CHAPTER 1

INTRODUCTION

1.1 MOTIVATION:

Today, it has achieved a great importance to be kept in mind two of the major parameters as power consumption and delay, while designing any electronic system, due to the reasons as:

Battery operated devices: Because of the presence of battery operated devices in a large number and the requirement of long or extended battery life, power consumption of the concerned electronic device should be very less. There are also some environmental constraints related to these devices. It is required to conserve the consumable energy in order to protect the environment. As there is a drastic increment in the power consumption in various electronic devices, so it has become critical to save environment by decreasing this increasing rate of power consumption.

Power consumption: According to the Moore's law, the number of transistors in small amount of area in semiconductor devices are increasing drastically, that's why higher power dissipates by that smaller amount of area and the design of thermal cooling unit and the packaging of the semiconductor devices is becoming more and more complex and costly too. Soo, low power electronic systems are required to handle this issue. For an electronic circuit, the dynamic power dissipation directly proportional to the load capacitance and activity factor. Generally, the load capacitance is high as the bus required to be linked with a number of modules and to be routed upto a longer distance, soo the activity factor also high. For the reason of the greater values of load capacitance and activity factor, the power consumption parameter can contribute upto 50% of the total value in a typical electronic device or system. Bus inverter encoding scheme is an effective technique which helps in reducing the amount of power consumption by decreasing the number of toggles i.e. the activity in the bus lines. In order to determine the kind of bus encoding scheme which suits best for a particular system, it is required to analyze its purpose and the environmental constraints respectively. That's why the scheme of bus encoding plays an important role in any electronic system.

1.2 INTRODUCTION:

With the development of technology in the field of VLSI engineering , it is becoming an important issue to keep in mind the two major parameters as-power consumption and delay , to achieve a better design everytime we use deep submicron technology. The power consumed by various buses in different VLSI circuits, relatively effects the performance and costs of the various devices. In order to achieve the power reduction to its maximized percentage, we are using a method ‘bus invert coding’ which is quite different from the conventional method as traditional schemes deals with the whole bus lines which somewhat effected the desired parameters as power reduction, propagation delay etc.

Inter wire coupling which is considered to be the main cause of power dissipation, delay in output generations and also crosstalk, is reduced upto a great extent utilizing ‘bus invert coding’ method in designs of circuits. For the mandatory desired performance of high speed chips, the details of various interconnects should be known which is the depending factor of performance of chip in deep submicron technology used for VLSI/ULSI designs.

The power sensitive devices like mobile phones, video games etc. require maximum reduction in power consumption in order to relatively increase its battery life or moreover the realiability of such devices. Such type of devices comes with microprocessors as their processing unit along with memories as their storage units. In present times the manufacturing of such devices is carried out using CMOS technology, in which most of the power consumed is in the form of dynamic power. The dynamic power can be examined by approximating the switching (toggling of bits) and switched capacitance.

Bus transition reduction is very much significant in order to control the reduction in power consumption. In our proposed work we have designed an 8 bit encoder utilizing ‘bus invert coding’ and ‘ pipelining’ methods for achieving more better results with reduced switching along with high speed circuit.

1.3 Power losses in digital circuits:

Generally, when we are discussing about the designing and manufacturing of various digital circuits or chips in micron and deep sub-micron technologies then the term “power dissipation” has a great significance. Power dissipation is the amount of power dissipated by the certain circuit during operation or in idle conditions. So, that we can classify the power dissipation in CMOS circuits into two categories namely- static power dissipation and dynamic power dissipation respectively.

1.3.1 Static power dissipation:

The leakage current in transistors is the main cause for static power dissipation. Static CMOS circuits are more efficient to power used in earlier technologies. There are also another factors too exist which are also somewhat responsibe for the static power dissipation as- subthreshold conduction due to OFF state of transistors in the circuit; due to the tunneling current of gate oxide; reverse biased diodes’ leakage; contention current existing in ratioed circuits.

1.3.2 Dynamic power dissipation:

There are two main factors which are responsible for the dynamic power dissipation. These are switching power and short circuit power. Switching power dissipation occurs when there is occurring a transition from 0 to 1 or from 1 to 0. The probability of such transitions is known as “switching activity”. We can express the dynamic power as:

$$P = \alpha(V2)CF$$

where,P = Dynamic power dissipation, α = Switching activity, V= Power supply

C= Load Capacitance, F= Clock Frequency.

The equation for power dissipation in a bus is as follows:

$$P = \alpha(V2)CFN$$

where N = Width of the bus.

Inferring from the above equation, the power dissipation in a bus proportional to the switching activity and the width of the bus. To reduce power dissipation on a bus the input the input data and the data on the bus is reduced.

1.4 Bus Invert Coding:

Bus coding can be defined as a method of converting one data stream into another form of data stream in order to be launched on the data bus more efficiently for better results . Moreover bus coding served efficiently for various desiring factors as reduction in number

of pins, compression of the transmitting data stream, cross talk reduction in data bit stream etc. Bus invert coding is proposed in order to reduce dynamic power consumed by the buses in the system, it also reduces the switching or bus transition which in turn proves to be the effecting factor of dynamic power. The purpose of bus coding is to decrease the hamming distance between the two consecutive data bus values. As the whole above process is proportionally dependent upon the hamming distance, it is determined to be an effective way to reduce the whole process factor, which in turn greatly reduces the power consumption (Dynamic Power).

In todays era of high speed, more reliable, great battery life, less weight, along with efficient features, it is quite significant to improve the power consumption by the devices using various developed and developing techniques which are increasingly reducing its percentage and giving more and more better performance.

1.4.1 Advantages of using bus invert coding:

a) Bus coding technique greatly helps in reducing the dynamic power consumption.
b) It also reduces the propagation delay along with increase in fast system response which in turn increase system speed.
c) Selection of correct bus coding techniques will surely results in better performance of the system or VLSI/ULSI chips.
d) It is an easy and efficient way to meet the desired parameters in various systems along with betterment in its properties.

1.5 Switching:

Switching is basically meant for the number of bus transitions per time slot in a circuitory. It is the switching of data bits from 0 state to 1 state or from 1 state to 0 state respectively. As we have earlier explained that the switching or number of bus transitions is directly proportional to the amount of power dissipated by the concerned circuitory. Greater the value of switching, higher will be the power consumption or vice-versa. So, if we want to design a circuit which consumes less power, then the switching value should be controlled using some method or technique.

1.6 Majority Voter:

It is a circuit which is precisely manufactured in order to produce a selective signal depending upon the first input data value and its inverted one. It is quite similar in working of a comparator as in this circuit there is a comparison held between a data input value and its inverted value for the purpose of generating a selection signal.

1.7 Pipelining:

Pipelining is one of the specifically proven technique for improving the performance of digital systems respectively. Pipelining is used in combinational logic in order to improve the system throughput. In pipelining concept, the larger combinational blocks are further partitioned into two smaller blocks or more than two blocks independent from each other during operation. These blocks or combinational logics are connected to registers. Each of the splitted blocks have approximately reduced delay than the original delay of the system. As each and every combinational block is having its own register and thus, each part works independently and have no impact on each other's performance, while operating upon the two different values simultaneously. The cycle time get reduced due to the decrement in the maximum delay of the system's combinational logic. We have used this concept in our proposed bus encoder scheme.

The concept of branching a broad bus into several smaller sub-buses, which keeps all these sub-buses independent from each other and all the sub-buses are able to work in parallel, is known as 'pipelining'. Extra invert lines are added to the circuit which adds up to an extra area, it is an observed disadvantage of pipelining concept. It is also having an advantage of reducing the delay produced by the majority voter, which in turn effects the power consumption for which we are very much concerned in our work. We have used 'bus invert coding' concept along with 'pipelining' concept in our proposed work for better results.

1.8 SYSTEM OVERVIEW

- **Coding Language : Verilog or VHDL**
- **Tools:- : Xilinx Software 9.2ise**

Xilinx tool will be used to design proposed architecture. It supports both VHDL and Verilog language. ISE simulator would be used to simulate the design for functional verification. I

will use Verilog language. It is easy to learn, most syntaxes are similar to C language, Verification environment can be made using Verilog means no need of other languages, Case sensitive, almost tools support this language. Explored Xilinx tool and designed basic gates, Combinational logic (Mux, Encoder, Decoder, Multipliers etc) in Verilog.

1.9 ORGANIZATION OF THESIS

This whole report is abinding of various sections describing different concepts in chapters likewise. Thus, each and every section describes some specific points about the concerned topics as:

Chapter 1- It includes introduction which convey the goal of this designed system and overview of designed system.

Chapter 2- It presents literature review and the related work done in bus bus invert coding along with the problem statement and the methodology used in designing the proposed works.

Chapter 3- It presents the flow of proposed work, schemes related to it and circuit diagram.

Chapter 4- The experimental results which has been studied and simulated and also simulation reports of the proposed and previous works;.

