparison of Total power consumption (in uW):

Fast

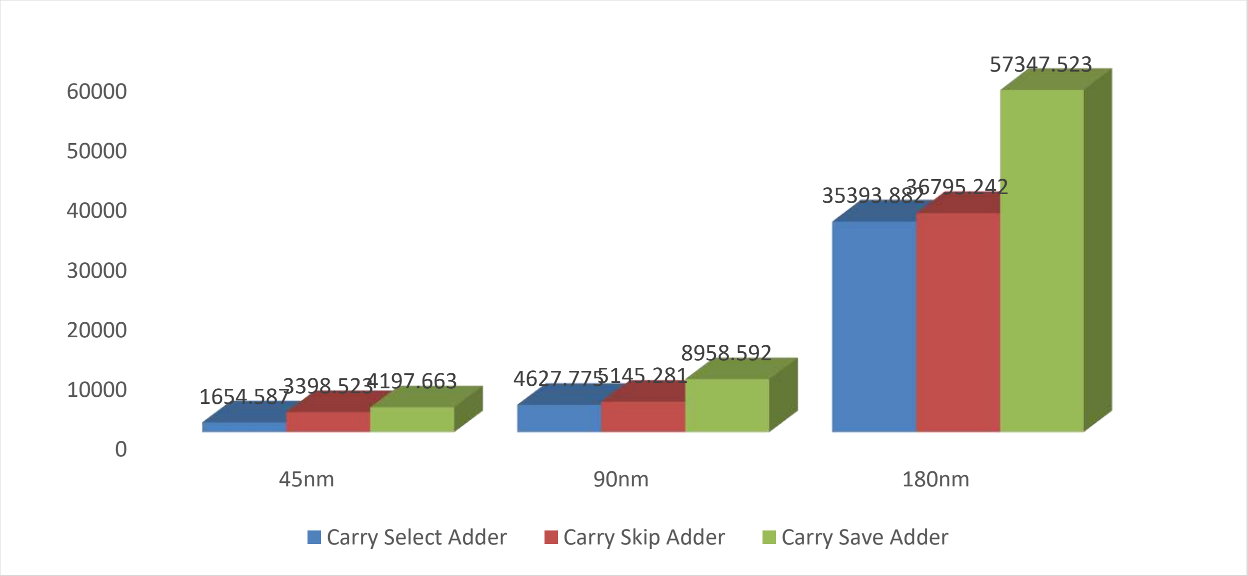


Fig 44. Comparison of Total Power Consumption Graph

Slow

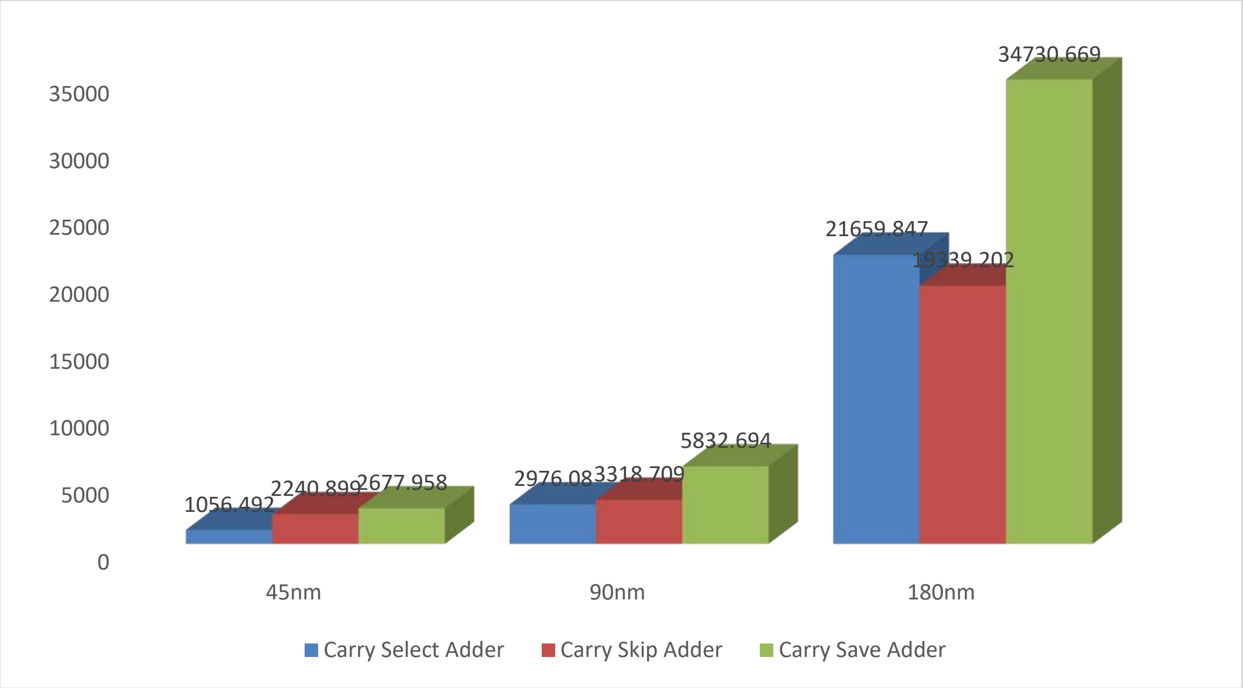


Fig 45. Comparison of Total Power Consumption Graph

Comparison of Total Delay (in ps):

Fast:

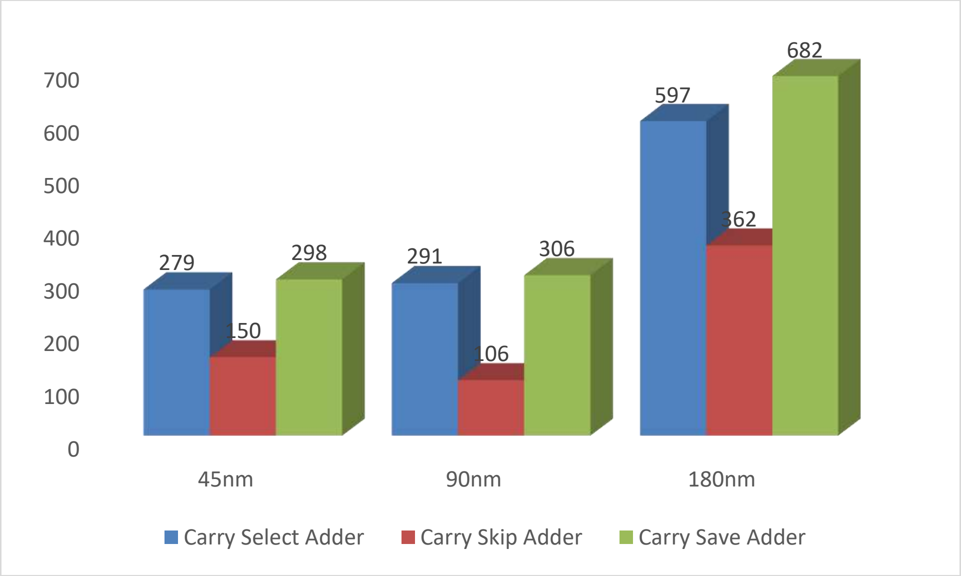


Fig 46. Comparison of Total Delay Graph

Slow:

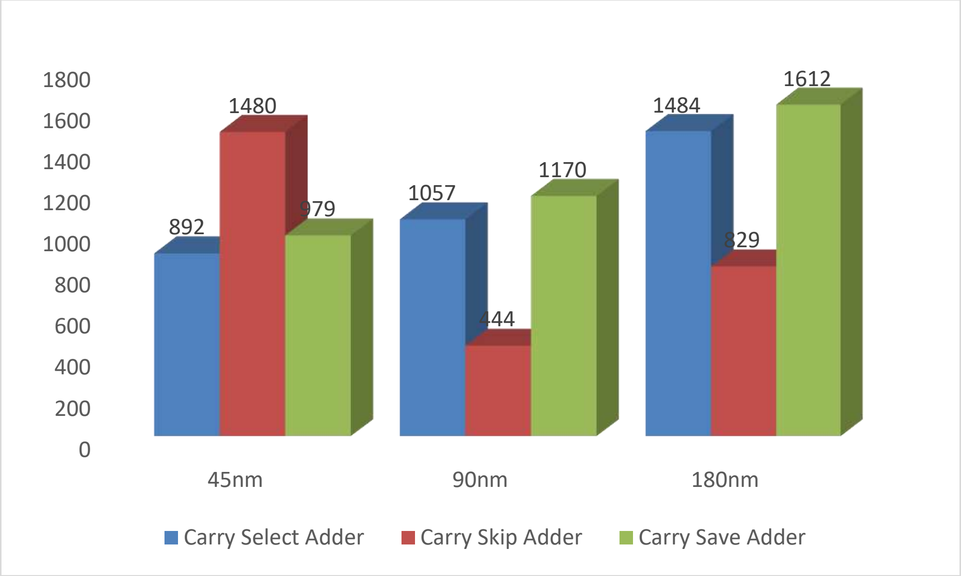


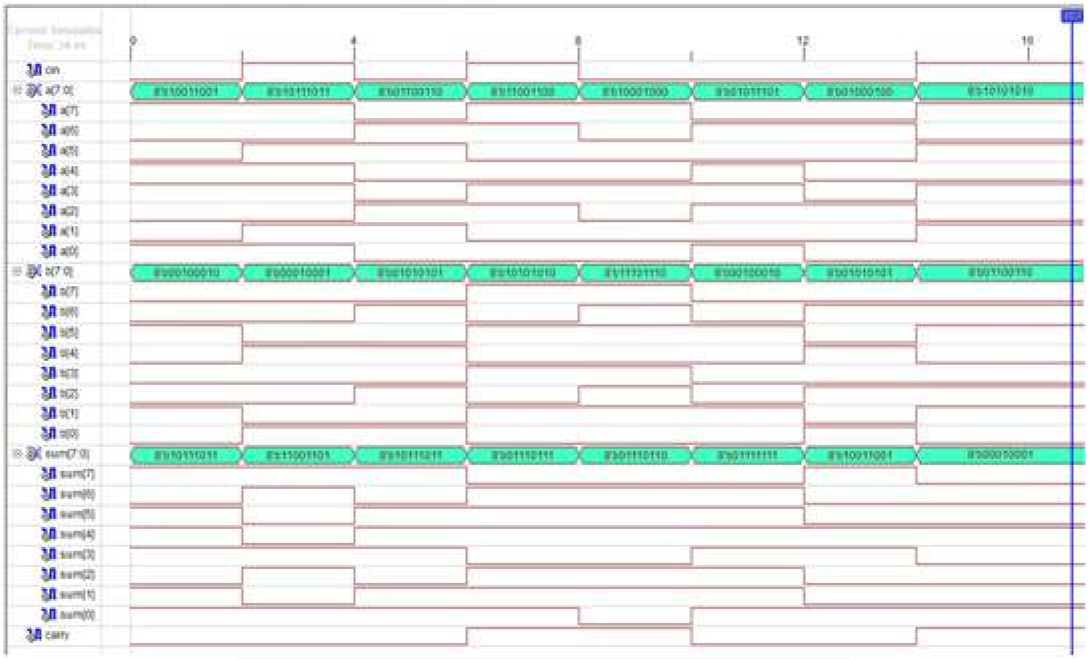
Fig 47. Comparison of Total Delay Graph

6.2 SIMULATION AND SYNTHESIS RESULTS OF 8BIT FAST ADDERS:

6.2.1.1 CARRY SELECT ADDER:

Carry select adder has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.