Chapter 5 - It outlines the future scope and draws the conclusion about the proposed work.

Chapter 6 - It outlines the references taken for our proposed work, respectively.

CHAPTER-2
LITERATURE SURVEY

2.1 LITERATURE REVIEW

This chapter represents an overview of background research to the project. The literature also gives the brief idea about the technical, operational and economical feasibility study.

A pipelined 8B10B encoder for a high speed SerDes was being proposed by Song Yu-yun and Hu Qing-Sheng. They presented a reduced power consumption and high speed model named as 'pipelined 8B10B encoder'. Their presented encoder performs better than the previous designs and noticeable results were found in respect of area, power, delay respectively. In order to avoid the limitation of speed parameter of the system as in traditional schemes, a pipelined bus encoder design was proposed by them. The delay of the critical paths was reduced upto a great extent by distributing longer paths into several smaller data paths. Using the pipelining concept, a high speed 8B10B encoder design is proposed using 0.18 micrometer CMOS technology and standard cell library. From the post simulation results, it was obtained that the encoder can efficiently work upto a rate of 7Gbps with a core area of 76.86 mocrometer x 76.86 micrometer. The power consumption along with a 1.8 V power supply voltage is S.0317 mV.
Advantages:

1. they used pipelined architecture so delay is less.
2. For an 8-bit data bus, it has reduced power consumption.

Disadvantages:

1. This algorithm is not efficient for more than 8 bits.
2. Propagation delay is more so it is too slow.

Junkai sun and Anping Jiang has given the concept of a new modified design that improves the bus invert coding method in order to reduce the power consumption in data bus which is the main parameter responsible for performance of the system necessarily. As according to the conventional method used previously, whole bus lines were utilized for further application of bus invert coding method, respectively, and this scheme is somewhat different from the conventional one as in this scheme several larger buses are partitioned into various smaller sub-buses which then further utilized for bus invert coding method application independently which definitely improves the system performance. The bus invert coding method can reduce at least 25% of the power consumption for an 8-bit data bus, quite better than the traditional method which gives the reduction of 18.75% respectively.

Advantages:

1. Modified the bus invert coding to maximize power consumption reduction.
2. Each partitioned sub-bus has been coded independently.
3. For an 8-bit data bus, it has reduced upto 25% of the power consumption.

Disadvantages:

1. This proposed algorithm is more efficient only for general purpose systems, not efficient to use in complex systems.
2. Propagation delay is more so it is too slow.

S.R. Malathi and R. Ramya Asmi, introduced a technique for power consumption optimization of compressed code systems, by complementing the tramsmitted bits on the data bus respectively. Usually, compression raises the toggling of bits, as the descutancies in the tramsmitted bits of data bus get removed. The compression or decompression and toggling

reduction is accompanied by arithmetic coding technique which is done by shift invert coding method. It is also a challenging task to maintain a correct balance between the compression ratio and toggling reduction. Two extra bits are needed for this technique for power optimized coding , apart from the data bus width for compression. This technique is proposed from the concepts of shift invert coding method and Arithmetic coding method, in order to reduce no. of bus transitions and thus, reduces the power consumption in various embedded systems respectively. The memory access get reduced using arithmetic coding. It also gives an increased compression ratio as maximum data in the file is unique. During data access, this unique data is responsible for most of the toggling. To reduce this toggling of bits, shift invert coding method is used. It is proved from various experiments conducted that shift invert coding is better than bus invert coding method.

Advantages:

1. reduce no. of bus transitions .
2. For an 8-bit data bus, it has reduced power consumption.

Disadvantages:

1. This proposed algorithm is more efficient only for general purpose systems, not efficient to use in complex systems.
2. Algorithm is too complex so Area is increased.

Due to unavailability of such applications, we could not do a comparison of results for the work done in the same field.

Kumari Khushboo and Ghanshyam Jangid has given the concept of a new design in order for betterment and modification of bus invert coding method , quite different from the traditional one to improve the two most important performance parameters as speed of the system and power consumption optimization respectively using reversible logic. A satisfactory improvement in system's area and speed were being found from their work. A high performance bus encoder

technique was proposed by them using a new gate namely “KK Gate”. Noticeable changes in speed, area, garbage o/p, constant i/p and gate count were found from their work.

Advantages:

1. Modified the bus invert coding to maximize power consumption reduction.
2. improve the speed of the system.

Disadvantages:

1. due to reversible logic area is increased.

Jayapreetha Natesan and Damu Radhakirshnan presented a new technique of coding as “shift invert coding” which is superior to the previous method of bus invert coding. A noticeable decrement in number of bus transitions from the simulation results which was definitely better from the simulation results results of bus invert coding. Regardless of data bus width, this technique requires only 2 extra bits for reduced power operation and no assumption is required related to the nature of the data in the data bus. This new technique was being proposed using right shift and left shift operations respectively. It is an easy and convenient scheme that enhances the bus invert coding method. When a random data sequence is taken then, it is found from the simulation results that the average no. of transitions were reduced greatly using shift invert coding method which is better than the bus invert coding method results. The presented technique does not have a great area overhead and also no assumption is taken for data bus values. Also this shift invert coding better compatibles with the data buses having arbitrary widths.

Advantages:

1. Modified the bus invert coding to reduce switching reduction.
2. The algorithm can be easily modified for more bits(16 bits , 32 bits)

Disadvantages:

3. This proposed algorithm is more efficient only for general purpose systems, not efficient to use in complex system
4. Propagation delay is more so it is too slow.

CHAPTER-3

PROPOSED WORK

As I have earlier about the reduction in number of bus transitions that is correlated with switching greatly reduces the amount of power dissipated that is the dynamic power in modern VLSI circuits designed using various submicron and deep submicron technologies. One of the way to decrease these number of bus transitions and the switching is “bus invert coding ”. This technique greatly helps in reducing the switching which in turn decreases the amount of power dissipation. But the results obtained after using this technique named “bus invert coding” can be still modulated with the help of “proposed work” and “pipelining technique”, which gives more better results as switching activities and delay are reduced upto a great extent.

In our proposed work, we have proposed a new bus invert coding method along with “pipelining technique” in order to get more better results from the past works, with minimum delay, minimum power dissipation along with greater, speed of the circuit. We have used “Xilinx 9.2i” for our proposed work as the tool. Firstly, we will explain “bus invert encoding” then, “ proposed bus invert encoding” after that finally, the “pipelining technique” in expanded form using the majority voter circuit along with my proposed bus encoder circuit.

3.1 Bus Invert Coding:

Bus-Invert method is used to, coding and decoding the I/O which lowers the bus activity and thus reduces the input-output peak power dissipation, by 50% and also the I/O average power dissipation by up to 25% . This happens because buses most have very large capacitances associated with them and consequently dissipate a lot of power. . In general, we get that, using the above technique namely ‘bus invert coding’ the power consumption is decreased upto 18.75%.

Bus invert coding can be explained by considering its activity upon some data bus stream. Let us consider an uniformly distributed random sequence of data bits. Assuming an N-bit broad data bus for any given time-slot of data bits can have any of 2N possible values with equal probability. An average idea of number of bus transitions per slot of time will be N/2 and the possibility of transition of each data bus line will be 1/2.

Bus invert coding method is having an algorithm which can be represented as follows:

a) Determination of hamming distances between existing and next data values.
b) If the hamming distance's value is greater than N/2 then invert is set to 1 and the value of next bus is made equal to the inverted value of next data.
c) If the hamming distance's value is smaller than N/2 then invert is set to 0 and the value of next bus is made equal to the next data value.
d) Elsif the hamming distance's value is equal to N/2 then invert is kept untouched or same as before and the next bus value is made equal to the next data value when inv.= 0 or inverted next data value when inv.=1.
e) The contents of the data bus on the receiving end are conditionally inverted or kept unchanged according to the invert line.

Mathematical Representation:

$(Bus^t, Invert^t) = \{(bus^t,0) , Hamm.^t < N/2\}$

$\{(inv.\ bus^t,1) , Hamm.^t > N/2\}$

$\{(inv.\ bus^t,1) , Hamm.^t = N/2$

$(bus^t,0) , Hamm.^t = N/2\}$

Where Bus^t stands for bus value at time t; $Invert^t$ stands for invert line value at time t; bus^t stands for data value at time t; $Hamm.^t$ stands for hamming distance between data value and bus value at time t.

Properties Of Bus Invert Coding Function:

There must be a bijection in bus encoding or decoding function which necessarily wants the encoding function to have below behavior.

a) There should be a unique encoded value for every data stream to be launched on data bus, and each of the encoded value should be particularly decoded for the similar original value.
b) It should be made quite possible to decode and encode each and every data value originated from the source side.

3.1.1 Transition active reduction of an 8-bit data bus:

Let us take an 8-bit data bus, for example: suppose the hamming distance distribution to be a binomial distribution observed between the next data value and the present data value, so that we see that the maximum probability observed will be for the hamming distance valuing 4. And according to the bus invert algorithm given above it has been observed that for 4, no gain is there

present in the contents of data bus during invert. That's why coding will be ineffective for this value i.e 4. Now using the bus invert code, the maximum hamming distance between the next data value and the present one will be observed as 4, which is the exact half of the uncoded data bus. Thus, the dynamic power dissipation will be subjected to a decrement of 50% when the technique of "bus invert coding" is used in the circuit.

The number of the coded data bus transitions per time-slot can be expected in advance and the expected value can be calculated using the following formula:

$E(n) = \sum^{N}_{i=0} i.\ H(i)$……..eq.(1)

Or $E(n) = N.\ p_{invert} + \sum^{N/2}_{i=0} i.\ H(i)$…….eq.(2)

Here H(i) represents the probability of hamming distance i. Equation (1) is for the uncoded data bus while equation (2) is for the data bus which has been modulated using "bus invert coding" respectively.

The average decrement in the switching activity of an 8-bit data bus will be from 4 to 3.27 in case when the data bus is encoded using 'bus invert coding' method respectively.

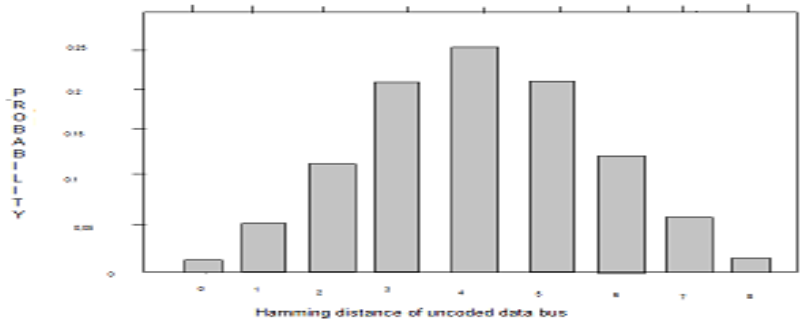


Fig 3.1: Hamming distance's binomial distribution of next data value and the present one (N = 8).

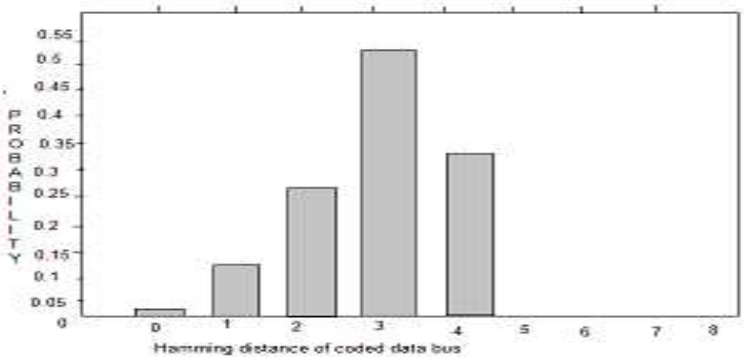


Fig3.2: Probability distribution of Hamming distances of the upcoming data values for an 8-bit data bus

3.1.2 Implementation of bus invert coding technique:

When we want to implement the 'bus invert coding' method, we require certain extra circuitory on the data path that means certain increase in area usage which sometimes becomes the reason for inferior performance of the circuit. In case, when this method is to be implemented for an 8-bit data bus, we require: 16 XOR gates; a majority voter circuitory; the bus drivers along with the latches respectively at the transmitting side for the purpose of encoding of the 8-bit data bus.

Whereas on the reception side, decoding is accommodated using 8 XOR gates along with the bus receivers. In order to transfer the inverted value to the receiver's side, an extra bus is to be added in the circuit, along with the above requirements.

The figure represents the circuit to apply the 'bus invert coding' method in manner to an 8-bit data bus possibly. This possible scheme represents the respective arrangement of the above defined required circuits and components at the receiver as well as at the transmitting side of the circuit. The XOR gates are meant for the comparison of data bits on the 8-bit data bus while the majority voter circuit is implemented in the bus encoder circuit below, for the purpose of generating a selective signal on the basis of input data value and its inverted value. Thus, the majority voter circuit also does the work of comparing the two values that is the data value which is being inputted from the transmitter side and its inverted value and generates certain selective signal in accordance with both values respectively.

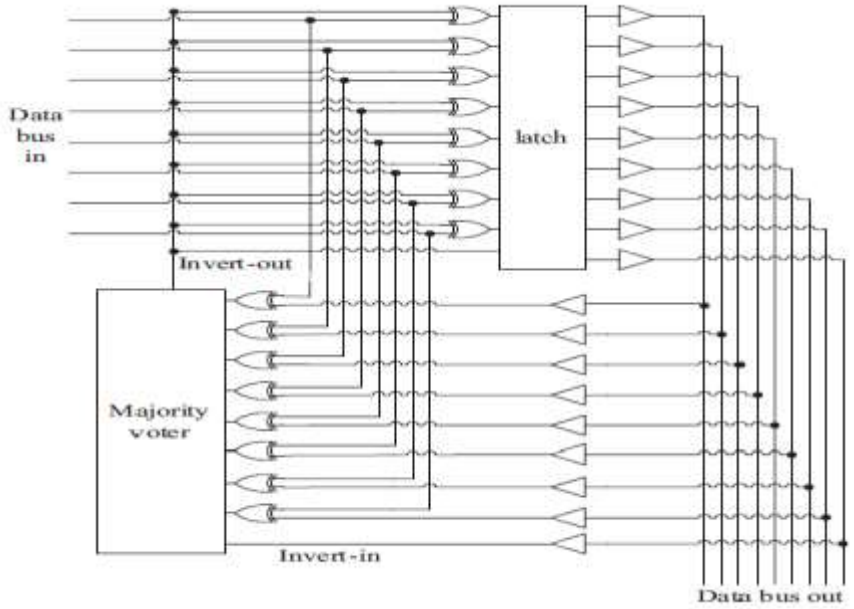


Figure 3.3: Bus Invert Encoder

Majority voter is a circuit that is precisely designed for the generation of selective signals based upon the first data input value and its inverted value. Thus, it generates a selective signal in accordance with the comparison of two values i.e. first data input value and its inverted value.

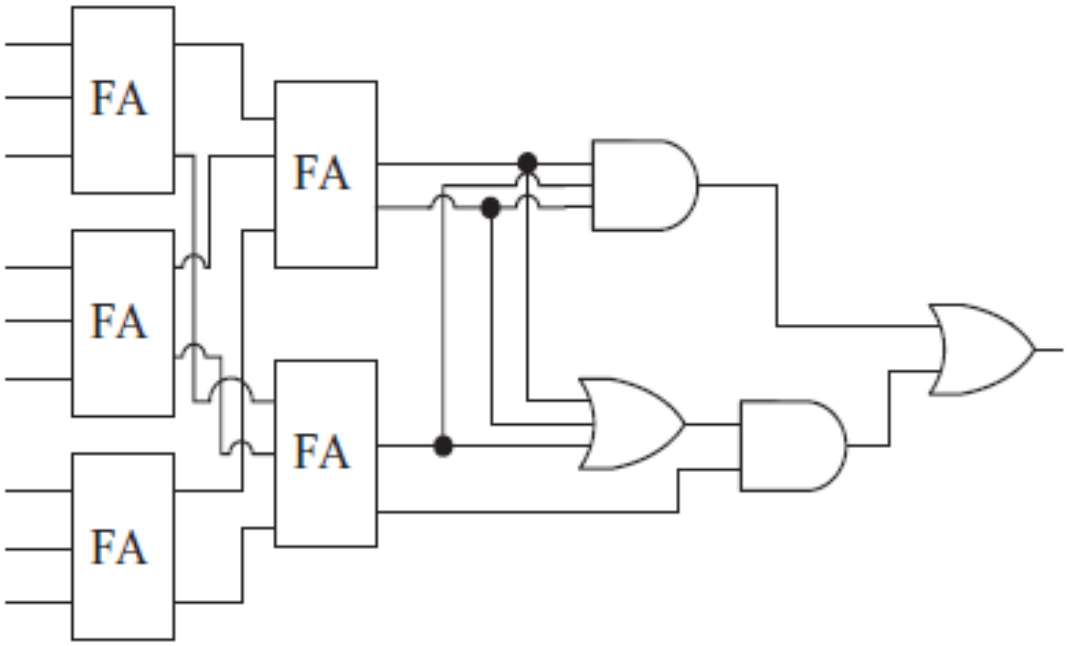


Fig 3.4: Majority voter.

The majority voter circuit can be implemented with a tree of full adders.