Fig 48. Output waveform of Carry select adder



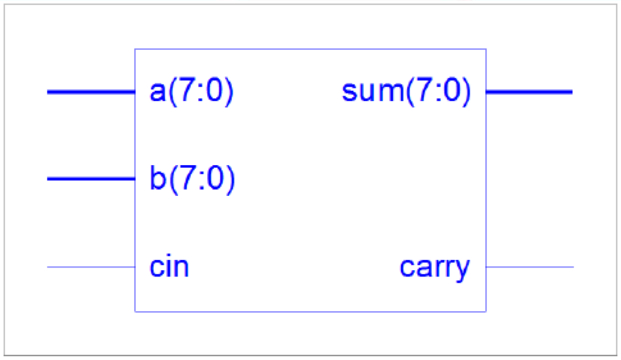


Fig 49. Carry select adder

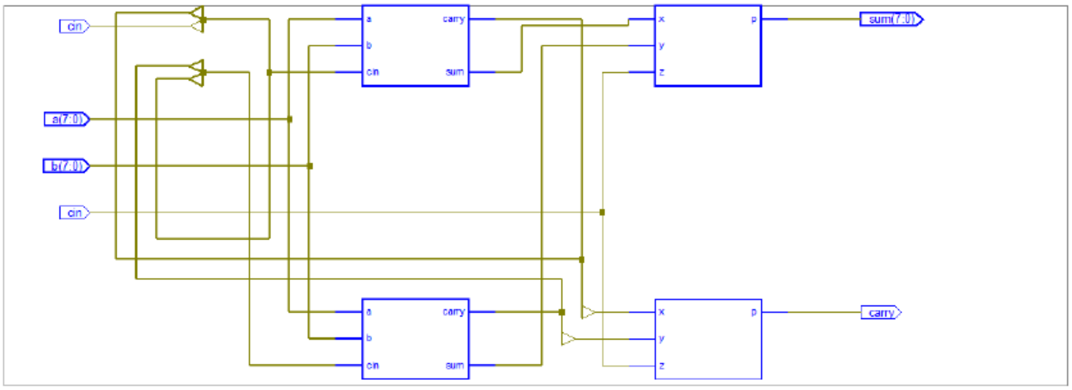


Fig 50. RTL view of carry select adder

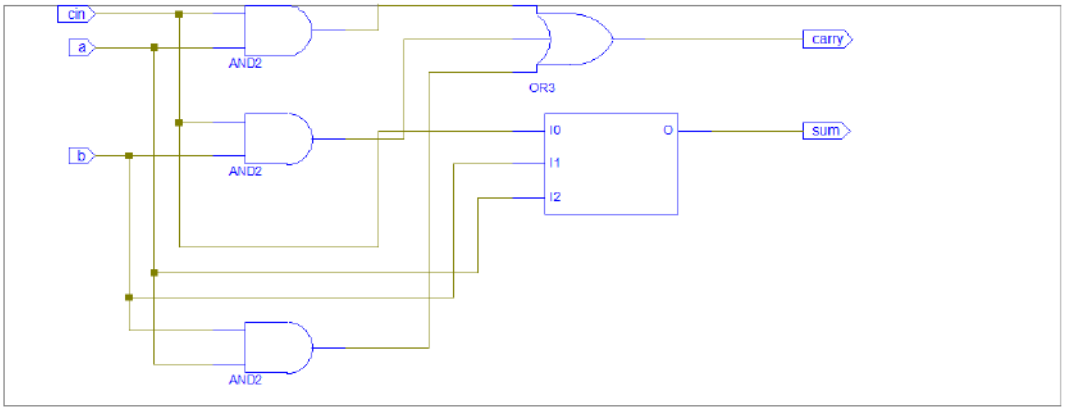


Fig 51. RTL view of full adder

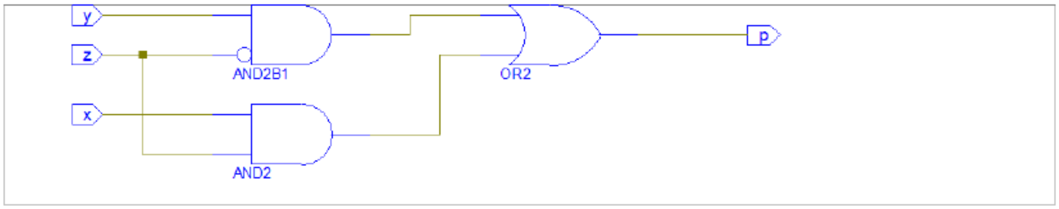


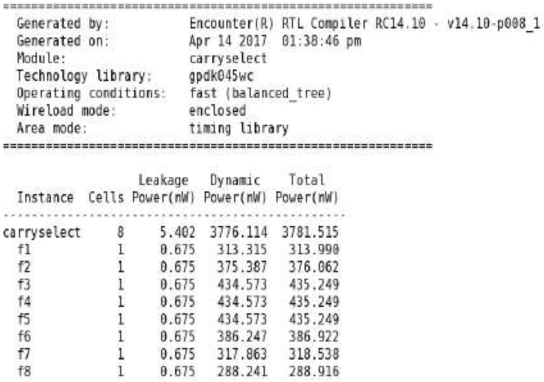
Fig 52. RTL view of multiplexer

6.2.1.2 POWER SYNTHESIS RESULT:

For the synthesis, here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

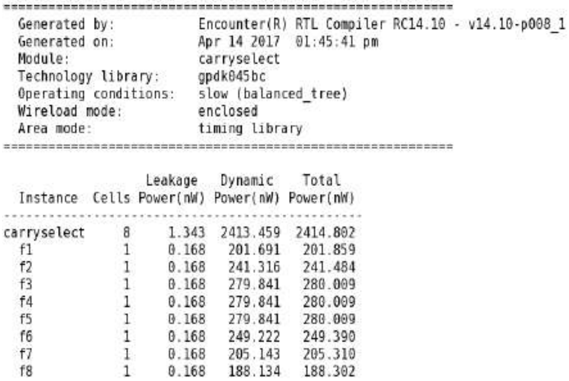
D. Using 45nm Technology

Fast.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  01:38:46 pm
  Module:                 carryselect
  Technology library:     gpdk045wc
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                     Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
carryselect     8     5.402  3776.114  3781.515
  f1            1     0.675   313.315   313.990
  f2            1     0.675   375.387   376.062
  f3            1     0.675   434.573   435.249
  f4            1     0.675   434.573   435.249
  f5            1     0.675   434.573   435.249
  f6            1     0.675   386.247   386.922
  f7            1     0.675   317.863   318.538
  f8            1     0.675   288.241   288.916
```

Slow.lib

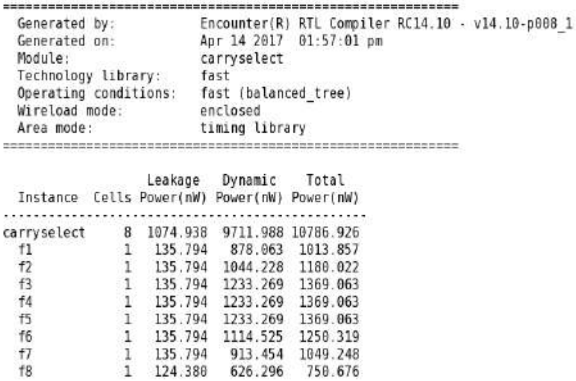


```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  01:45:41 pm
  Module:                 carryselect
  Technology library:     gpdk045bc
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                     Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
carryselect     8     1.343  2413.459  2414.802
  f1            1     0.168   201.691   201.859
  f2            1     0.168   241.316   241.484
  f3            1     0.168   279.841   280.009
  f4            1     0.168   279.841   280.009
  f5            1     0.168   279.841   280.009
  f6            1     0.168   249.222   249.390
  f7            1     0.168   205.143   205.310
  f8            1     0.168   188.134   188.302
```

Fig 53. Power synthesis report for Carry select adder using 45nm

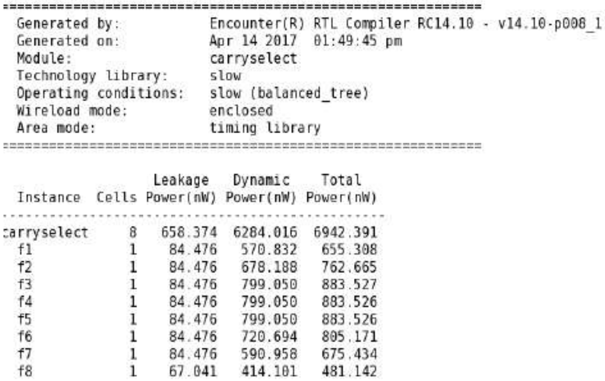
E. Using 90nm Technology

Fast.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  01:57:01 pm
  Module:                 carryselect
  Technology library:     fast
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                     Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
carryselect     8  1074.938  9711.988 10786.926
  f1            1   135.794   878.063  1013.857
  f2            1   135.794  1044.228  1180.022
  f3            1   135.794  1233.269  1369.063
  f4            1   135.794  1233.269  1369.063
  f5            1   135.794  1233.269  1369.063
  f6            1   135.794  1114.525  1250.319
  f7            1   135.794   913.454  1049.248
  f8            1   124.380   626.296   750.676
```

Slow.lib

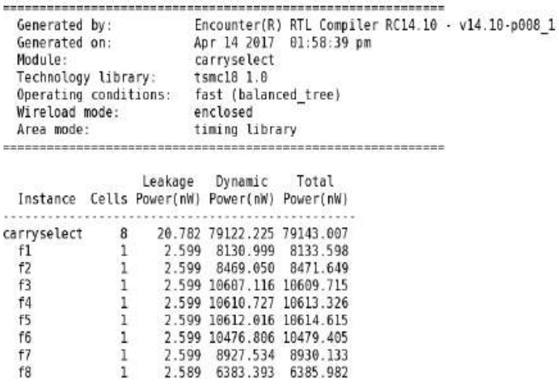


```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  01:49:45 pm
  Module:                 carryselect
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                     Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
carryselect     8   658.374  6284.016  6942.391
  f1            1    84.476   570.832   655.308
  f2            1    84.476   678.188   762.665
  f3            1    84.476   799.050   883.527
  f4            1    84.476   799.050   883.526
  f5            1    84.476   799.050   883.526
  f6            1    84.476   720.694   805.171
  f7            1    84.476   590.958   675.434
  f8            1    67.041   414.101   481.142
```

Fig 54. Power synthesis report for Carry select adder using 90nm

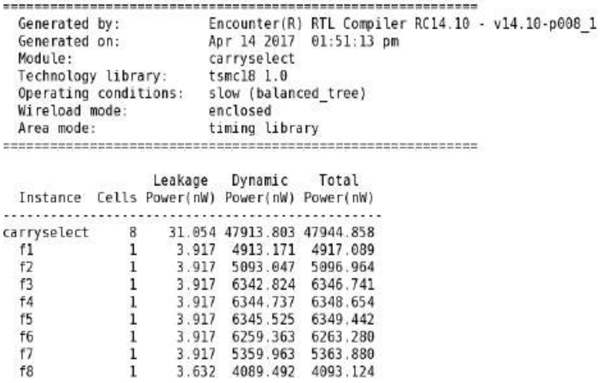
F. Using 180nm Technology:

Fast.lib:



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  01:58:39 pm
  Module:                 carryselect
  Technology library:     tsmc18 1.0
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                   Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
-----------------------------------------------
carryselect     8    20.782 79122.225 79143.007
  f1            1     2.599  8130.999  8133.598
  f2            1     2.599  8469.050  8471.649
  f3            1     2.599 10607.116 10609.715
  f4            1     2.599 10610.727 10613.326
  f5            1     2.599 10612.016 10614.615
  f6            1     2.599 10476.806 10479.405
  f7            1     2.599  8927.534  8930.133
  f8            1     2.589  6383.393  6385.982
```

Slow.lib:



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  01:51:13 pm
  Module:                 carryselect
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                   Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
-----------------------------------------------
carryselect     8    31.054 47913.803 47944.858
  f1            1     3.917  4913.171  4917.089
  f2            1     3.917  5093.047  5096.964
  f3            1     3.917  6342.824  6346.741
  f4            1     3.917  6344.737  6348.654
  f5            1     3.917  6345.525  6349.442
  f6            1     3.917  6259.363  6263.280
  f7            1     3.917  5359.963  5363.880
  f8            1     3.632  4089.492  4093.124
```

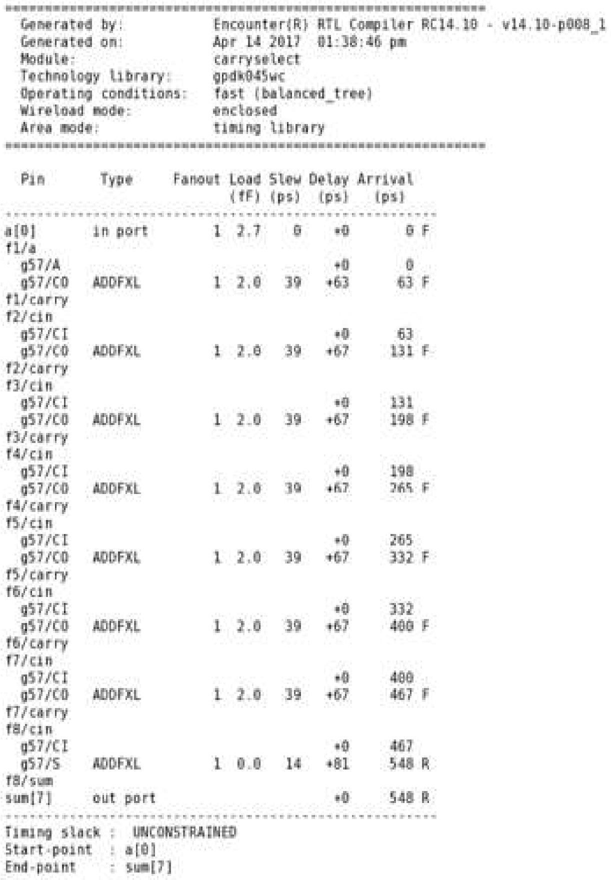
Fig 55. Power synthesis report for Carry select adder using 180nm

6.2.1.3 DELAY AND TIMING SYNTHESIS RESULTS:

Same as in case of power synthesis, here we are using here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

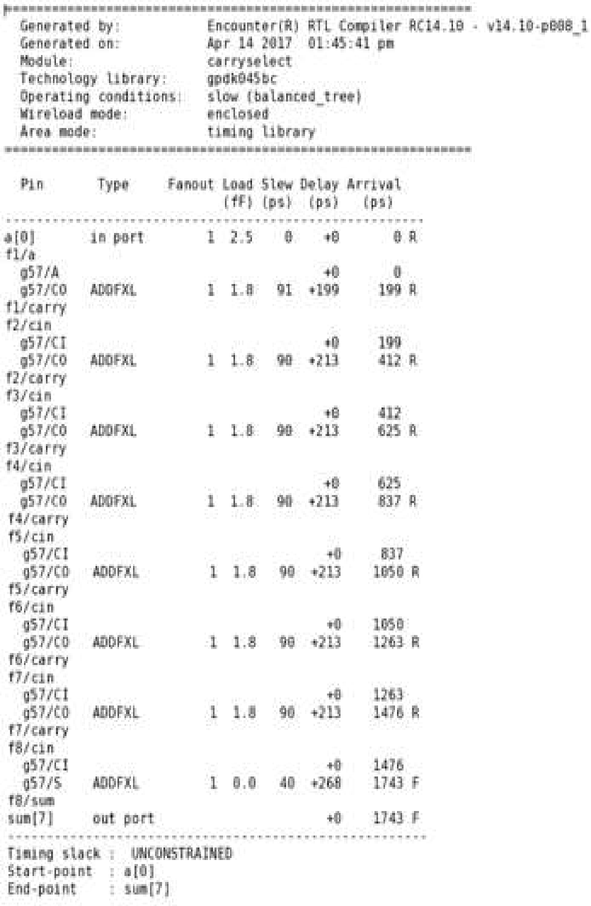
D. Using 45nm Technology

Fast.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:38:46 pm
  Module:                  carryselect
  Technology library:      gpdk045wc
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[0]       in port       1  2.7    0    +0       0 F
f1/a
  g57/A                                 +0       0
  g57/CO   ADDFXL        1  2.0   39   +63      63 F
f1/carry
f2/cin
  g57/CI                                +0      63
  g57/CO   ADDFXL        1  2.0   39   +67     131 F
f2/carry
f3/cin
  g57/CI                                +0     131
  g57/CO   ADDFXL        1  2.0   39   +67     198 F
f3/carry
f4/cin
  g57/CI                                +0     198
  g57/CO   ADDFXL        1  2.0   39   +67     265 F
f4/carry
f5/cin
  g57/CI                                +0     265
  g57/CO   ADDFXL        1  2.0   39   +67     332 F
f5/carry
f6/cin
  g57/CI                                +0     332
  g57/CO   ADDFXL        1  2.0   39   +67     400 F
f6/carry
f7/cin
  g57/CI                                +0     400
  g57/CO   ADDFXL        1  2.0   39   +67     467 F
f7/carry
f8/cin
  g57/CI                                +0     467
  g57/S    ADDFXL        1  0.0   14   +81     548 R
f8/sum
sum[7]     out port                     +0     548 R
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[7]
```

Slow.lib



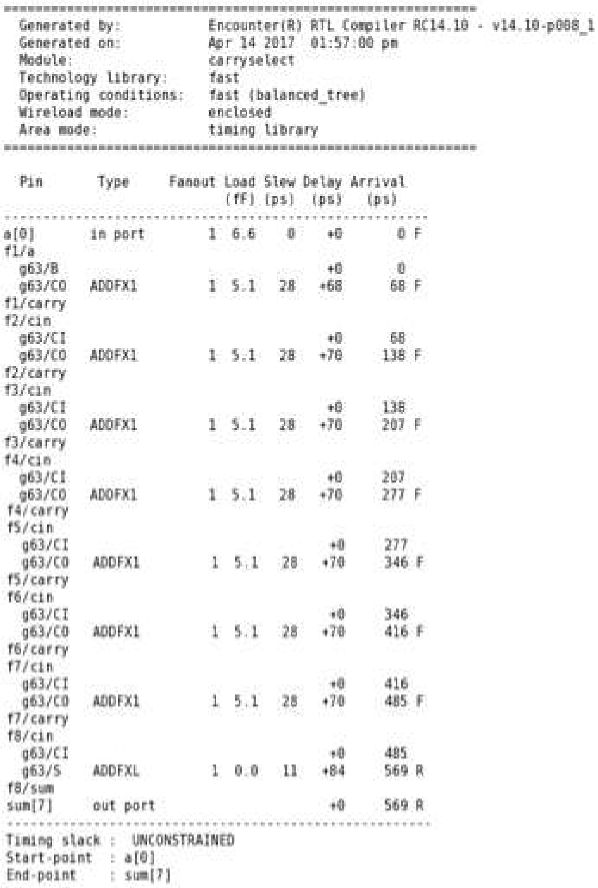
```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:45:41 pm
  Module:                  carryselect
  Technology library:      gpdk045bc
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[0]       in port       1  2.5    0    +0       0 R
f1/a
  g57/A                                 +0       0
  g57/CO   ADDFXL        1  1.8   91  +199     199 R
f1/carry
f2/cin
  g57/CI                                +0     199
  g57/CO   ADDFXL        1  1.8   90  +213     412 R
f2/carry
f3/cin
  g57/CI                                +0     412
  g57/CO   ADDFXL        1  1.8   90  +213     625 R
f3/carry
f4/cin
  g57/CI                                +0     625
  g57/CO   ADDFXL        1  1.8   90  +213     837 R
f4/carry
f5/cin
  g57/CI                                +0     837
  g57/CO   ADDFXL        1  1.8   90  +213    1050 R
f5/carry
f6/cin
  g57/CI                                +0    1050
  g57/CO   ADDFXL        1  1.8   90  +213    1263 R
f6/carry
f7/cin
  g57/CI                                +0    1263
  g57/CO   ADDFXL        1  1.8   90  +213    1476 R
f7/carry
f8/cin
  g57/CI                                +0    1476
  g57/S    ADDFXL        1  0.0   40  +268    1743 F
f8/sum
sum[7]     out port                     +0    1743 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[7]
```

Fig 56. Timing Synthesis Report for Carry select adder using 45nm

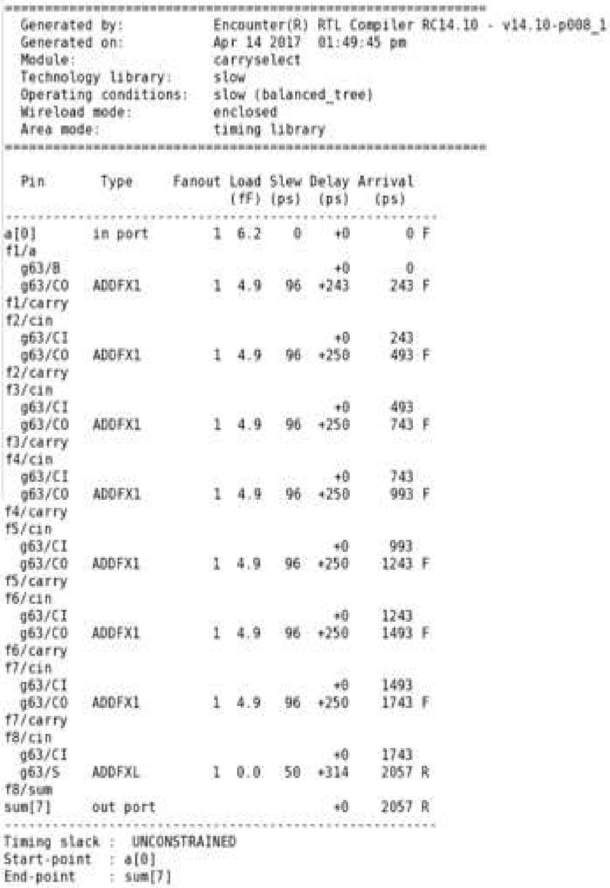
E. Using 90nm Technology

Fast.lib

Slow.lib