We can observe the power consumption through the three parts as follows:

$P_{data\ bus} = ½\ C_{data\ bus}\ V^2\ f\ .\ Num_{data\ bus}$ ……..(eq. 1)

$P_{invert} = ½\ C_{invert}\ V^2\ f\ .Num_{invert}$ …….(eq. 2)

$P_{int} = ½\ C_{int}\ V^2\ f\ .\ [Num_{coding} + Num_{decoding}]$ …….(eq. 3)

Equations shows that the power consumption of any digital circuits are depended on source voltage , capacitance and frequency (the change in data per second or switching). source voltage , capacitance totally depend on technology . if we reduce it other issues are involved. But we can reduce frequency (the change in data per second or switching) by bus encodings.

3.1.3 Limitations of bus invert encoder:

a) the chances of maximum switching activities is five. The chances of it is five.

b) the chances of switching activities is four. The chances of it is four.

c) the propagation delay is higher.

d) A supplement circuitory being added up in the design as there are encoder and decoder circuits to be added around the bus. This will consume some specific amount of dynamic power

e) The leakage power will also increase due to this additional circuitory. In case when the base activity factor is not quite high, bus encoding method not proved to be an appropriate option as there will be higher leakage power which in turn degrades the whole system's energy consumption.

f) In case when the bus timing exists in the critical data path , addition of this circuitory in the path will definitely degrade the timing path and use of bus encoding method will proved to be inappropriate one.

3.1.4 Proposed bus encoder:

We have seen that some limitation like more switching is there in convention bus encoder. To overcome this limitation we proposed a new bus encoder which is more efficient than convention bus encoder.

The architecture of proposed bus encoder is shown in fig.

The proposed bus encoder architecture and algorithm are little bit same to bus invert encoder. The comparator and working of majority voter are same but architecture is different.

3.1.4.1 Architecture:

The proposed bus encoder is consisting of a comparator; two majority voters along with a multiplexer. The data input is comprising of 8-bits and its obvious to obtain 8-bit data as output too. A comparison is held between each bit of Din and Dout respectively. So, the comparator will generate the certain sequence of compared bits of inputs and outputs, which then forwarded to the two majority voters. These majority voters will generate selective control signals based upon the data sequences which were inserted into them. finally the whole will be multiplexed using multiplexer and at the end the output is taken and also it will be inserted again at the transmitting end for further comparison.

The compatibility of bus coding method with small N data bus, is the reason for their better performance with small data buses. The larger data buses can be divided into several smaller sub-buses which can be coded individually and particularly. Assuming a random uniform distribution of data bits over a large bus then also the data on the sub-buses are randomly distributed too and these sub-buses are not dependent on each other more likely.

3.1.4.2 Algorithm:

Bus invert coding can be explained by considering its activity upon some data bus stream. Let us consider an uniformly distributed random sequence of data bits. Assuming an 8-bit broad data bus for any given time-slot of data bits can have any of 16 possible values with equal probability. An average idea of number of bus transitions per slot of time will be 8/2 =4 and the possibility of transition of each data bus line will be 1/2.

Proposed Bus invert coding method is designed on following algorithm which can be represented as follows:

a) Determination of hamming distances between existing first four data bus value D_out[3:0] and next first four data values Din[3:0].

b) Determination of hamming distances between existing last four data bus value D_out[7:4] and next last four data values Din[7:4].

c) If the hamming distance's value for Din[3:0]. is greater than two then invert(Cnt[0]) is set to 1 and the value of next bus D_out[3:0] is made equal to the inverted value of next data Din[3:0].

d) If the hamming distance's value for Din[7:4]. is greater than two then invert(Cnt[1]) is set to 1 and the value of next bus D_out[7:4] is made equal to the inverted value of next data Din[7:4].

e) If the hamming distance's value is smaller than two then invert(Cnt[0]) is set to 0 and the value of next bus D_out[3:0].is made equal to the next data value Din[3:0].

f) If the hamming distance's value is smaller than two then invert(Cnt[1]) is set to 0 and the value of next bus D_out[7:4] is made equal to the next data value Din[7:4].

g) Elsif the hamming distance's value is equal to two then invert is kept untouched or same as before and the next bus value D_out[3:0] is made equal to the next data value Din[3:0].when inv.= 0 or inverted next data Din[3:0]. value when inv.=1.

h) Elsif the hamming distance's value is equal to two then invert is kept untouched or same as before and the next bus value D_out[7:4] is made equal to the next data value Din[7:4].when inv.= 0 or inverted next data Din[7:4]. value when inv.=1.

i) The contents of the data bus on the receiving end are conditionally inverted or kept unchanged according to the invert line.

Mathematical Representation:

$(Bus^t\ [7:4], Bus^t\ [3:0], Invert^t[1:0]) = \{(\ bus^t\ [7:4], bus^t\ [3:0],\ 00)\}$ if

$\{\ Hamm.^t2 < N/2\ ,\ Hamm.^t\ 1< N/2\ \}$

$=\{(\ bus^t\ [7:4], inv.bus^t\ [3:0],\ 01)\}$ if

$\{Hamm.^t2 < N/2\ ,\ Hamm.^t\ 1>N/2\ \}$

$=\{(inv.\ bus^t\ [7:4], bus^t\ [3:0],\ 10)\}$ if

$\{\ Hamm.^t2 > N/2\ ,\ Hamm.^t\ 1< N/2\ \}$

$=\{\ (inv.bus^t\ [7:4], inv.bus^t\ [3:0],\ 11)\}\ ,$

$Hamm.^t2 > N/2\ ,\ Hamm.^t\ 1> N/2\ \}$

Where $Bus^t\ [7:4], Bus^t\ [3:0]$, stands for four MSB bus value and four LSB bus value at time t; $Invert^t$ stands for invert line value at time t; $bus^t\ [7:4], bus^t\ [3:0]$, stands for four MSB data value and four LSB data value at time t; $Hamm.^t1$ stands for hamming distance between data value $bus^t\ [3:0]$, and bus value $Bus^t\ [3:0]$, at time t. $Hamm.^t2$ stands for hamming distance between data value $bus^t\ [7:4]$, and bus value $Bus^t\ [7:4]$, at time t.

3.2 Switching:

Switching is basically meant for the number of bus transitions per time slot in a circuitory. It is the switching of data bits from 0 state to 1 state or from 1 state to 0 state respectively. As we have earlier explained that the switching or number of bus transitions is directly proportional to the amount of power dissipated by the concerned circuitory. Greater the value of switching, higher will be the power consumption or vice-versa. So, if we want to design a circuit which consumes less power, then the switching value should be controlled using some method or technique. We have used ‘bus invert’ method in our proposed work to reduce switching value or the number of bus transitions. The main focus has been given on the reduction of power dissipated which becomes possible by controlling the switching in the circuit.

Let us consider the following 8-bit data bus:

Data bus- D7 D6 D5 D4 D3 D2 D1 D0
Present data- b7 b6 b5 b4 b3 b2 b1 b0

The data values contained in (D0-D7) is "10101110" and in (b0-b7) is "01011111". So, that a comparison will be held between each of the input value and the current data bus value using comparator circuit i.e the XOR block as represented in the bus encoder circuit. XORing of the (D7-D0) will be done with (b7-b0) respectively which can be represented as follows:

(D0) XOR (b0) gives o/p as '1' ;
(D1) XOR (b1) gives o/p as '1' ;
(D2) XOR (b2) gives o/p as '1' ;
(D3) XOR (b3) gives o/p as '1' ;
(D4) XOR (b4) gives o/p as '0' ;
(D5) XOR (b5) gives o/p as '0' ;
(D6) XOR (b6) gives o/p as '0' ;
(D7) XOR (b7) gives o/p as '1' ;

Finally the 'XOR' operation is performed for all the 8 bits of data bus values and present data bus values. The number of ones represents the value of switching, i.e.
(No. of ones = value of switching).

So, that using "bus invert encoding" method, the number of bus transitions or we can say that switching is '5' for the above example.

Now, if we use 'bus invert encoding' then the following conclusion will be obtained:

Data bus- "10101110"
Present data- "01011111"
Let (C7-C0) denotes comparator output

(D) XOR (B) Comparator's output

C7	C6	C5	C4	C3	C2	C1	C0

If we consider the C7-C4 , C3-C0 separate block as proposed work . than probability(case) of no. of switching are shown in tables

Case 1: When no. of switching is '4' ,the probability are '5' as five conditions are possible as follows:

(C7-C4)	(C3-C0)	Switching
1111	0000	4
0000	1111	4
0001	0111	4
0111	0001	4
1010	1010	4

Case 2: When switching is '3', then probability will be '4' as four conditions are possible as follows:

(C7-C4)	(C3-C0)	Switching
1111	0001	3
0001	1111	3
1010	1011	3
1011	1010	3

Thus, it has been obtained that the application of 'bus invert encoding' algorithm will reduce switiching activity in most of the cases. But still there are cases in which, the above method is not efficient. For such cases we will apply the concept of 'partitioning' of the comparator into two parts.

After that we are sending the comparators output to the two majority voters which counts the no. of switching in each blocks.