```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  01:57:00 pm
  Module:                 carryselect
  Technology library:     fast
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-----------------------------------------------------
a[0]       in port        1  6.6    0    +0       0 F
f1/a
  g63/B                                  +0       0
  g63/CO   ADDFX1         1  5.1   28   +68      68 F
f1/carry
f2/cin
  g63/CI                                 +0      68
  g63/CO   ADDFX1         1  5.1   28   +70     138 F
f2/carry
f3/cin
  g63/CI                                 +0     138
  g63/CO   ADDFX1         1  5.1   28   +70     207 F
f3/carry
f4/cin
  g63/CI                                 +0     207
  g63/CO   ADDFX1         1  5.1   28   +70     277 F
f4/carry
f5/cin
  g63/CI                                 +0     277
  g63/CO   ADDFX1         1  5.1   28   +70     346 F
f5/carry
f6/cin
  g63/CI                                 +0     346
  g63/CO   ADDFX1         1  5.1   28   +70     416 F
f6/carry
f7/cin
  g63/CI                                 +0     416
  g63/CO   ADDFX1         1  5.1   28   +70     485 F
f7/carry
f8/cin
  g63/CI                                 +0     485
  g63/S    ADDFXL         1  0.0   11   +84     569 R
f8/sum
sum[7]     out port                      +0     569 R
-----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[7]
```

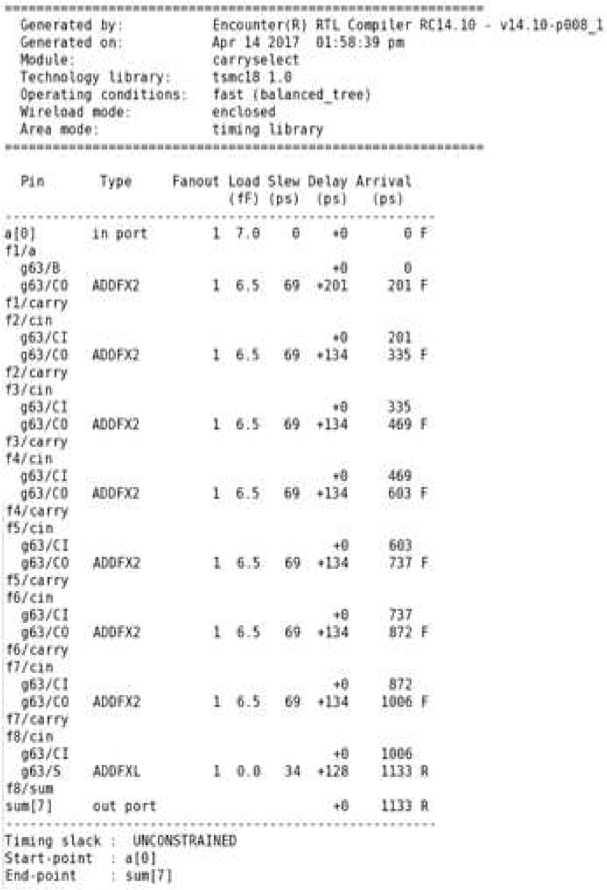


```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  01:49:45 pm
  Module:                 carryselect
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin       Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-----------------------------------------------------
a[0]       in port        1  6.2    0    +0       0 F
f1/a
  g63/B                                  +0       0
  g63/CO   ADDFX1         1  4.9   96  +243     243 F
f1/carry
f2/cin
  g63/CI                                 +0     243
  g63/CO   ADDFX1         1  4.9   96  +250     493 F
f2/carry
f3/cin
  g63/CI                                 +0     493
  g63/CO   ADDFX1         1  4.9   96  +250     743 F
f3/carry
f4/cin
  g63/CI                                 +0     743
  g63/CO   ADDFX1         1  4.9   96  +250     993 F
f4/carry
f5/cin
  g63/CI                                 +0     993
  g63/CO   ADDFX1         1  4.9   96  +250    1243 F
f5/carry
f6/cin
  g63/CI                                 +0    1243
  g63/CO   ADDFX1         1  4.9   96  +250    1493 F
f6/carry
f7/cin
  g63/CI                                 +0    1493
  g63/CO   ADDFX1         1  4.9   96  +250    1743 F
f7/carry
f8/cin
  g63/CI                                 +0    1743
  g63/S    ADDFXL         1  0.0   50  +314    2057 R
f8/sum
sum[7]     out port                      +0    2057 R
-----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[7]
```

Fig 57. Timing Synthesis Report for Carry select adder using 90nm

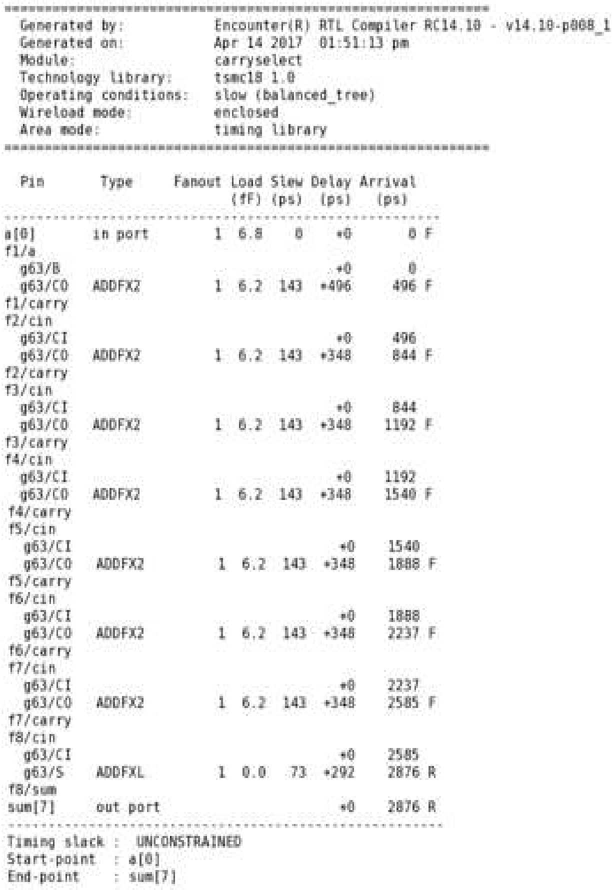
F. Using 180nm Technology:

Fast.lib



```
==============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:58:39 pm
  Module:                  carryselect
  Technology library:      tsmc18 1.0
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
==============================================================

  Pin       Type      Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
--------------------------------------------------------
a[0]       in port         1  7.0    0    +0      0 F
f1/a
  g63/B                               +0      0
  g63/CO   ADDFX2          1  6.5   69  +201    201 F
f1/carry
f2/cin
  g63/CI                              +0    201
  g63/CO   ADDFX2          1  6.5   69  +134    335 F
f2/carry
f3/cin
  g63/CI                              +0    335
  g63/CO   ADDFX2          1  6.5   69  +134    469 F
f3/carry
f4/cin
  g63/CI                              +0    469
  g63/CO   ADDFX2          1  6.5   69  +134    603 F
f4/carry
f5/cin
  g63/CI                              +0    603
  g63/CO   ADDFX2          1  6.5   69  +134    737 F
f5/carry
f6/cin
  g63/CI                              +0    737
  g63/CO   ADDFX2          1  6.5   69  +134    872 F
f6/carry
f7/cin
  g63/CI                              +0    872
  g63/CO   ADDFX2          1  6.5   69  +134   1006 F
f7/carry
f8/cin
  g63/CI                              +0   1006
  g63/S    ADDFXL          1  0.0   34  +128   1133 R
f8/sum
sum[7]     out port                   +0   1133 R
--------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[7]
```

Slow.lib



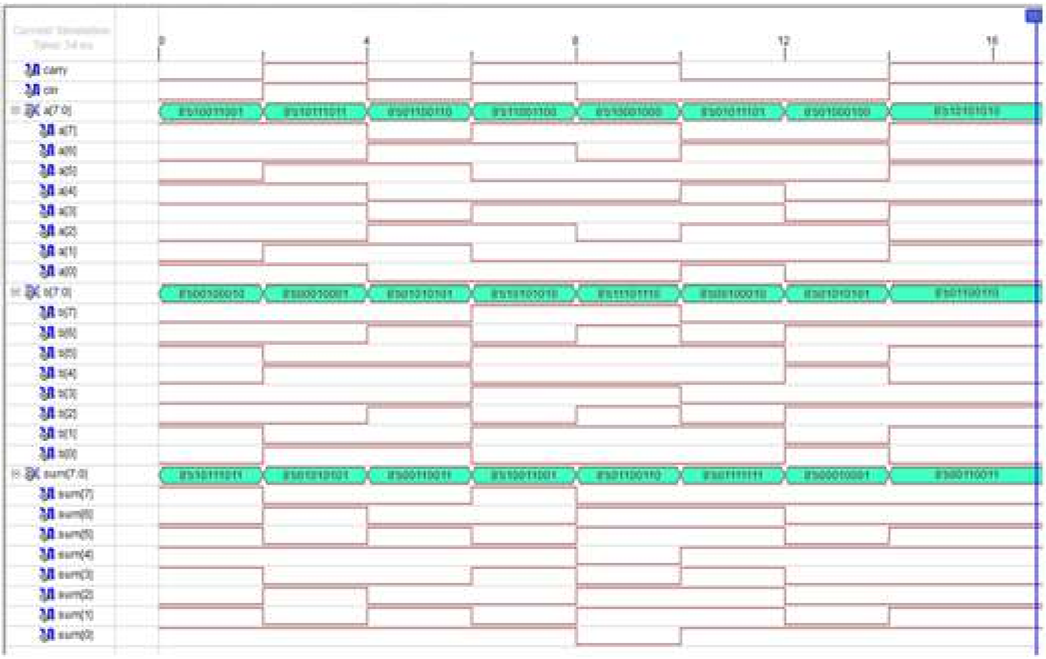
```
==============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:51:13 pm
  Module:                  carryselect
  Technology library:      tsmc18 1.0
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
==============================================================

  Pin       Type      Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
--------------------------------------------------------
a[0]       in port         1  6.8    0    +0      0 F
f1/a
  g63/B                               +0      0
  g63/CO   ADDFX2          1  6.2  143  +496    496 F
f1/carry
f2/cin
  g63/CI                              +0    496
  g63/CO   ADDFX2          1  6.2  143  +348    844 F
f2/carry
f3/cin
  g63/CI                              +0    844
  g63/CO   ADDFX2          1  6.2  143  +348   1192 F
f3/carry
f4/cin
  g63/CI                              +0   1192
  g63/CO   ADDFX2          1  6.2  143  +348   1540 F
f4/carry
f5/cin
  g63/CI                              +0   1540
  g63/CO   ADDFX2          1  6.2  143  +348   1888 F
f5/carry
f6/cin
  g63/CI                              +0   1888
  g63/CO   ADDFX2          1  6.2  143  +348   2237 F
f6/carry
f7/cin
  g63/CI                              +0   2237
  g63/CO   ADDFX2          1  6.2  143  +348   2585 F
f7/carry
f8/cin
  g63/CI                              +0   2585
  g63/S    ADDFXL          1  0.0   73  +292   2876 R
f8/sum
sum[7]     out port                   +0   2876 R
--------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : sum[7]
```

Fig 58. Timing Synthesis Report for Carry select adder using 180nm

6.2.2.1 CARRY SKIP ADDER:

It has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.

Fig 59. Output waveform of Carry Skip Adder



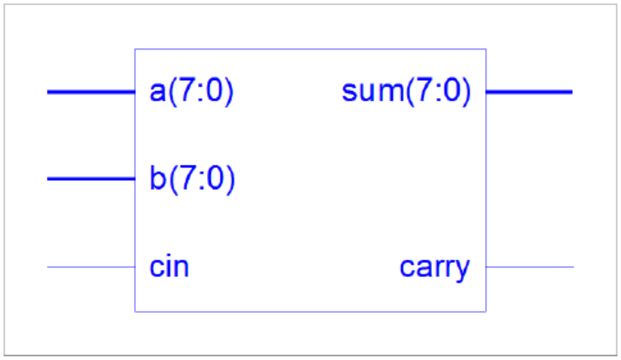


Fig 60. Carry skip adder

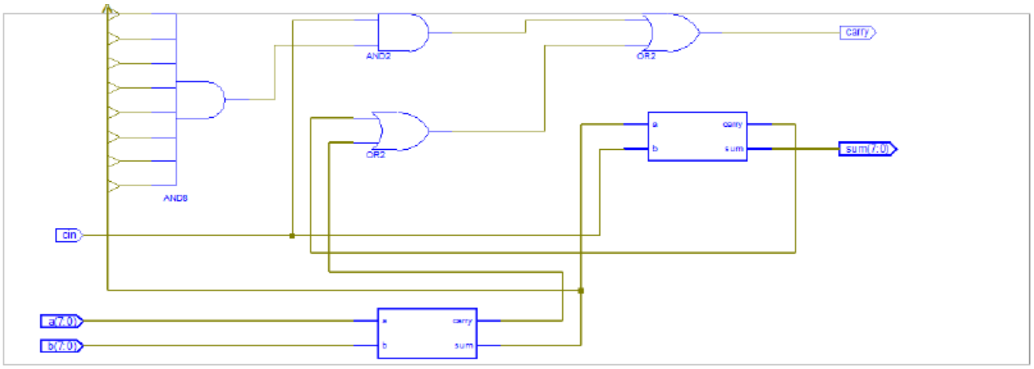


Fig 61. RTL view of carry skip adder

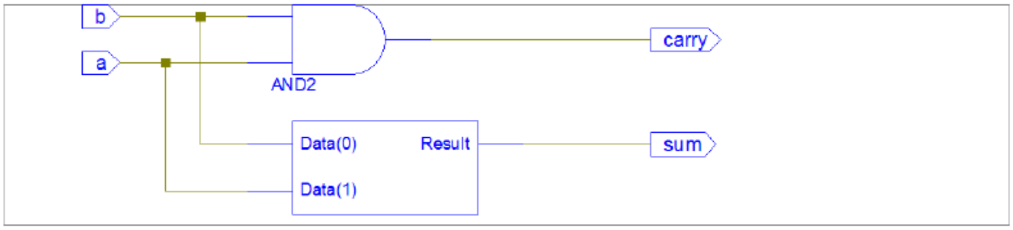


Fig 62. RTL view of half adder

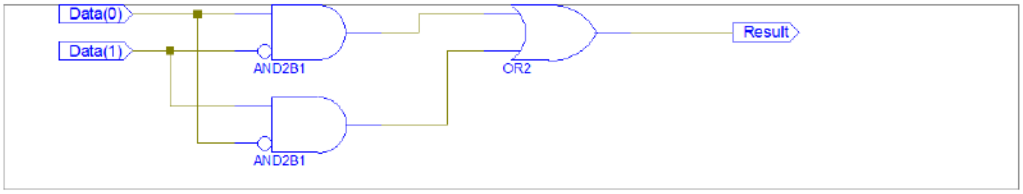


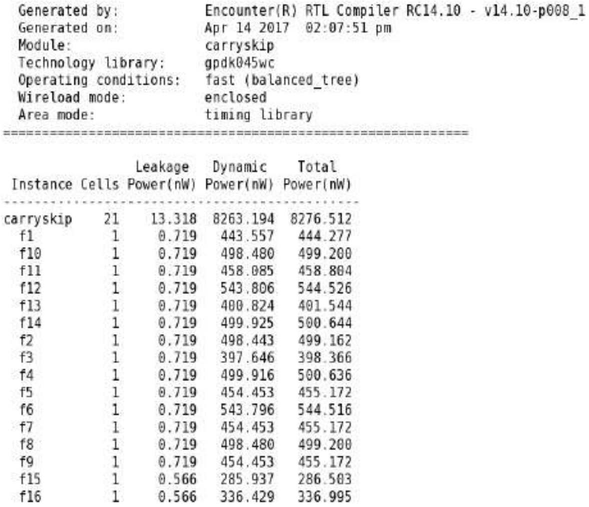
Fig 63. RTL view of XOR gate

6.2.2.2 POWER SYNTHESIS RESULT:

For the synthesis, here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

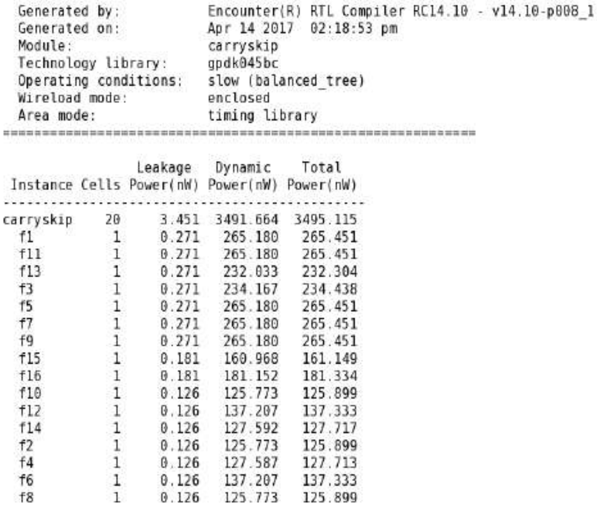
D. Using 45nm Technology

Fast.lib:



```
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  02:07:51 pm
  Module:                  carryskip
  Technology library:      gpdk045wc
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                   Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip    21    13.318  8263.194  8276.512
  f1          1     0.719   443.557   444.277
  f10         1     0.719   498.480   499.200
  f11         1     0.719   458.085   458.804
  f12         1     0.719   543.806   544.526
  f13         1     0.719   400.824   401.544
  f14         1     0.719   499.925   500.644
  f2          1     0.719   498.443   499.162
  f3          1     0.719   397.646   398.366
  f4          1     0.719   499.916   500.636
  f5          1     0.719   454.453   455.172
  f6          1     0.719   543.796   544.516
  f7          1     0.719   454.453   455.172
  f8          1     0.719   498.480   499.200
  f9          1     0.719   454.453   455.172
  f15         1     0.566   285.937   286.503
  f16         1     0.566   336.429   336.995
```

Slow.lib:



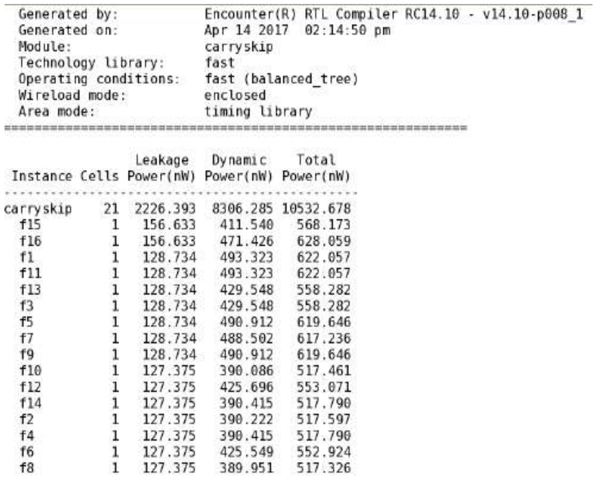
```
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  02:18:53 pm
  Module:                  carryskip
  Technology library:      gpdk045bc
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                   Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip    20     3.451  3491.664  3495.115
  f1          1     0.271   265.180   265.451
  f11         1     0.271   265.180   265.451
  f13         1     0.271   232.033   232.304
  f3          1     0.271   234.167   234.438
  f5          1     0.271   265.180   265.451
  f7          1     0.271   265.180   265.451
  f9          1     0.271   265.180   265.451
  f15         1     0.181   160.968   161.149
  f16         1     0.181   181.152   181.334
  f10         1     0.126   125.773   125.899
  f12         1     0.126   137.207   137.333
  f14         1     0.126   127.592   127.717
  f2          1     0.126   125.773   125.899
  f4          1     0.126   127.587   127.713
  f6          1     0.126   137.207   137.333
  f8          1     0.126   125.773   125.899
```

Fig 64. Power synthesis report for Carry skip adder using 45nm

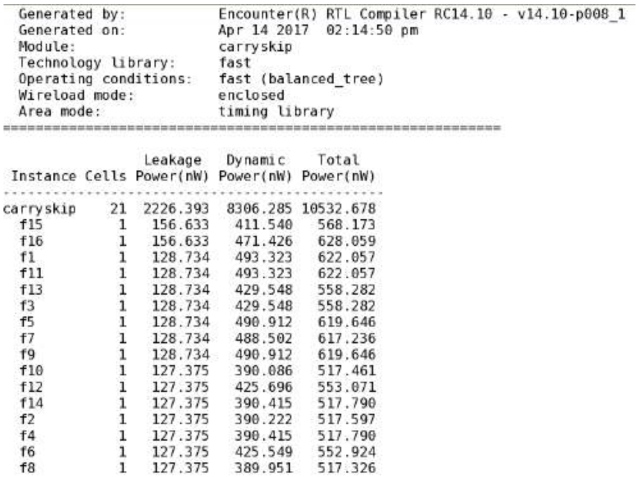
E. Using 90nm Technology

Fast.lib

Slow.lib



```
 Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
 Generated on:            Apr 14 2017  02:14:50 pm
 Module:                  carryskip
 Technology library:      fast
 Operating conditions:    fast (balanced_tree)
 Wireload mode:           enclosed
 Area mode:               timing library
============================================================

                Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip    21  2226.393  8306.285 10532.678
  f15         1   156.633   411.540   568.173
  f16         1   156.633   471.426   628.059
  f1          1   128.734   493.323   622.057
  f11         1   128.734   493.323   622.057
  f13         1   128.734   429.548   558.282
  f3          1   128.734   429.548   558.282
  f5          1   128.734   490.912   619.646
  f7          1   128.734   488.502   617.236
  f9          1   128.734   490.912   619.646
  f10         1   127.375   390.086   517.461
  f12         1   127.375   425.696   553.071
  f14         1   127.375   390.415   517.790
  f2          1   127.375   390.222   517.597
  f4          1   127.375   390.415   517.790
  f6          1   127.375   425.549   552.924
  f8          1   127.375   389.951   517.326
```



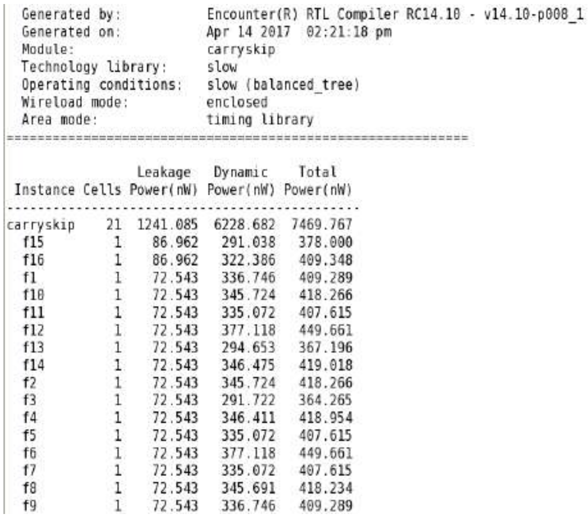
```
 Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
 Generated on:            Apr 14 2017  02:14:50 pm
 Module:                  carryskip
 Technology library:      fast
 Operating conditions:    fast (balanced_tree)
 Wireload mode:           enclosed
 Area mode:               timing library
============================================================

                Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip    21  2226.393  8306.285 10532.678
  f15         1   156.633   411.540   568.173
  f16         1   156.633   471.426   628.059
  f1          1   128.734   493.323   622.057
  f11         1   128.734   493.323   622.057
  f13         1   128.734   429.548   558.282
  f3          1   128.734   429.548   558.282
  f5          1   128.734   490.912   619.646
  f7          1   128.734   488.502   617.236
  f9          1   128.734   490.912   619.646
  f10         1   127.375   390.086   517.461
  f12         1   127.375   425.696   553.071
  f14         1   127.375   390.415   517.790
  f2          1   127.375   390.222   517.597
  f4          1   127.375   390.415   517.790
  f6          1   127.375   425.549   552.924
  f8          1   127.375   389.951   517.326
```

Fig 65. Power synthesis report for Carry skip adder using 90nm

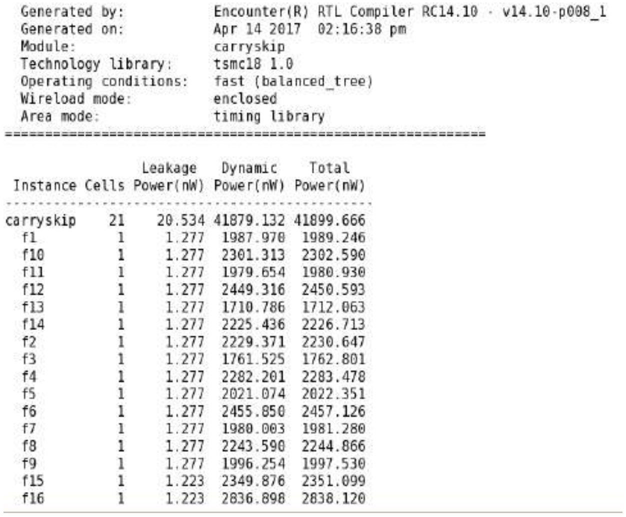
F. Using 180nm Technology:

Fast.lib:

Slow.lib:



```
 Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
 Generated on:            Apr 14 2017  02:21:18 pm
 Module:                  carryskip
 Technology library:      slow
 Operating conditions:    slow (balanced_tree)
 Wireload mode:           enclosed
 Area mode:               timing library
============================================================

                Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip    21  1241.085  6228.682  7469.767
  f15         1    86.962   291.038   378.000
  f16         1    86.962   322.386   409.348
  f1          1    72.543   336.746   409.289
  f10         1    72.543   345.724   418.266
  f11         1    72.543   335.072   407.615
  f12         1    72.543   377.118   449.661
  f13         1    72.543   294.653   367.196
  f14         1    72.543   346.475   419.018
  f2          1    72.543   345.724   418.266
  f3          1    72.543   291.722   364.265
  f4          1    72.543   346.411   418.954
  f5          1    72.543   335.072   407.615
  f6          1    72.543   377.118   449.661
  f7          1    72.543   335.072   407.615
  f8          1    72.543   345.691   418.234
  f9          1    72.543   336.746   409.289
```



```
 Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
 Generated on:            Apr 14 2017  02:16:38 pm
 Module:                  carryskip
 Technology library:      tsmc18 1.0
 Operating conditions:    fast (balanced_tree)
 Wireload mode:           enclosed
 Area mode:               timing library
============================================================

                Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------
carryskip    21    20.534 41879.132 41899.666
  f1          1     1.277  1987.970  1989.246
  f10         1     1.277  2301.313  2302.590
  f11         1     1.277  1979.654  1980.930
  f12         1     1.277  2449.316  2450.593
  f13         1     1.277  1710.786  1712.063
  f14         1     1.277  2225.436  2226.713
  f2          1     1.277  2229.371  2230.647
  f3          1     1.277  1761.525  1762.801
  f4          1     1.277  2282.201  2283.478
  f5          1     1.277  2021.074  2022.351
  f6          1     1.277  2455.850  2457.126
  f7          1     1.277  1980.003  1981.280
  f8          1     1.277  2243.590  2244.866
  f9          1     1.277  1996.254  1997.530
  f15         1     1.223  2349.876  2351.099
  f16         1     1.223  2836.898  2838.120
```

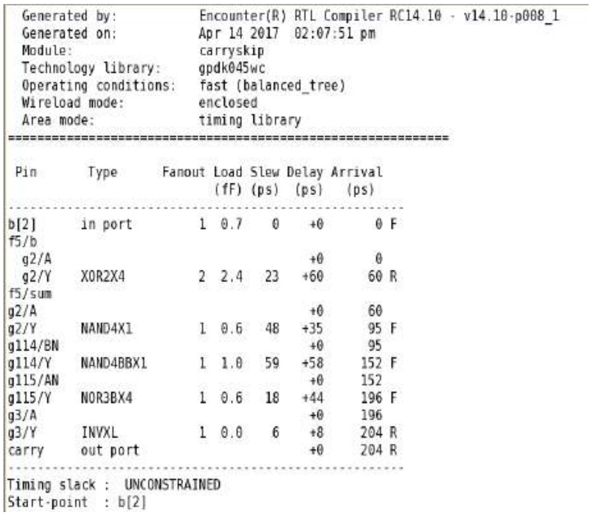
Fig 66. Power synthesis report for Carry skip adder using 180nm

6.2.2.3 DELAY AND TIMING SYNTHESIS RESULTS:

Same as in case of power synthesis, here we are using here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

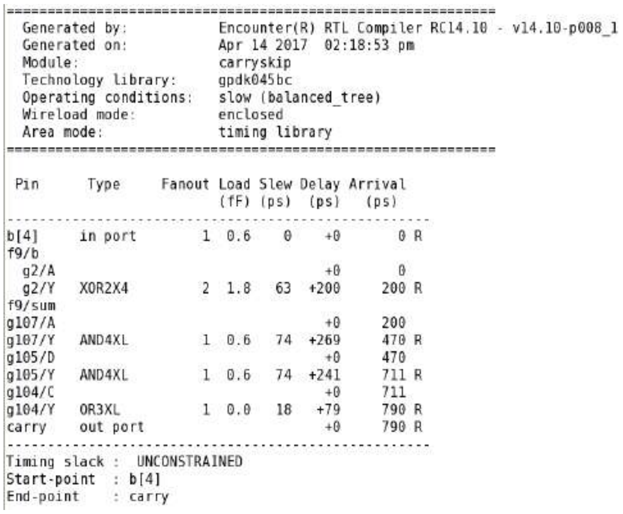
B. Using 45nm Technology

Fast.lib



```
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  02:07:51 pm
  Module:                  carryskip
  Technology library:      gpdk045wc
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