3.3 Majority Voter:

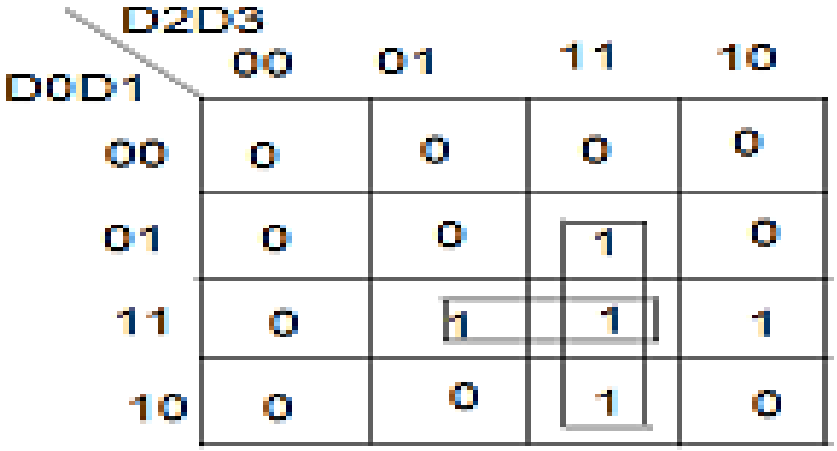
Majority voter is a circuit that is precisely designed for the generation of selective signals based upon the first data input value and its inverted value. Thus, it generates a selective signal in accordance with the comparison of two values i.e. data input value and bus value.

We are using two majority voter to counts the no. of switching in each blocks.

Property of proposed majority voter:

The output(Cnt) of majority voter will be 1 if the no. of one is more than two in input signals . otherwise output (Cnt)will be 0.

The majority voter circuit is implemented with using this Property and K-map**.**



Cnt = D2 D3 D1 + D2 D3 D0 + D0 D1 D2 + D0 D1 D3

Cnt = D2 D3 (D1+D0) + D0 D1 (D2+D3)

The above expressions showing the operation of majority voter. The architecture of majority voter is shown in following fig.

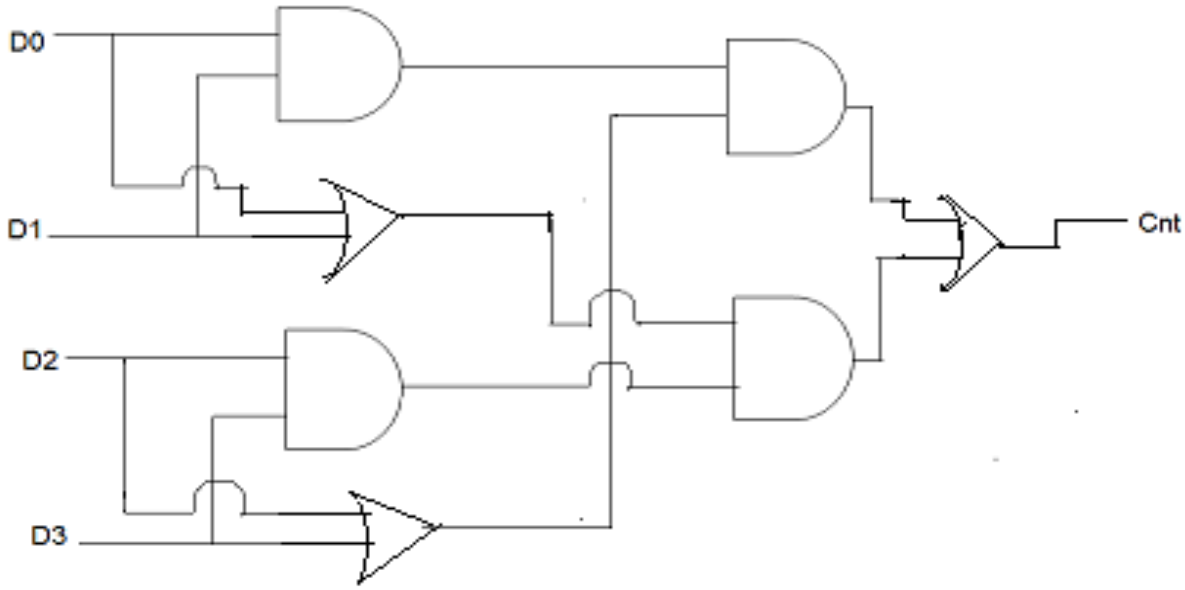


Fig 3.5: proposed majority voter

Switching reduction:

If we take same example and compare with proposed work than we will see that if we use ' proposed bus invert encoding' then the following conclusion will be obtained:

Data bus- "10101110"

Present data- "01011111"

The data values contained in (D0-D7) is "10101110" and in (b0-b7) is "01011111". So, that a comparison will be held between each of the input value and the current data bus value using comparator circuit i.e the XOR block as represented in the bus encoder circuit. XORing of the (D7-D0) will be done with (b7-b0) respectively which can be represented as follows:

(D0) XOR (b0) gives o/p as '1' ;

(D1) XOR (b1) gives o/p as '1' ;

(D2) XOR (b2) gives o/p as '1' ;

(D3) XOR (b3) gives o/p as '1' ;

(D4) XOR (b4) gives o/p as '0' ;

(D5) XOR (b5) gives o/p as '0' ;

(D6) XOR (b6) gives o/p as '0' ;

(D7) XOR (b7) gives o/p as '1' ;

And the ‘ proposed bus invert encoding algorithm says that if the number of bus transitions is more than 2 in each block then, the present data will get inverted. So, that the present data will be now changed to “10100000” , now if we compare between the data bus and present data then it is clearly acceptable that the switching or number of bus transitions will get reduced to ‘1’ from ‘3’ as bus invert coding method.

Let (C7-C0) denotes comparator output

(D) XOR (B) Comparator’s output

C7	C6	C5	C4	C3	C2	C1	C0

If we consider the C7-C4 , C3-C0 separate block as proposed work . if we apply proposed algorithm:

Case 1: When no. of switching is ‘4’ ,the probability are ‘5’ as five conditions are possible according to the conventional bus encoder are follows is reduced in proposed work :

(C7-C4)	(C3-C0)	Switching in conventional bus encoder	Switching in proposed bus encoder
1111	0000	4	0
0000	1111	4	0
0001	0111	4	2
0111	0001	4	2
1010	1010	4	4

Switching is ‘4’ and probability is ‘5’, switching get reduced by ‘1’.

So, the average power consumption will be 80%.

Case 2: When no. of switching is '3' ,the probability are '4' as four conditions are possible according to the conventional bus encoder are follows is reduced in proposed work :

(C7-C4)	(C3-C0)	Switching in conventional bus encoder	Switching in proposed bus encoder
1111	0001	3	1
0001	1111	3	1
1010	1011	3	3
1011	1010	3	3

Switching is '3' and probability is '4' , switching get reduced by '1'.
So, the average power consumption will be now 50%.

3.1.4 Proposed pipelined bus encoder:

Introduction:

We have implemented a new bus encoder which has less switching activities than convention bus encoder. It is more efficient in the term of power losses but still we have issue with propagation delay . so we further improved the architecture of Proposed bus encoder. We are using pipelined concept to minimize the propagation delay. Due to the presence of above previously defined drawbacks of the 'bus encoder', we have proposed a new bus encoder using 'xilinx 9.2i', which overcome the above limitations to a quite level and gives more better results.

3.4 Pipelining:

Pipelining is one of the specifically proven technique for improving the performance of digital systems respectively. Pipelining is used in combinational logic in order to improve the system throughput. In pipelining concept, the larger combinational blocks are further partitioned into two smaller blocks or more than two blocks independent from each other during operation. These blocks or combinational logics are connected to registers. Each of the splitted blocks have approximately reduced delay than the original delay of the system. As each and every combinational block is having its own register and thus, each part works independently and have

no impact on each other's performance, while operating upon the two different values simultaneously. The cycle time get reduced due to the decrement in the maximum delay of the system's combinational logic. We have used this concept in our proposed bus encoder scheme. The concept of branching a broad bus into several smaller sub-buses, which keeps all these sub-buses independent from each other and all the sub-buses are able to work in parallel, is known as 'pipelining'. Extra invert lines are added to the circuit which adds up to an extra area, it is an observed disadvantage of pipelining concept. It is also having an advantage of reducing the delay produced by the majority voter, which in turn effects the power consumption for which we are very much concerned in our work. We have used 'proposed bus invert coding' concept along with 'pipelining' concept in our proposed work for better results.

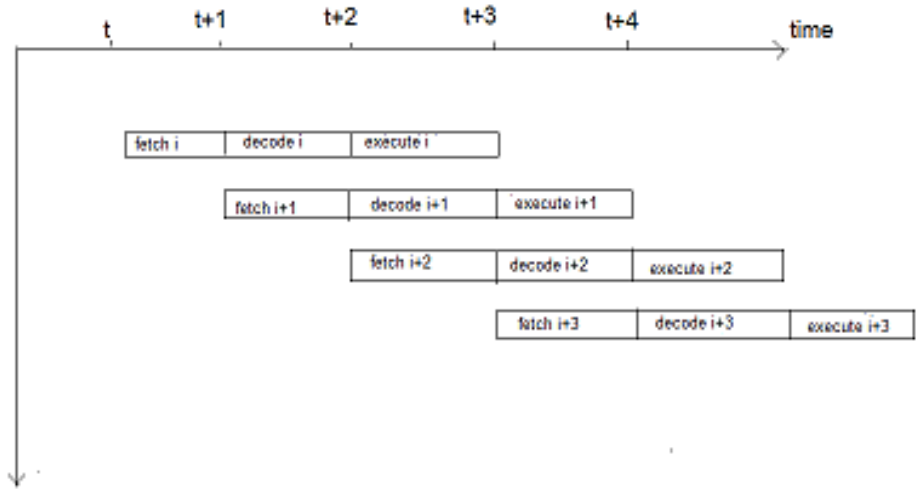


Fig3.6: timing diagram for three stage pipelining concept.

As we can clearly see in the above image that, each operation will remain independent from each other, along with the performance of more than one operation at once.

The concept of branching a broad system into several smaller system , which keeps all these sub-system independent from each other and all the system are able to work in parallel, is known as 'pipelining'. In pipelining concept, the strategy is so maintained that the dividation of larger data buses do not effect the operations in each of the resultant data bus parts. Thus, each and every part of the larger data bus or we can say the sub-buses remain isolated from each other.

pipelining concept speeds up the execution of various tasks ongoing within the circuit as more than one operation can be performed simultaneously. It is the proven effective way to indulge concurrent activity within the system.

Problems In Pipelining:

a) Structural problems may generate due to conflicts in hardware resource. These are the problems which can be resulted in case when hardware is unavailable to service all parallel combinations of pipeline stages.
b) Data problems may occur in case when one command is related with the results of another command that has still not generated the desired results.
c) There exist some commands or branches of pipeline which can change the flow of commands in sequence, these cases results in control problems.
d) Latency may increase due to 'high' pipelining that is the propagating time for some signal will be increase.

Possible Solutions To Reduce Problems In Pipelining:

a) Structural problems can be controlled by stalling, refactoring the pipeline, resource distribution to reduce memory pressure.
b) Data problems can be avoided by stalling, data bypassing and their combination for complex conflicts.

Control problems can be avoided by stalling, branching delay and prediction of the next command

3.5 Proposed pipelined bus encoder:

Architecture :

The architecture of proposed bus encoder is shown in figure 3.7.

The proposed bus encoder architecture and algorithm are little bit same to bus invert encoder .

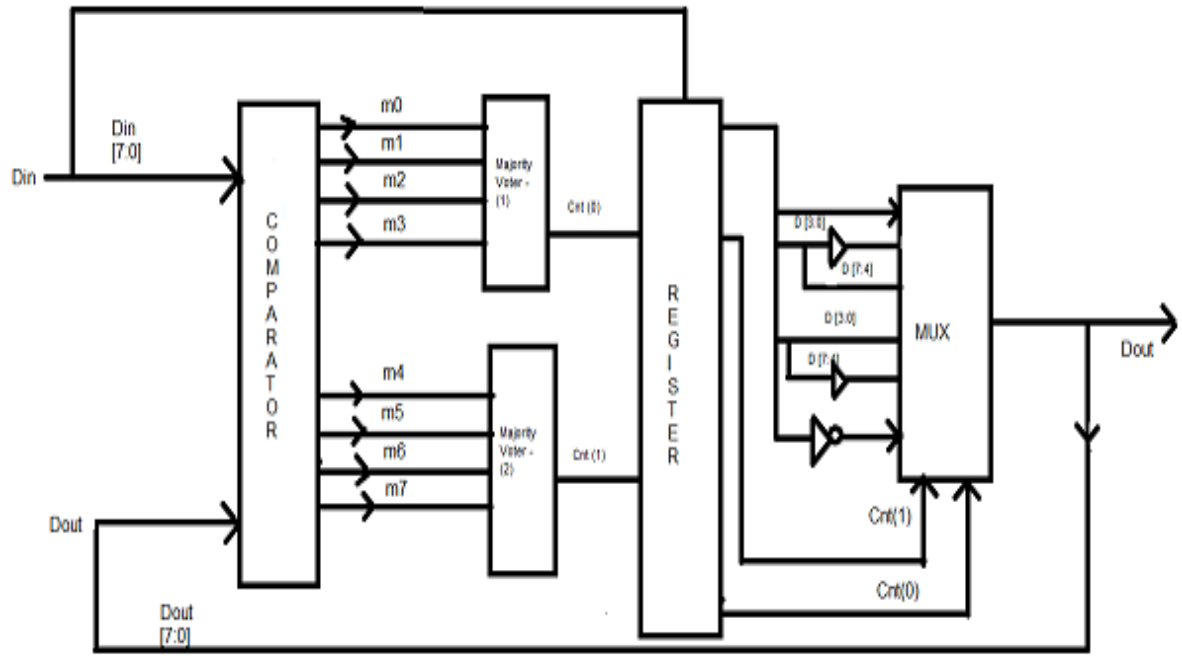


Fig 3.7: Proposed pipelined bus encoder:

The proposed bus encoder is consisting of a comparator; two majority voters along with a multiplexer. The data input is comprising of 8-bits and its obvious to obtain 8-bit data as output too. A comparison is held between each bit of Din and Dout respectively. So, the comparator will generate the certain sequence of compared bits of inputs and outputs, which then forwarded to the two majority voters. These majority voters will generate selective control signals based upon the data sequences which were inserted into register . when clock is applied at register finally the whole will be multiplexed using multiplexer and at the end the output is taken and also it will be inserted again at the transmitting end for further comparison.

The architecture of majority voter is shown in following fig.

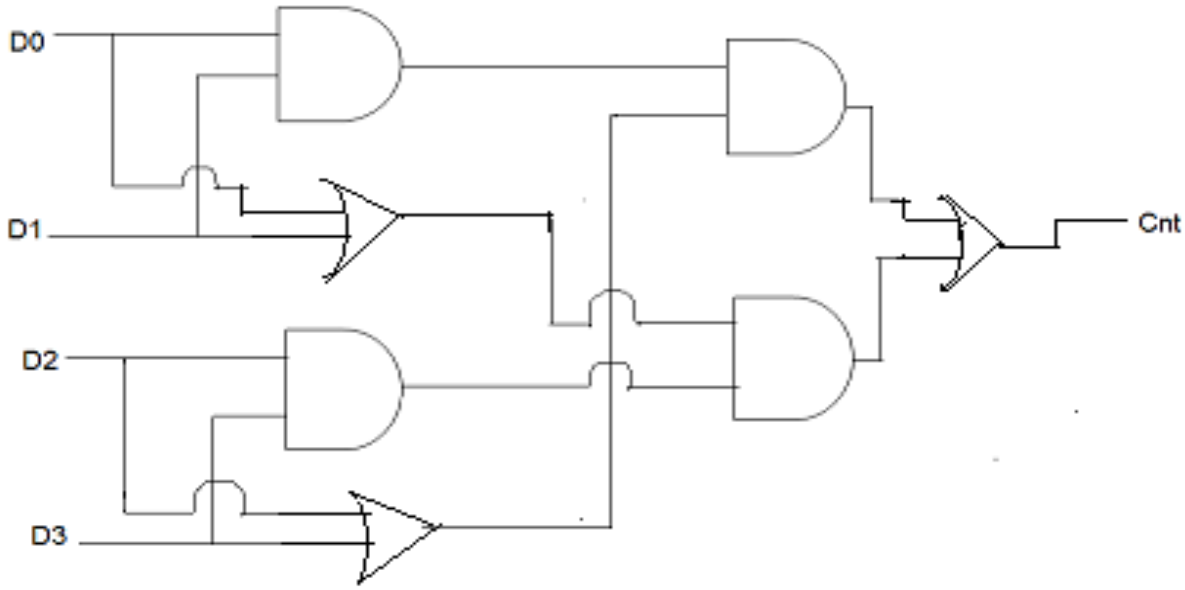


Fig 3.8: proposed majority voter

The compatibility of bus coding method with small N data bus, is the reason for their better performance with small data buses. The larger data buses can be divided into several smaller sub-buses which can be coded individually and particularly. Assuming a random uniform distribution of data bits over a large bus then also the data on the sub-buses are randomly distributed too and these sub-buses are not dependent on each other more likely.