 Pin       Type      Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
----------------------------------------------------
b[2]      in port        1  0.7    0    +0       0 F
f5/b
  g2/A                                  +0       0
  g2/Y    XOR2X4         2  2.4   23   +60      60 R
f5/sum
g2/A                                    +0      60
g2/Y      NAND4X1        1  0.6   48   +35      95 F
g114/BN                                 +0      95
g114/Y    NAND4BBX1      1  1.0   59   +58     152 F
g115/AN                                 +0     152
g115/Y    NOR3BX4        1  0.6   18   +44     196 F
g3/A                                    +0     196
g3/Y      INVXL          1  0.0    6    +8     204 R
carry     out port                      +0     204 R
----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[2]
```

Slow.lib



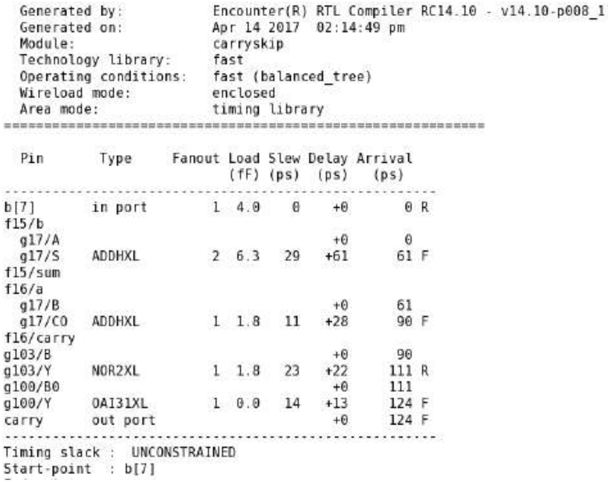
```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  02:18:53 pm
  Module:                  carryskip
  Technology library:      gpdk045bc
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

 Pin       Type      Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
----------------------------------------------------
b[4]      in port        1  0.6    0    +0       0 R
f9/b
  g2/A                                  +0       0
  g2/Y    XOR2X4         2  1.8   63  +200     200 R
f9/sum
g107/A                                  +0     200
g107/Y    AND4XL         1  0.6   74  +269     470 R
g105/D                                  +0     470
g105/Y    AND4XL         1  0.6   74  +241     711 R
g104/C                                  +0     711
g104/Y    OR3XL          1  0.0   18   +79     790 R
carry     out port                      +0     790 R
----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[4]
End-point    : carry
```

Fig 67. Timing Synthesis Report for Carry skip adder using 45nm

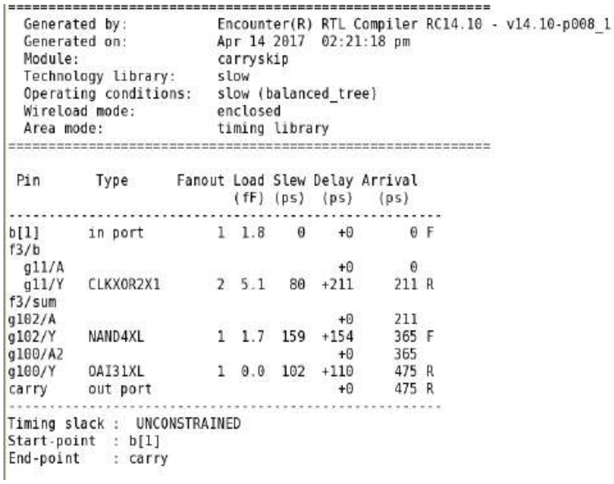
C. Using 90nm Technology

Fast.lib

Slow.lib



```
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  02:14:49 pm
  Module:                  carryskip
  Technology library:      fast
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
-----------------------------------------------------
b[7]        in port         1  4.0    0    +0       0 R
f15/b
  g17/A                                    +0       0
  g17/S     ADDHXL          2  6.3   29   +61      61 F
f15/sum
f16/a
  g17/B                                    +0      61
  g17/CO    ADDHXL          1  1.8   11   +28      90 F
f16/carry
g103/B                                     +0      90
g103/Y      NOR2XL          1  1.8   23   +22     111 R
g100/B0                                    +0     111
g100/Y      OAI31XL         1  0.0   14   +13     124 F
carry       out port                       +0     124 F
-----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[7]
```



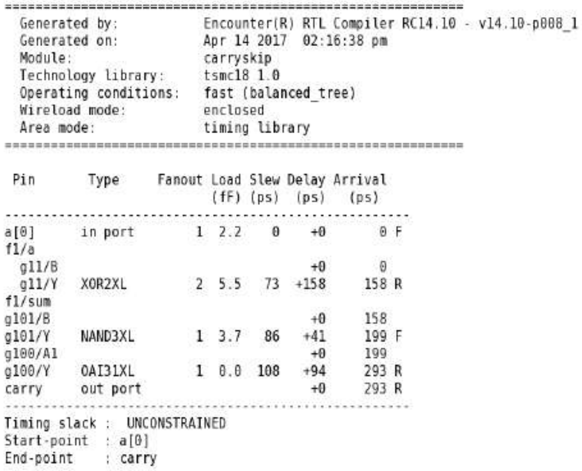
```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  02:21:18 pm
  Module:                  carryskip
  Technology library:      slow
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

 Pin        Type      Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
-----------------------------------------------------
b[1]       in port         1  1.8    0    +0       0 F
f3/b
  g11/A                                   +0       0
  g11/Y    CLKXOR2X1       2  5.1   80  +211     211 R
f3/sum
g102/A                                    +0     211
g102/Y     NAND4XL         1  1.7  159  +154     365 F
g100/A2                                   +0     365
g100/Y     OAI31XL         1  0.0  102  +110     475 R
carry      out port                       +0     475 R
-----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[1]
End-point    : carry
```

Fig 68. Timing Synthesis Report for Carry skip adder using 90nm

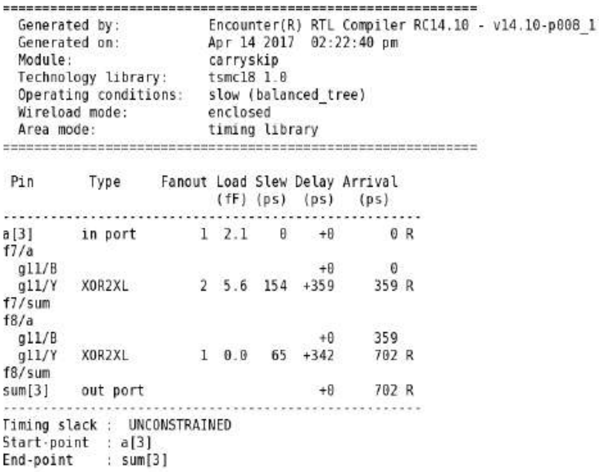
D. Using 180nm Technology:

Fast.lib

Slow.lib



```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  02:16:38 pm
  Module:                  carryskip
  Technology library:      tsmc18 1.0
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

 Pin        Type      Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
-----------------------------------------------------
a[0]       in port         1  2.2    0    +0       0 F
f1/a
  g11/B                                   +0       0
  g11/Y    XOR2XL          2  5.5   73  +158     158 R
f1/sum
g101/B                                    +0     158
g101/Y     NAND3XL         1  3.7   86   +41     199 F
g100/A1                                   +0     199
g100/Y     OAI31XL         1  0.0  108   +94     293 R
carry      out port                       +0     293 R
-----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[0]
End-point    : carry
```



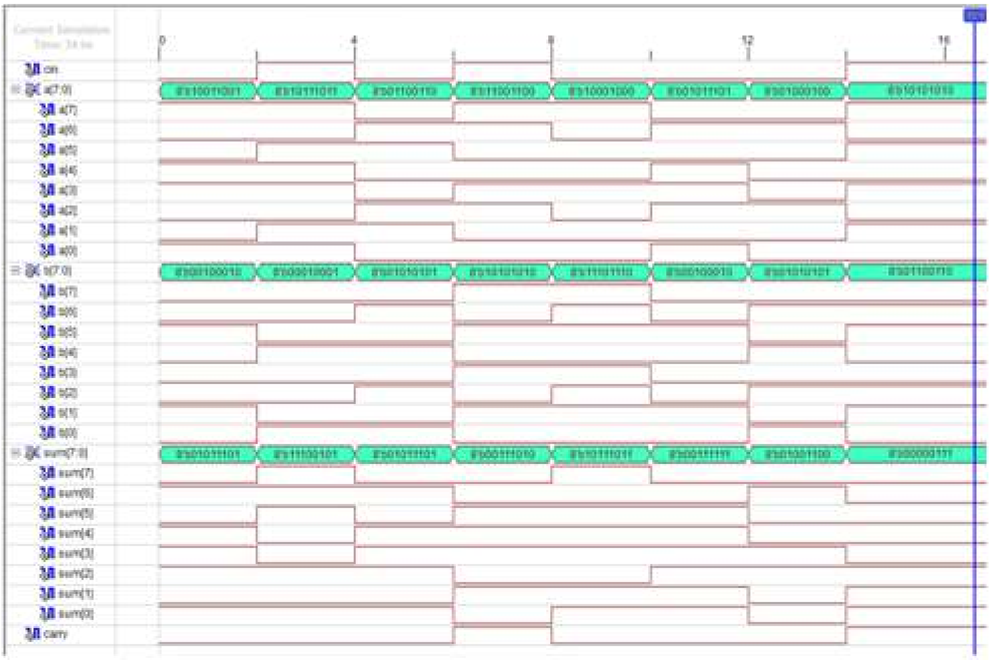
```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  02:22:40 pm
  Module:                  carryskip
  Technology library:      tsmc18 1.0
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

 Pin        Type      Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
-----------------------------------------------------
a[3]       in port         1  2.1    0    +0       0 R
f7/a
  g11/B                                   +0       0
  g11/Y    XOR2XL          2  5.6  154  +359     359 R
f7/sum
f8/a
  g11/B                                   +0     359
  g11/Y    XOR2XL          1  0.0   65  +342     702 R
f8/sum
sum[3]     out port                       +0     702 R
-----------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[3]
End-point    : sum[3]
```

Fig 69. Timing Synthesis Report for Carry skip adder using 180nm

6.2.3.1 CARRY SAVE ADDER:

It has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.

Fig 70. Output waveform of carry save adder



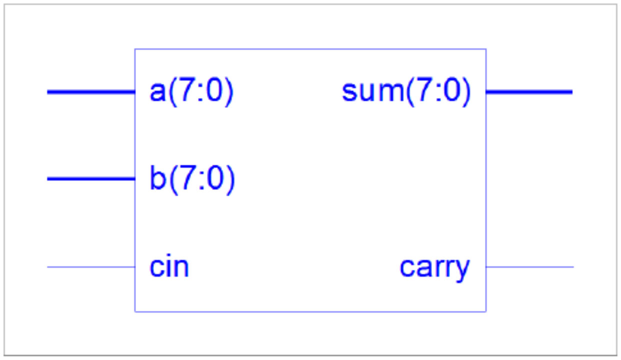


Fig 71. Carry save adder

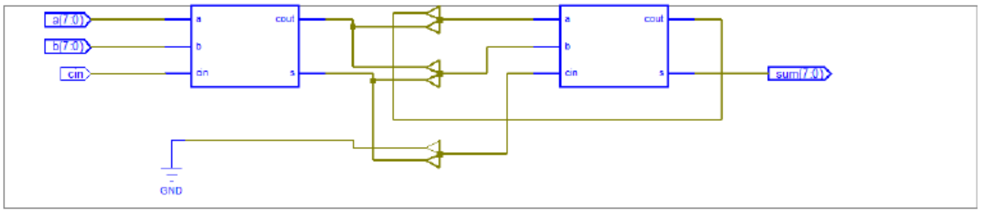


Fig 72. RTL view of carry save adder

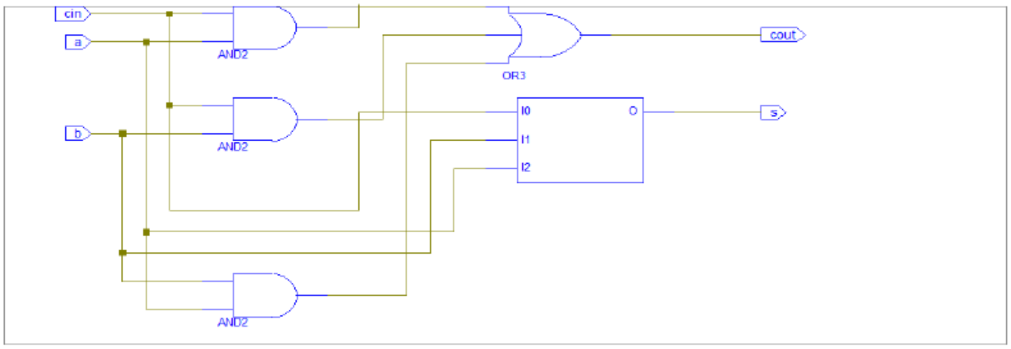


Fig 73. RTL view of full adder

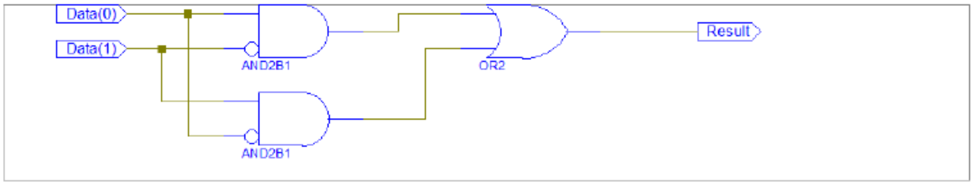
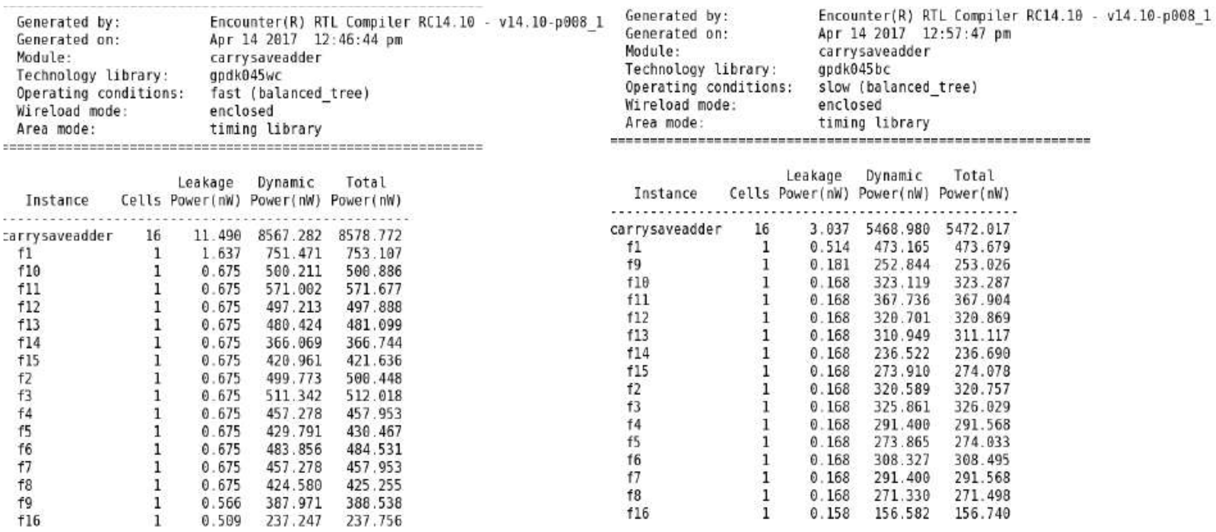


Fig 74. RTL view of XOR Gate

6.2.3.2 POWER SYNTHESIS RESULT:

For the synthesis, here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

B. Using 45nm Technology



Fast.lib