Working :

To reduce the propagation delay we designed a new pipelined based eight bits bus invert encoder which architecture is given in figure. We split proposed 8-bit bus invert encoder in two blocks. They block are independently. In first combination block we compare the input data and current bus data. Then we generate majority voter output signal. The second block generate output data value using majority voter and input data. The first block generated majority voter output connected to the input signals of the register. Same input data signals are connected to the input of register . the each block has less propagation delay than proposed bus invert encoder. Each block operates independently Working on two values at the same time. The left block Start computing for a new data values while the right hand block complete the function for the value started at the last cycle. We have reduced the cycle time of the machine because we have cut the maximum delay through the small combination logic

CHAPTER-4

Results and Simulation

In our work, we will give the images of the results obtained from the previous work as well as from the proposed work. We evaluate the performance of conventional and proposed bus encoders and implement them on Spartan – 3 FPGA families. For Design Entry and delay report we synthesize these using Xilinx ISE 9.2i. We use verilog as hardware description language. This chapter also gives the simulation or we can say the synthesis report obtained from the implementations of past and proposed works which clearly represents the better performance of our proposed work from the past works as there is a noticeable reduction in delay as well as power consumption parameters.

4.1Bus invert encoder:

The convention works over the bus invert encoder has given satisfactory results which are given below. The circuit for the bus invert encoder along with their FPGA implementation and waveforms have been obtained by using Xilinx 9.2i. The technology schematic of bus invert encoder is shown in figure 4.1. The technology schematic of bus invert encoder is shown in figure 4.2. The simulation waveform is also shown in figure 4.4.

Where Din[7:0] is input data and Cnt is output of majority voter , D_out[7:0] is output data or we can say encoded data . We use behavior level modeling in designing in verilog.

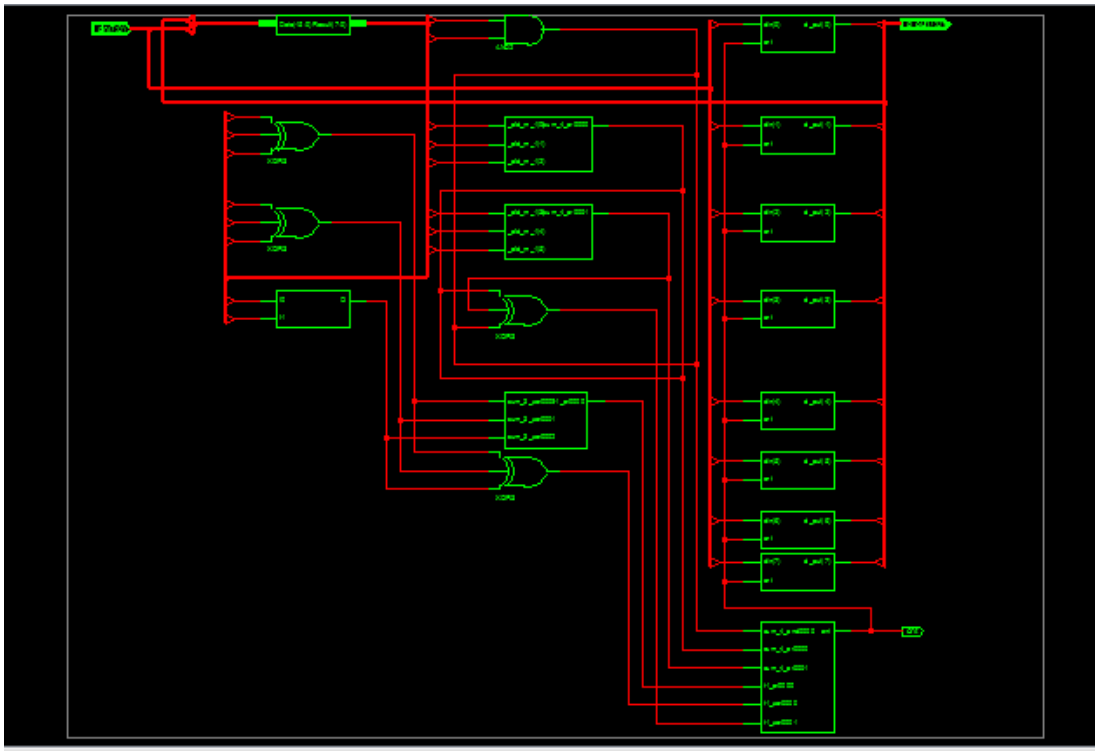


Fig 4.1: RTL view of the bus invert encoder

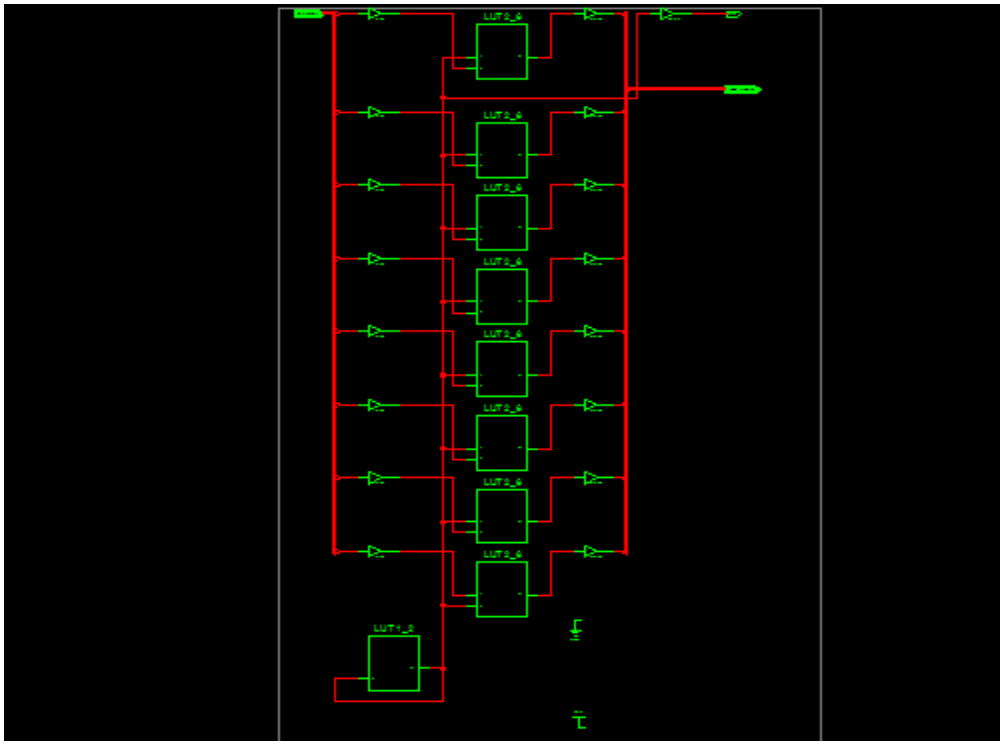


Fig 4.2: FPGA implementation of the bus invert encoder.

```
=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 8 / 8
-------------------------------------------------------------------------
Delay:               7.760ns (Levels of Logic = 3)
  Source:            din<6> (PAD)
  Destination:       d_out<6> (PAD)

  Data Path: din<6> to d_out<6>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             1   0.715   0.976  din_6_IBUF (din_6_IBUF)
     LUT2:I0->O            1   0.479   0.681  d_out<6>1 (d_out_6_OBUF)
     OBUF:I->O                 4.909          d_out_6_OBUF (d_out<6>)
    ----------------------------------------
    Total                      7.760ns (6.103ns logic, 1.657ns route)
                                       (78.6% logic, 21.4% route)

=========================================================================
CPU : 7.22 / 7.57 s | Elapsed : 7.00 / 7.00 s

-->

Total memory usage is 166052 kilobytes
```