```
------------------------------------------------------------
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  12:46:44 pm
  Module:                 carrysaveadder
  Technology library:     gpdk045wc
  Operating conditions:   fast (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                        Leakage   Dynamic    Total
   Instance     Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------------
carrysaveadder     16    11.490  8567.282  8578.772
  f1                1     1.637   751.471   753.107
  f10               1     0.675   500.211   500.886
  f11               1     0.675   571.002   571.677
  f12               1     0.675   497.213   497.888
  f13               1     0.675   480.424   481.099
  f14               1     0.675   366.069   366.744
  f15               1     0.675   420.961   421.636
  f2                1     0.675   499.773   500.448
  f3                1     0.675   511.342   512.018
  f4                1     0.675   457.278   457.953
  f5                1     0.675   429.791   430.467
  f6                1     0.675   483.856   484.531
  f7                1     0.675   457.278   457.953
  f8                1     0.675   424.580   425.255
  f9                1     0.566   387.971   388.538
  f16               1     0.509   237.247   237.756
```

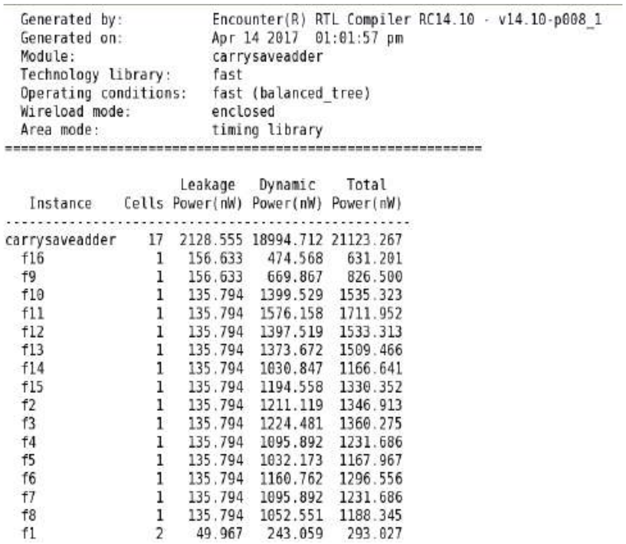
Slow.lib

```
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           Apr 14 2017  12:57:47 pm
  Module:                 carrysaveadder
  Technology library:     gpdk045bc
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                        Leakage   Dynamic    Total
   Instance     Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------------
carrysaveadder     16     3.037  5468.980  5472.017
  f1                1     0.514   473.165   473.679
  f9                1     0.181   252.844   253.026
  f10               1     0.168   323.119   323.287
  f11               1     0.168   367.736   367.904
  f12               1     0.168   320.701   320.869
  f13               1     0.168   310.949   311.117
  f14               1     0.168   236.522   236.690
  f15               1     0.168   273.910   274.078
  f2                1     0.168   320.589   320.757
  f3                1     0.168   325.861   326.029
  f4                1     0.168   291.400   291.568
  f5                1     0.168   273.865   274.033
  f6                1     0.168   308.327   308.495
  f7                1     0.168   291.400   291.568
  f8                1     0.168   271.330   271.498
  f16               1     0.158   156.582   156.740
```

Fig 75. Power synthesis report for Carry save adder using 45nm

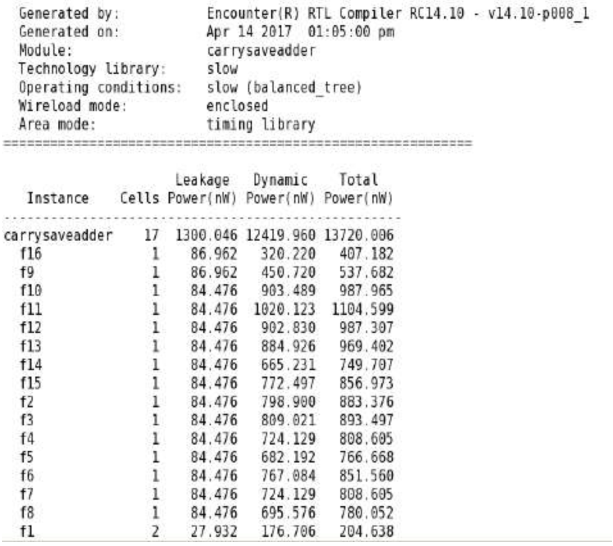
C. Using 90nm Technology

Fast.lib



```
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:01:57 pm
  Module:                  carrysaveadder
  Technology library:      fast
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                     Leakage   Dynamic    Total
   Instance    Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------------
carrysaveadder    17  2128.555 18994.712 21123.267
  f16              1   156.633   474.568   631.201
  f9               1   156.633   669.867   826.500
  f10              1   135.794  1399.529  1535.323
  f11              1   135.794  1576.158  1711.952
  f12              1   135.794  1397.519  1533.313
  f13              1   135.794  1373.672  1509.466
  f14              1   135.794  1030.847  1166.641
  f15              1   135.794  1194.558  1330.352
  f2               1   135.794  1211.119  1346.913
  f3               1   135.794  1224.481  1360.275
  f4               1   135.794  1095.892  1231.686
  f5               1   135.794  1032.173  1167.967
  f6               1   135.794  1160.762  1296.556
  f7               1   135.794  1095.892  1231.686
  f8               1   135.794  1052.551  1188.345
  f1               2    49.967   243.059   293.027
```

Slow.lib

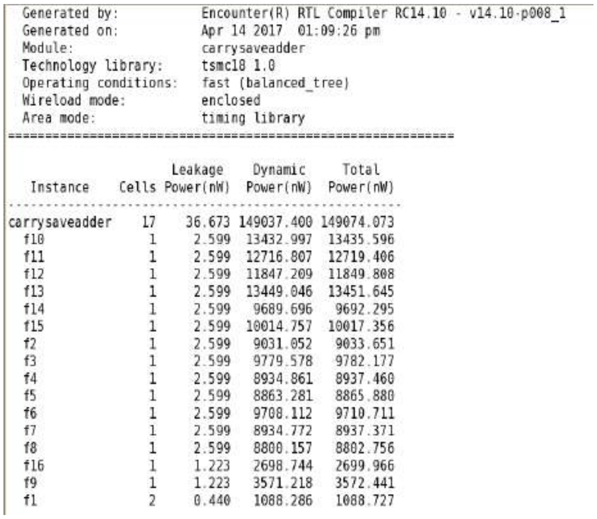


```
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:05:00 pm
  Module:                  carrysaveadder
  Technology library:      slow
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                     Leakage   Dynamic    Total
   Instance    Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------------
carrysaveadder    17  1300.046 12419.960 13720.006
  f16              1    86.962   320.220   407.182
  f9               1    86.962   450.720   537.682
  f10              1    84.476   903.489   987.965
  f11              1    84.476  1020.123  1104.599
  f12              1    84.476   902.830   987.307
  f13              1    84.476   884.926   969.402
  f14              1    84.476   665.231   749.707
  f15              1    84.476   772.497   856.973
  f2               1    84.476   798.900   883.376
  f3               1    84.476   809.021   893.497
  f4               1    84.476   724.129   808.605
  f5               1    84.476   682.192   766.668
  f6               1    84.476   767.084   851.560
  f7               1    84.476   724.129   808.605
  f8               1    84.476   695.576   780.052
  f1               2    27.932   176.706   204.638
```

Fig 76. Power synthesis report for Carry save adder using 90nm

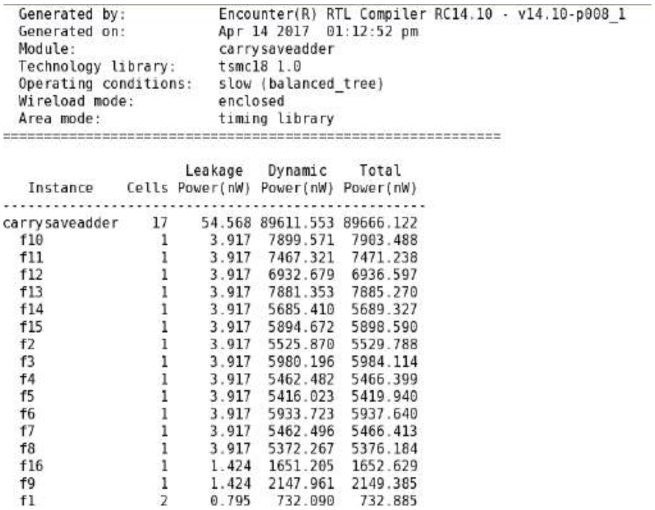
D. Using 180nm Technology:

Fast.lib:



```
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:09:26 pm
  Module:                  carrysaveadder
  Technology library:      tsmc18 1.0
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                     Leakage   Dynamic    Total
   Instance    Cells Power(nW) Power(nW) Power(nW)
-----------------------------------------------------
carrysaveadder    17    36.673 149037.400 149074.073
  f10              1     2.599  13432.997  13435.596
  f11              1     2.599  12716.807  12719.406
  f12              1     2.599  11847.209  11849.808
  f13              1     2.599  13449.046  13451.645
  f14              1     2.599   9689.696   9692.295
  f15              1     2.599  10014.757  10017.356
  f2               1     2.599   9031.052   9033.651
  f3               1     2.599   9779.578   9782.177
  f4               1     2.599   8934.861   8937.460
  f5               1     2.599   8863.281   8865.880
  f6               1     2.599   9708.112   9710.711
  f7               1     2.599   8934.772   8937.371
  f8               1     2.599   8800.157   8802.756
  f16              1     1.223   2698.744   2699.966
  f9               1     1.223   3571.218   3572.441
  f1               2     0.440   1088.286   1088.727
```

Slow.lib:



```
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:12:52 pm
  Module:                  carrysaveadder
  Technology library:      tsmc18 1.0
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

                     Leakage   Dynamic    Total
   Instance    Cells Power(nW) Power(nW) Power(nW)
---------------------------------------------------
carrysaveadder    17    54.568 89611.553 89666.122
  f10              1     3.917  7899.571  7903.488
  f11              1     3.917  7467.321  7471.238
  f12              1     3.917  6932.679  6936.597
  f13              1     3.917  7881.353  7885.270
  f14              1     3.917  5685.410  5689.327
  f15              1     3.917  5894.672  5898.590
  f2               1     3.917  5525.870  5529.788
  f3               1     3.917  5980.196  5984.114
  f4               1     3.917  5462.482  5466.399
  f5               1     3.917  5416.023  5419.940
  f6               1     3.917  5933.723  5937.640
  f7               1     3.917  5462.496  5466.413
  f8               1     3.917  5372.267  5376.184
  f16              1     1.424  1651.205  1652.629
  f9               1     1.424  2147.961  2149.385
  f1               2     0.795   732.090   732.885
```

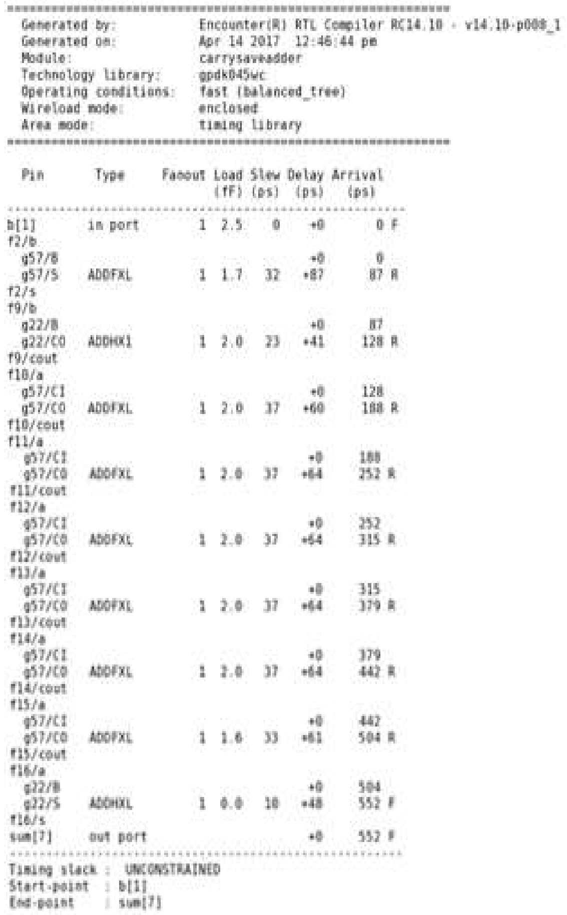
Fig 77. Power synthesis report for Carry save adder using 180nm

6.2.3.3 DELAY AND TIMING SYNTHESIS RESULTS:

Same as in case of power synthesis, here we are using here we are using fast.lib and slow.lib at supply voltage = 1.8 V for 180nm technology, supply voltage = 1.1 V for 90nm technology and supply voltage = 1.3 V for 45nm technology. As to these supply voltages the results for fast.lib and slow.lib are shown below:

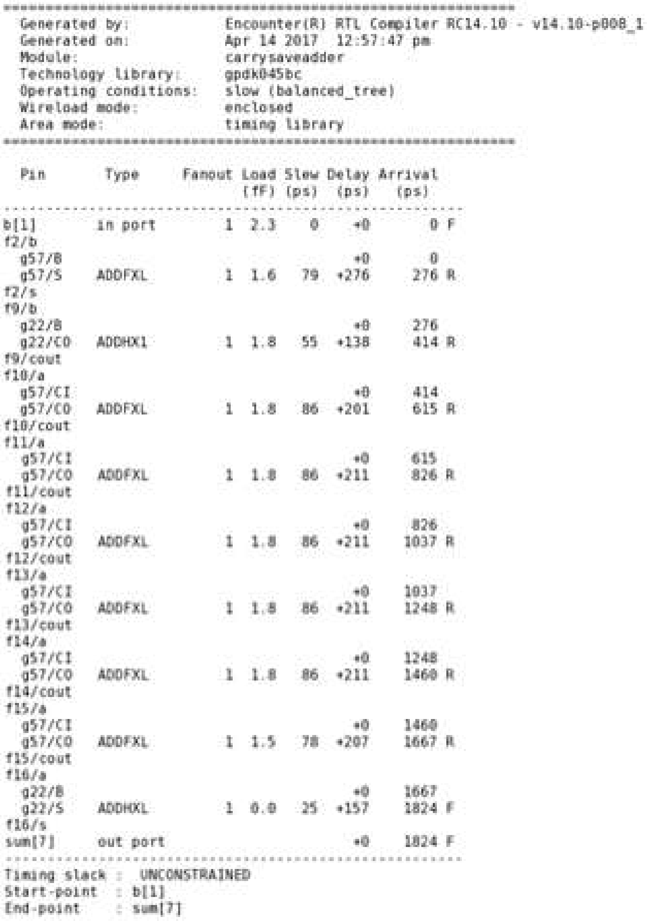
B. Using 45nm Technology

Fast.lib



```
================================================================
  Generated by:              Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:              Apr 14 2017  12:46:44 pm
  Module:                    carrysaveadder
  Technology library:        gpdk045wc
  Operating conditions:      fast (balanced_tree)
  Wireload mode:             enclosed
  Area mode:                 timing library
================================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
------------------------------------------------------
b[1]        in port        1  2.5    0    +0       0 F
f2/b
  g57/B                                   +0       0
  g57/S     ADDFXL         1  1.7   32   +87      87 R
f2/s
f9/b
  g22/B                                   +0      87
  g22/CO    ADDHX1         1  2.0   23   +41     128 R
f9/cout
f10/a
  g57/CI                                  +0     128
  g57/CO    ADDFXL         1  2.0   37   +60     188 R
f10/cout
f11/a
  g57/CI                                  +0     188
  g57/CO    ADDFXL         1  2.0   37   +64     252 R
f11/cout
f12/a
  g57/CI                                  +0     252
  g57/CO    ADDFXL         1  2.0   37   +64     315 R
f12/cout
f13/a
  g57/CI                                  +0     315
  g57/CO    ADDFXL         1  2.0   37   +64     379 R
f13/cout
f14/a
  g57/CI                                  +0     379
  g57/CO    ADDFXL         1  2.0   37   +64     442 R
f14/cout
f15/a
  g57/CI                                  +0     442
  g57/CO    ADDFXL         1  1.6   33   +61     504 R
f15/cout
f16/a
  g22/B                                   +0     504
  g22/S     ADDHXL         1  0.0   10   +48     552 F
f16/s
sum[7]      out port                      +0     552 F
------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[1]
End-point    : sum[7]
```

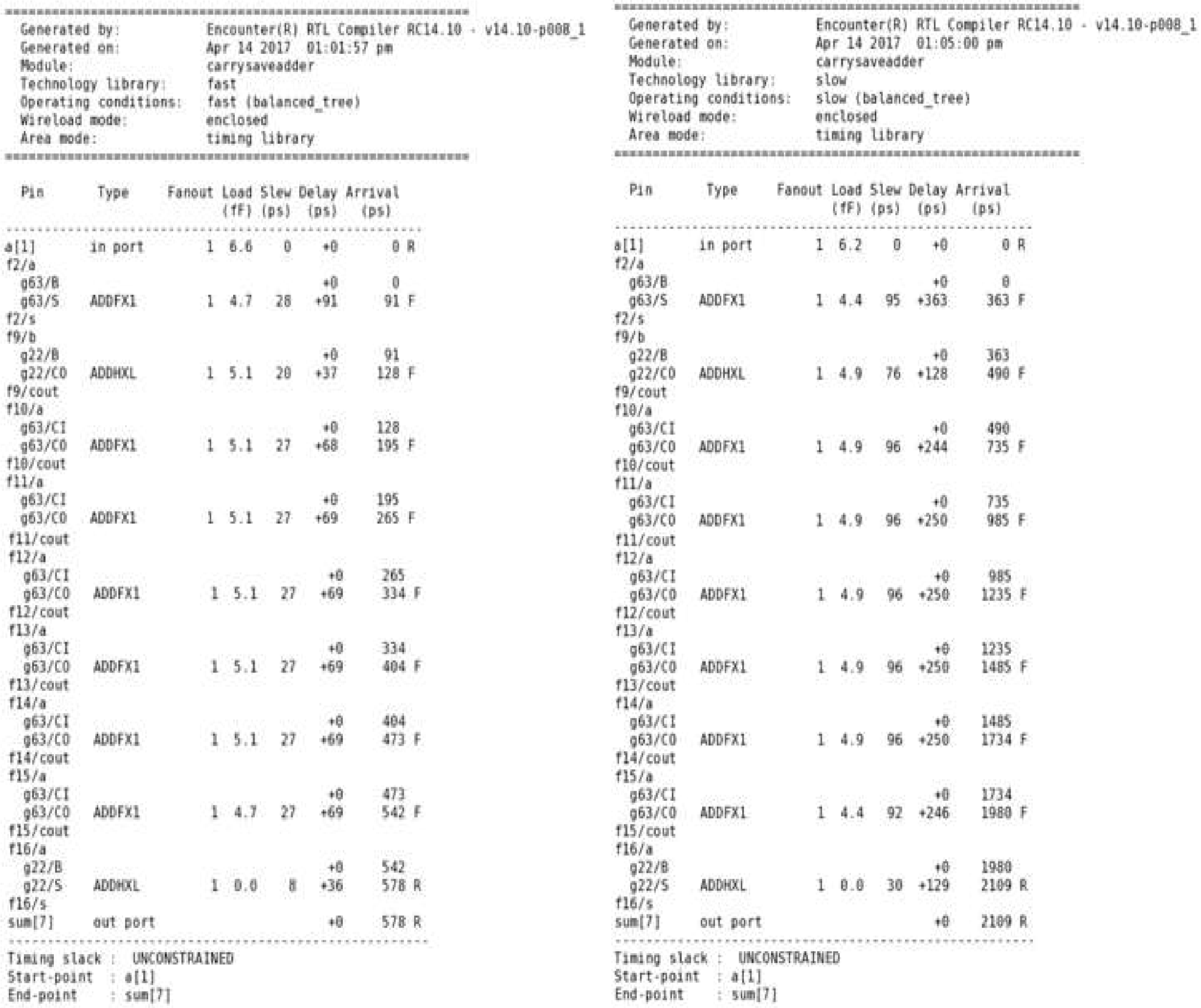
Slow.lib



```
================================================================
  Generated by:              Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:              Apr 14 2017  12:57:47 pm
  Module:                    carrysaveadder
  Technology library:        gpdk045bc
  Operating conditions:      slow (balanced_tree)
  Wireload mode:             enclosed
  Area mode:                 timing library
================================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
------------------------------------------------------
b[1]        in port        1  2.3    0    +0       0 F
f2/b
  g57/B                                   +0       0
  g57/S     ADDFXL         1  1.6   79  +276     276 R
f2/s
f9/b
  g22/B                                   +0     276
  g22/CO    ADDHX1         1  1.8   55  +138     414 R
f9/cout
f10/a
  g57/CI                                  +0     414
  g57/CO    ADDFXL         1  1.8   86  +201     615 R
f10/cout
f11/a
  g57/CI                                  +0     615
  g57/CO    ADDFXL         1  1.8   86  +211     826 R
f11/cout
f12/a
  g57/CI                                  +0     826
  g57/CO    ADDFXL         1  1.8   86  +211    1037 R
f12/cout
f13/a
  g57/CI                                  +0    1037
  g57/CO    ADDFXL         1  1.8   86  +211    1248 R
f13/cout
f14/a
  g57/CI                                  +0    1248
  g57/CO    ADDFXL         1  1.8   86  +211    1460 R
f14/cout
f15/a
  g57/CI                                  +0    1460
  g57/CO    ADDFXL         1  1.5   78  +207    1667 R
f15/cout
f16/a
  g22/B                                   +0    1667
  g22/S     ADDHXL         1  0.0   25  +157    1824 F
f16/s
sum[7]      out port                      +0    1824 F
------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b[1]
End-point    : sum[7]
```

Fig 78. Timing Synthesis Report for Carry save adder using 45nm

C. Using 90nm Technology



Fast.lib

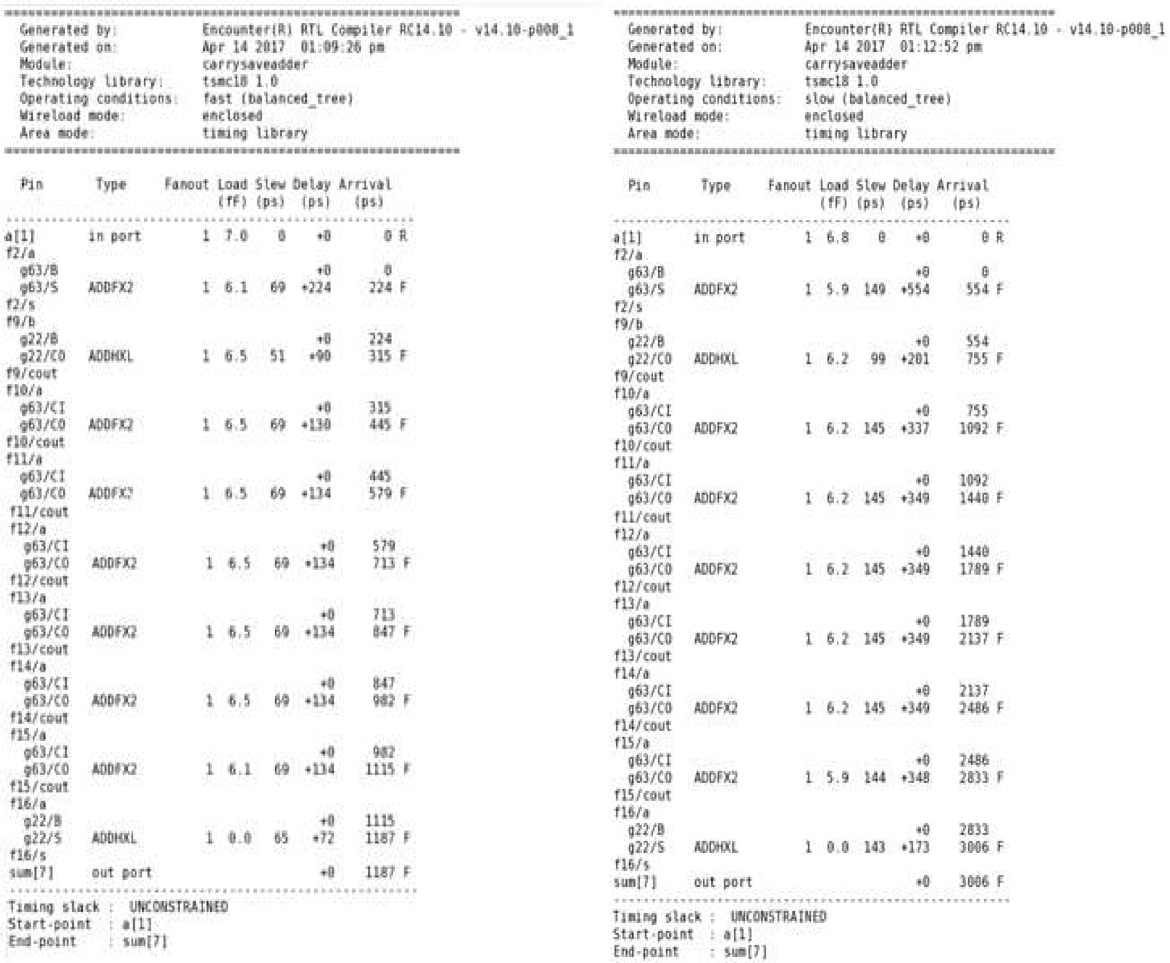
```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:01:57 pm
  Module:                  carrysaveadder
  Technology library:      fast
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin       Type      Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]       in port         1  6.6    0    +0       0 R
f2/a
  g63/B                                   +0       0
  g63/S    ADDFX1          1  4.7   28   +91      91 F
f2/s
f9/b
  g22/B                                   +0      91
  g22/CO   ADDHXL          1  5.1   20   +37     128 F
f9/cout
f10/a
  g63/CI                                  +0     128
  g63/CO   ADDFX1          1  5.1   27   +68     195 F
f10/cout
f11/a
  g63/CI                                  +0     195
  g63/CO   ADDFX1          1  5.1   27   +69     265 F
f11/cout
f12/a
  g63/CI                                  +0     265
  g63/CO   ADDFX1          1  5.1   27   +69     334 F
f12/cout
f13/a
  g63/CI                                  +0     334
  g63/CO   ADDFX1          1  5.1   27   +69     404 F
f13/cout
f14/a
  g63/CI                                  +0     404
  g63/CO   ADDFX1          1  5.1   27   +69     473 F
f14/cout
f15/a
  g63/CI                                  +0     473
  g63/CO   ADDFX1          1  4.7   27   +69     542 F
f15/cout
f16/a
  g22/B                                   +0     542
  g22/S    ADDHXL          1  0.0    8   +36     578 R
f16/s
sum[7]     out port                       +0     578 R
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : sum[7]
```

Slow.lib

```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:05:00 pm
  Module:                  carrysaveadder
  Technology library:      slow
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin       Type      Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]       in port         1  6.2    0    +0       0 R
f2/a
  g63/B                                   +0       0
  g63/S    ADDFX1          1  4.4   95  +363     363 F
f2/s
f9/b
  g22/B                                   +0     363
  g22/CO   ADDHXL          1  4.9   76  +128     490 F
f9/cout
f10/a
  g63/CI                                  +0     490
  g63/CO   ADDFX1          1  4.9   96  +244     735 F
f10/cout
f11/a
  g63/CI                                  +0     735
  g63/CO   ADDFX1          1  4.9   96  +250     985 F
f11/cout
f12/a
  g63/CI                                  +0     985
  g63/CO   ADDFX1          1  4.9   96  +250    1235 F
f12/cout
f13/a
  g63/CI                                  +0    1235
  g63/CO   ADDFX1          1  4.9   96  +250    1485 F
f13/cout
f14/a
  g63/CI                                  +0    1485
  g63/CO   ADDFX1          1  4.9   96  +250    1734 F
f14/cout
f15/a
  g63/CI                                  +0    1734
  g63/CO   ADDFX1          1  4.4   92  +246    1980 F
f15/cout
f16/a
  g22/B                                   +0    1980
  g22/S    ADDHXL          1  0.0   30  +129    2109 R
f16/s
sum[7]     out port                       +0    2109 R
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : sum[7]
```

Fig 79. Timing Synthesis Report for Carry save adder using 90nm

D. Using 180nm Technology:



Fast.lib

```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:09:26 pm
  Module:                  carrysaveadder
  Technology library:      tsmc18 1.0
  Operating conditions:    fast (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]        in port        1  7.0    0    +0       0 R
f2/a
  g63/B                                   +0       0
  g63/S     ADDFX2         1  6.1   69  +224     224 F
f2/s
f9/b
  g22/B                                   +0     224
  g22/CO    ADDHXL         1  6.5   51   +90     315 F
f9/cout
f10/a
  g63/CI                                  +0     315
  g63/CO    ADDFX2         1  6.5   69  +130     445 F
f10/cout
f11/a
  g63/CI                                  +0     445
  g63/CO    ADDFX2         1  6.5   69  +134     579 F
f11/cout
f12/a
  g63/CI                                  +0     579
  g63/CO    ADDFX2         1  6.5   69  +134     713 F
f12/cout
f13/a
  g63/CI                                  +0     713
  g63/CO    ADDFX2         1  6.5   69  +134     847 F
f13/cout
f14/a
  g63/CI                                  +0     847
  g63/CO    ADDFX2         1  6.5   69  +134     982 F
f14/cout
f15/a
  g63/CI                                  +0     982
  g63/CO    ADDFX2         1  6.1   69  +134    1115 F
f15/cout
f16/a
  g22/B                                   +0    1115
  g22/S     ADDHXL         1  0.0   65   +72    1187 F
f16/s
sum[7]      out port                      +0    1187 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : sum[7]
```

Slow.lib

```
============================================================
  Generated by:            Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:            Apr 14 2017  01:12:52 pm
  Module:                  carrysaveadder
  Technology library:      tsmc18 1.0
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
============================================================

  Pin        Type      Fanout Load Slew Delay Arrival
                              (fF) (ps)  (ps)   (ps)
-------------------------------------------------------
a[1]        in port        1  6.8    0    +0       0 R
f2/a
  g63/B                                   +0       0
  g63/S     ADDFX2         1  5.9  149  +554     554 F
f2/s
f9/b
  g22/B                                   +0     554
  g22/CO    ADDHXL         1  6.2   99  +201     755 F
f9/cout
f10/a
  g63/CI                                  +0     755
  g63/CO    ADDFX2         1  6.2  145  +337    1092 F
f10/cout
f11/a
  g63/CI                                  +0    1092
  g63/CO    ADDFX2         1  6.2  145  +349    1440 F
f11/cout
f12/a
  g63/CI                                  +0    1440
  g63/CO    ADDFX2         1  6.2  145  +349    1789 F
f12/cout
f13/a
  g63/CI                                  +0    1789
  g63/CO    ADDFX2         1  6.2  145  +349    2137 F
f13/cout
f14/a
  g63/CI                                  +0    2137
  g63/CO    ADDFX2         1  6.2  145  +349    2486 F
f14/cout
f15/a
  g63/CI                                  +0    2486
  g63/CO    ADDFX2         1  5.9  144  +348    2833 F
f15/cout
f16/a
  g22/B                                   +0    2833
  g22/S     ADDHXL         1  0.0  143  +173    3006 F
f16/s
sum[7]      out port                      +0    3006 F
-------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : a[1]
End-point    : sum[7]
```

Fig 80. Timing Synthesis Report for Carry save adder using 180nm

6.2.4 PERFORMANCE EVALUATION:

Table 4. Comparison between the parameters of Carry Select adder, Carry Skip adder, Carry Save adder.

Fast Adders	Technology used	Type	Cells	Total Power (uW)	Total Delay (ps)
Carry Select Adder	45nm	Fast	8	3781.515	548
		Slow	8	10786.93	569
	90nm	Fast	8	79143.01	1133
		Slow	8	2414.802	1743
	180nm	Fast	8	6942.391	2057
		Slow	8	47944.86	2876
Carry Skip Adder	45nm	Fast	21	8276.512	204
		Slow	21	10532.68	124
	90nm	Fast	21	41899.67	293
		Slow	20	3495.115	790
	180nm	Fast	21	7469.767	475
		Slow	21	26365.73	702
Carry Save Adder	45nm	Fast	16	8578.772	552
		Slow	17	21123.27	578
	90nm	Fast	17	149074.1	1187
		Slow	16	5472.017	1824
	180nm	Fast	17	13720.01	2109
		Slow	17	89666	3006

Comparison of Total No. of Cells:

Fast

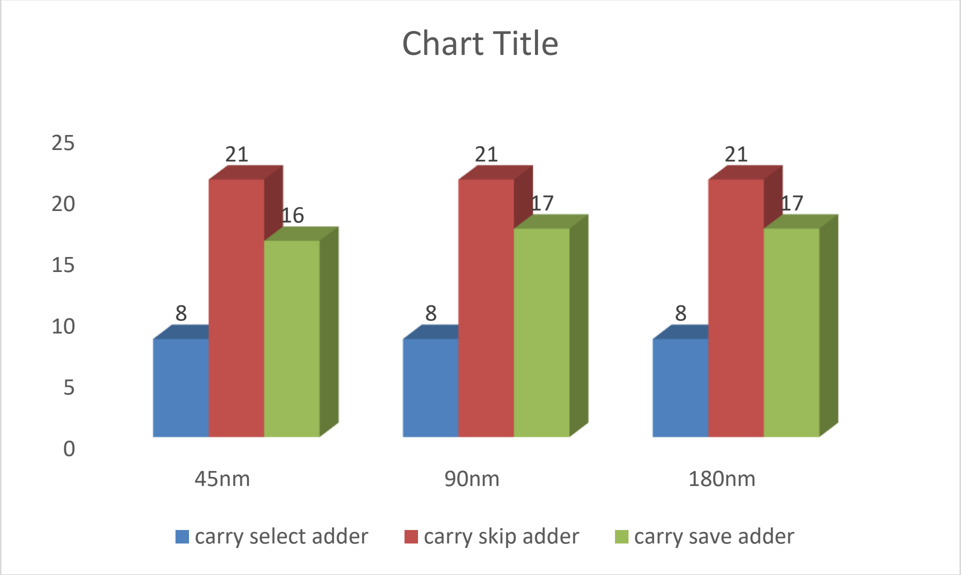


Fig 81. Comparison of Total No. of Cells Graph

Slow

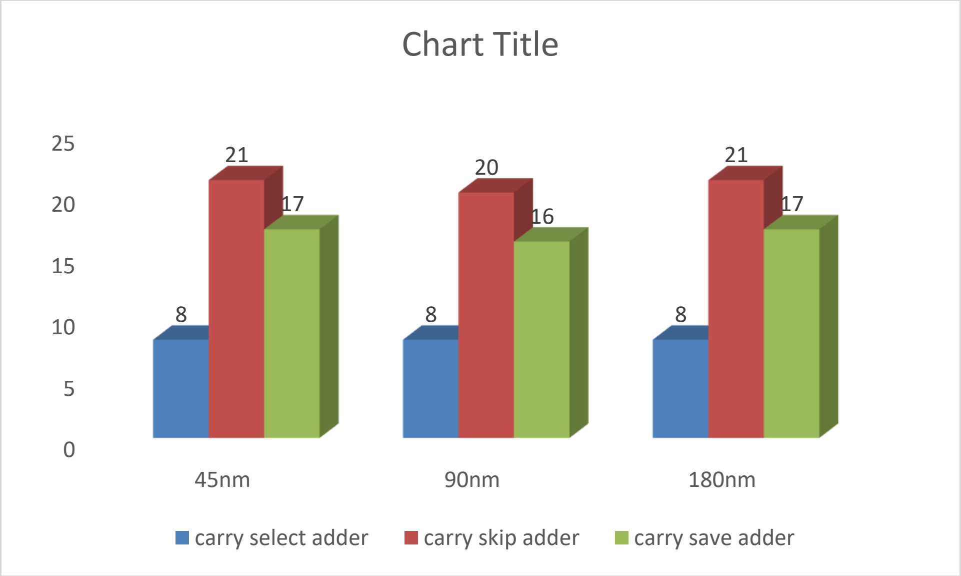


Fig 82. Comparison of Total No. of Cells Graph

Comparison of Total power consumption (in uW):

Fast

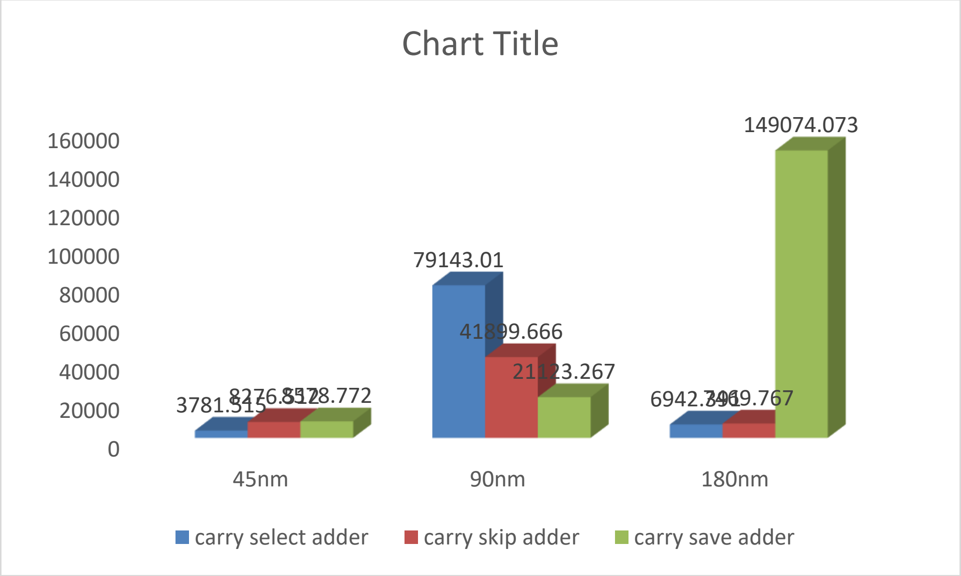


Fig 83. Comparison of Total Power Consumption Graph

Slow

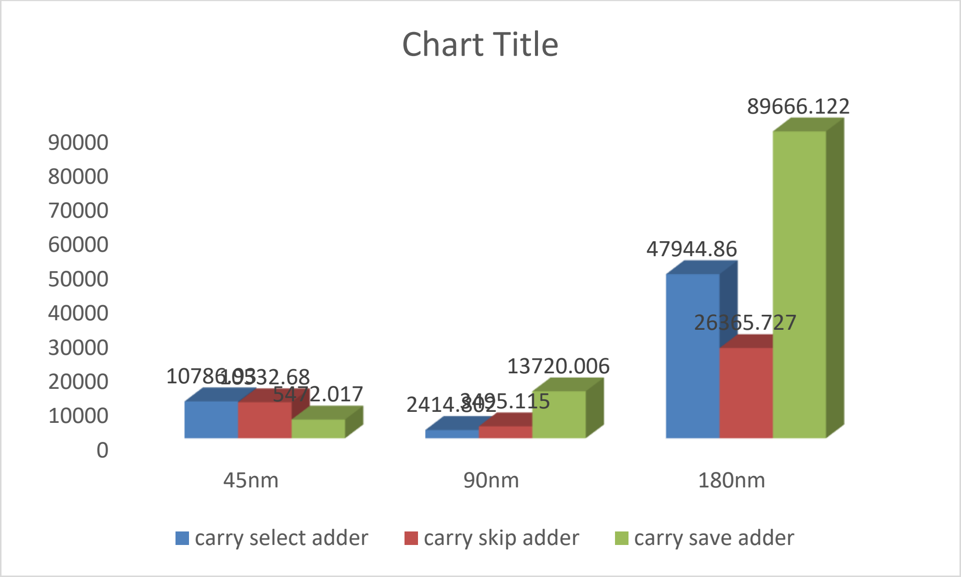


Fig 84. Comparison of Total Power Consumption Graph

Comparison of Total Delay (in ps):

Fast:

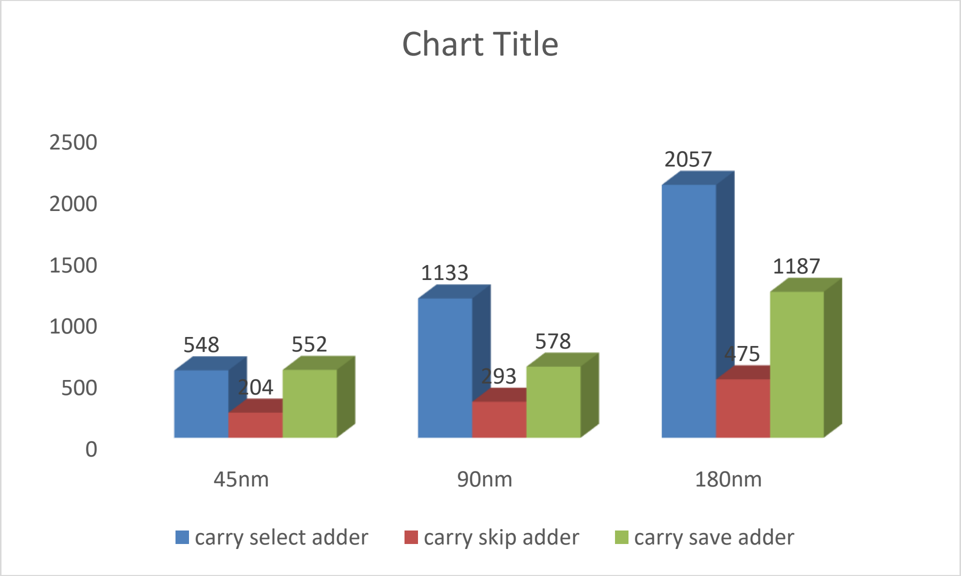


Fig 85. Comparison of Total Delay Graph

Slow:

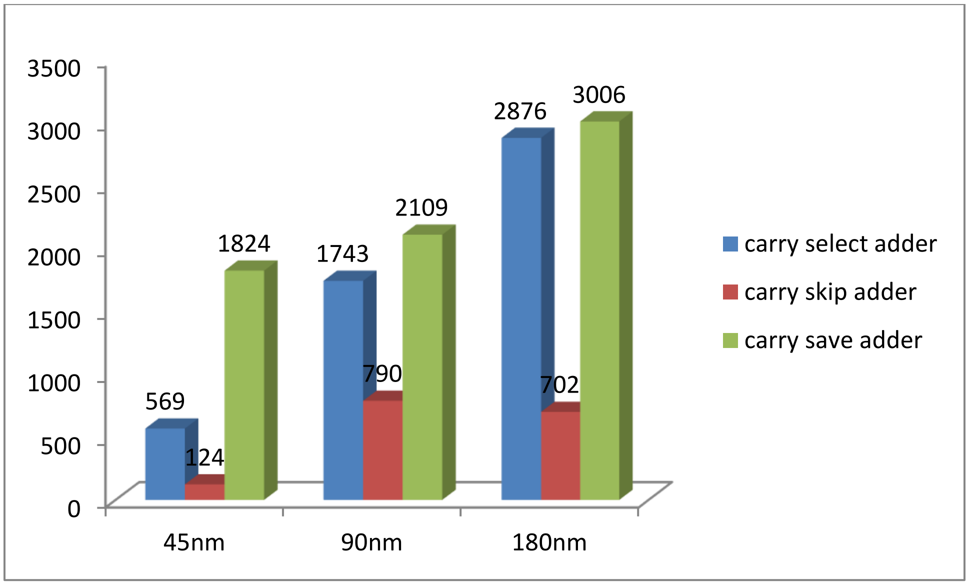


Fig 86. Comparison of Total Delay Graph

CHAPTER 7 CONCLUSION AND FUTURE SCOPE

Carry Select adder, Carry Skip Adder and Carry Save Adder each one of 4-bit and 8-bit has been implemented using Verilog in CADENCE. For simulation we have used Xilinx, NCsim and for synthesis we have used RTL compiler v14.10 have been used. As a result, it has been concluded that 4-bit and 8-bit Carry Select Adder gives the optimized result as compared to Carry Skip Adder and Carry Save Adder in all the technologies 45nm, 90nm, 180nm. When coming to the No. of cells Carry Select Adder uses less number of Cells when compared to Carry Skip and Carry Save Adder. While coming to total power consumption Carry Select Adder consumes less power compared to Carry Skip and Carry Save Adder. As per Delay parameter is concerned Carry skip adder is efficient manner in case of 4-bit and 8-bit Adders.

Depending upon the parametric analysis these fast adders have utilized in Multiplier Accumulator Unit (MAC) in DSP application.

In future, we compare with other Fast Adders and to will try to reduce the power consumption and delay by reducing the components and getting the optimal result.

CHAPTER 8 LIST OF REFERENCES

[1] Prashant Gurjar, Rashmi Solanki, Pooja Kansliwal, Mahendra Vucha, "VLSI Implementation of Adders for High Speed ALU" *International Journal of Computer Applications, September 2011.*

[2] Reena Rani, Laxmi Kanth Singh, Neelam Sharma, "FPGA Implementation of Fast Adders using Quaternary Signed Digit Number System" *2009 International Conference on Emerging Trends in Electronic and Photonic Devices & Systems (ELECTRO-2009)*

[3] Nuno Roma, Tiago Dias, Leonel Sousa, "Fast Adder Architectures: Modeling and Experimental Evaluation" *Portuguese Foundation for Science and for* Technology under the research project *Configurable and Optimized Processing Structures for Motion Estimation (COSME) POSI/CHS/40877/2001.*

[4] Jasbir Kaur, Lalith Sood, "Comparison Between Various Types of Adder Topologies" *InternatIonal Journal of Computer SCIenCe and technology IJCST Vol. 6, ISSue 1, Jan - MarCh 2015*

[5] Akash Kumar, Deepika Sharma, "Performance Analysis of Different Types of Adder for High Speed 32 Bit Multiply And Accumulate Unit" *International Journal of Engineering Research and Applications (IJERA) Vol. 3, Issue 4, Jul-Aug 2013, pp.1460-1462*

[6] Anand N, George Joseph, Johny S Raj, Jayakrishnan .P, "Implementation of Adder Structure with Fast Carry Network for High Speed Processor", *2013 InternationalConference on Green Computing, Communication and Conservation of Energy (ICGCE)*

[7] R. P. P Singh, Praveen Kumar, Balwinder Singh, "Performance Analysis Of Fast Adders Using VHDL" *2009 International Conference on Advances in Recent Technologies in Communication and Computing*

[8] Jucemar Monteiro, José Luís Güntzel, Luciano Agostini, "A1CSA: An Energy-Efficient Fast Adder Architecture for Cell-Based VLSI Design" *supported by the Brazilian Council for Scientific and Technological Development (CNPq).*

[9] Pavan Kumar.M.O.V, Kiran.M, "Design Of Optimal Fast Adder*" International Conference on Advanced Computing and Communication Systems (ICACCS-2013)*

[10] Ravikumar A Javali, Ramanath J Nayak, Ashish M Mhetar, Manjunath C Lakkannavar

"Design of high speed carry save adder using carry lookahead adder" *Proceedings of International Conference on Circuits, Communication, Control and Computing* (I4C 2014)

[11] Samiappa Sakthikumaran, S. Salivahanan, V. S. Kanchana Bhaaskaran, V. Kavinilavu "A Very Fast and Low Power Carry Select Adder Circuit" *Institute of Electrical and Electronics Engineers-2011*.

[12] Rajwinder Kaur, Amit Grover "Different Techniques used for Carry Select Adder – A Review" *International Journal of Computer Applications Volume 116 – No. 3, April 2015*

[13] Laxman Shanigarapu, Bhavana P. Shrivastava "Low-Power and High Speed Carry Select Adder" *International Journal of Scientific and Research Publications, Volume 3, Issue 8, August 2013.*

[14] Aamir A. Farooqui, **V**ojin G. Oklobdzija "General Data-Path Organization of MAC unit for VLSI Implementation of DSP Processors"

[15] A.Abdelgawad "Low Power Multiply Accumulate Unit (MAC) for Future Wireless Sensor Networks" *Institute of Electrical and Electronics Engineers-2013.*

[16] Yuyun Liao and David B. Roberts "A High-Performance and Low-Power 32-bit Multiply-Accumulate Unit with Single-Instruction-Multiple-Data (SIMD) Feature" *IEEE Journal Of Solid-State Circuits, Vol. 37, No. 7, July 2002*

[17] Ohsang Kwon, Kevin Nowka, Earl E. Swartzlander, Jr. "A 16-Bit MAC Design Using Fast 5:3 Compressor Cells" *Journal of VLSI Signal Processing 31, 77–89, 2002.*