Fig 4.3: Synthesis report for the bus invert encoder

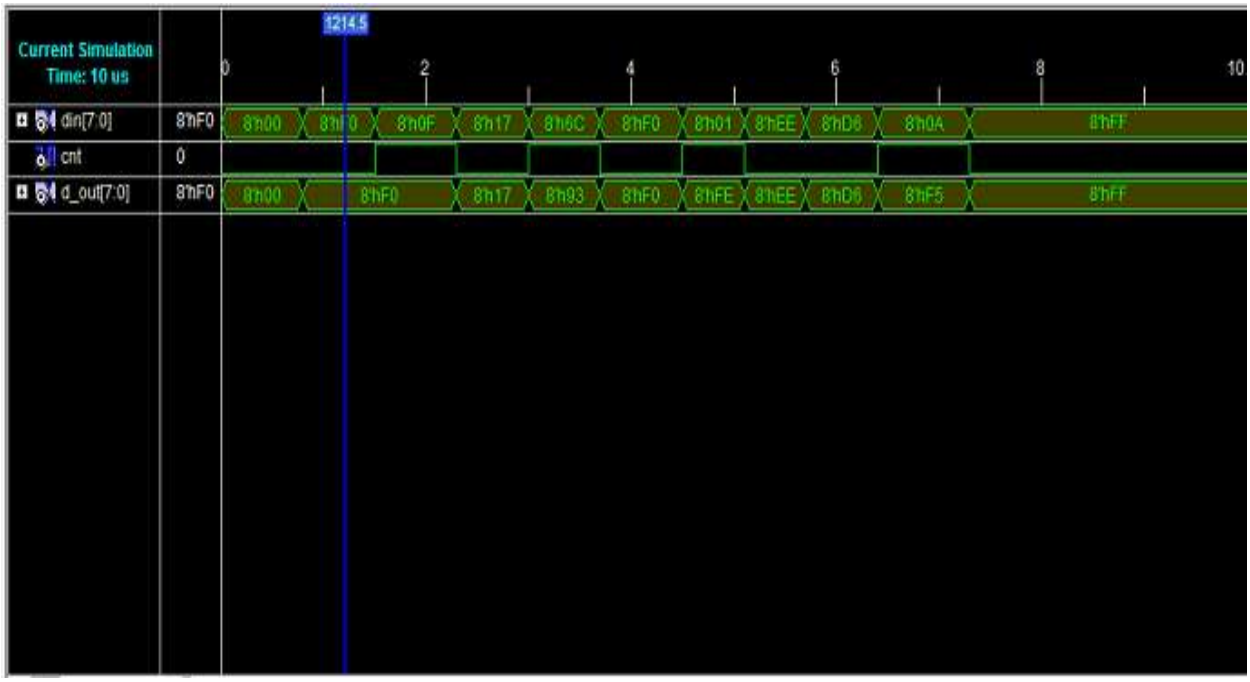


Fig 4.4: Waveforms for the bus invert encoder

4.2Proposed Bus Encoder:

The proposed bus encoder has been designed using bus invert algorithm.We are giving the circuit for the proposed bus encoder along with its synthesis report, waveforms and FPGA implementations.

The proposed works over the bus invert encoder has given satisfactory results which are given below. The circuit for the bus invert encoder along with their FPGA implementation and waveforms have been obtained by using Xilinx 9.2i. The technology schematic of bus invert encoder is shown in figure 4.5. The technology schematic of bus invert encoder is shown in figure 4.6. The simulation waveform is also shown in figure 4.8.

Where Din[7:0] is input data and Cnt[1:0] is output of majority voters, D_out[7:0] is output data or we can say encoded data . We use behavior level modeling in designing in verilog.

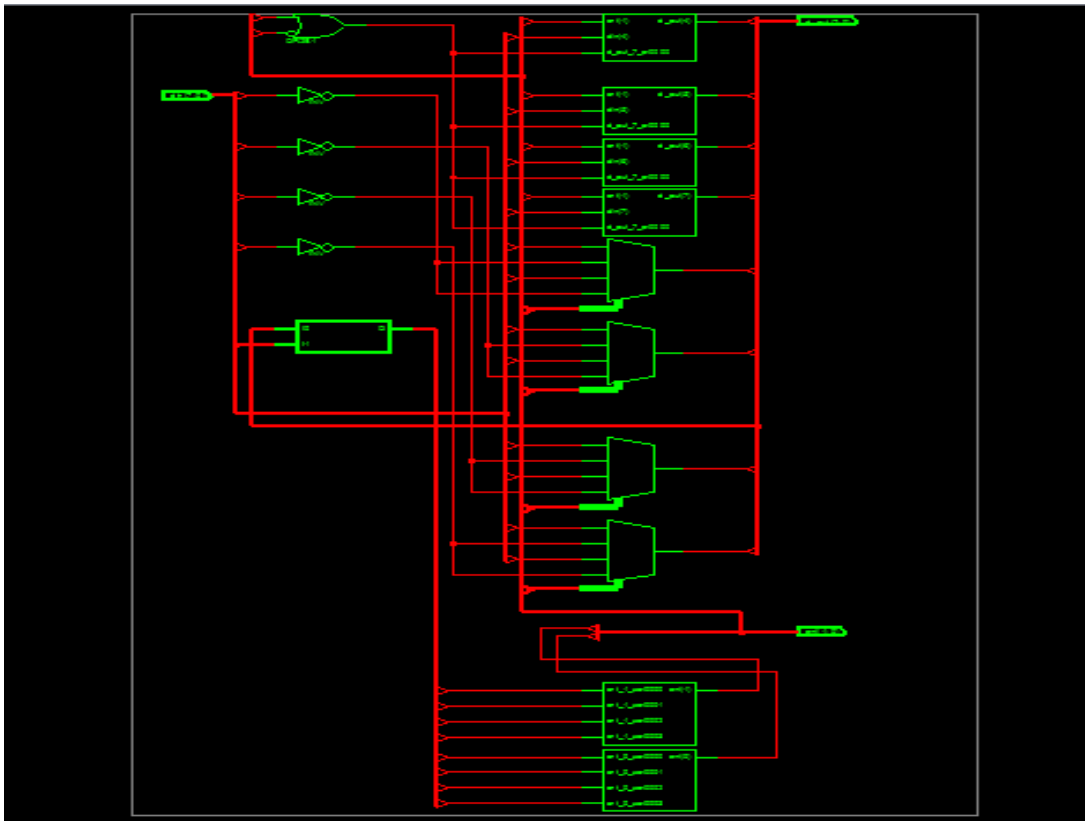


Fig 4.5: RTL view of proposed bus encoder

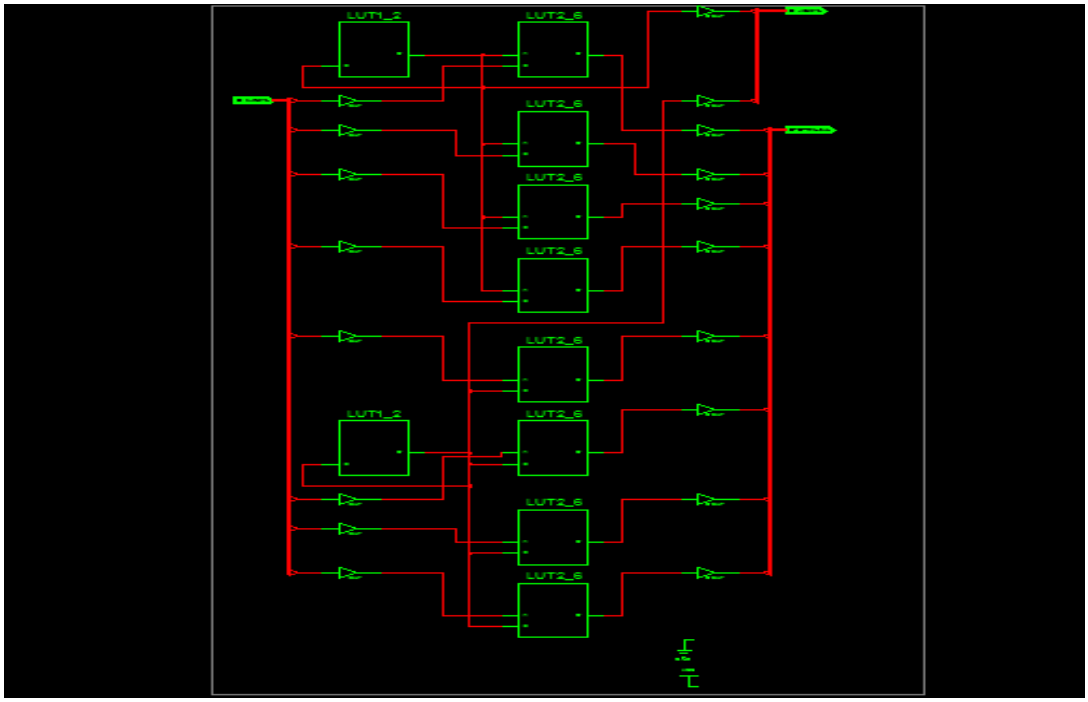


Fig 4.6: FPGA implementation of proposed bus encoder

```
--------------
All values displayed in nanoseconds (ns)

================================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 8 / 8
--------------------------------------------------------------------------------
Delay:               7.760ns (Levels of Logic = 3)
  Source:            din<3> (PAD)
  Destination:       d_out<3> (PAD)

  Data Path: din<3> to d_out<3>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             1    0.715   0.976  din_3_IBUF (din_3_IBUF)
     LUT2:I0->O            1    0.479   0.681  Mmux_d_out<3>11 (d_out_3_OBUF)
     OBUF:I->O                  4.909          d_out_3_OBUF (d_out<3>)
    ----------------------------------------
    Total                      7.760ns (6.103ns logic, 1.657ns route)
                                       (78.6% logic, 21.4% route)

================================================================================
CPU : 7.93 / 8.14 s | Elapsed : 8.00 / 8.00 s

-->
```

Fig 4.7: Synthesis report for the proposed bus encoder

Proposed bus encoder waveforms:

The following are the waveforms obtained for proposed bus encoder:

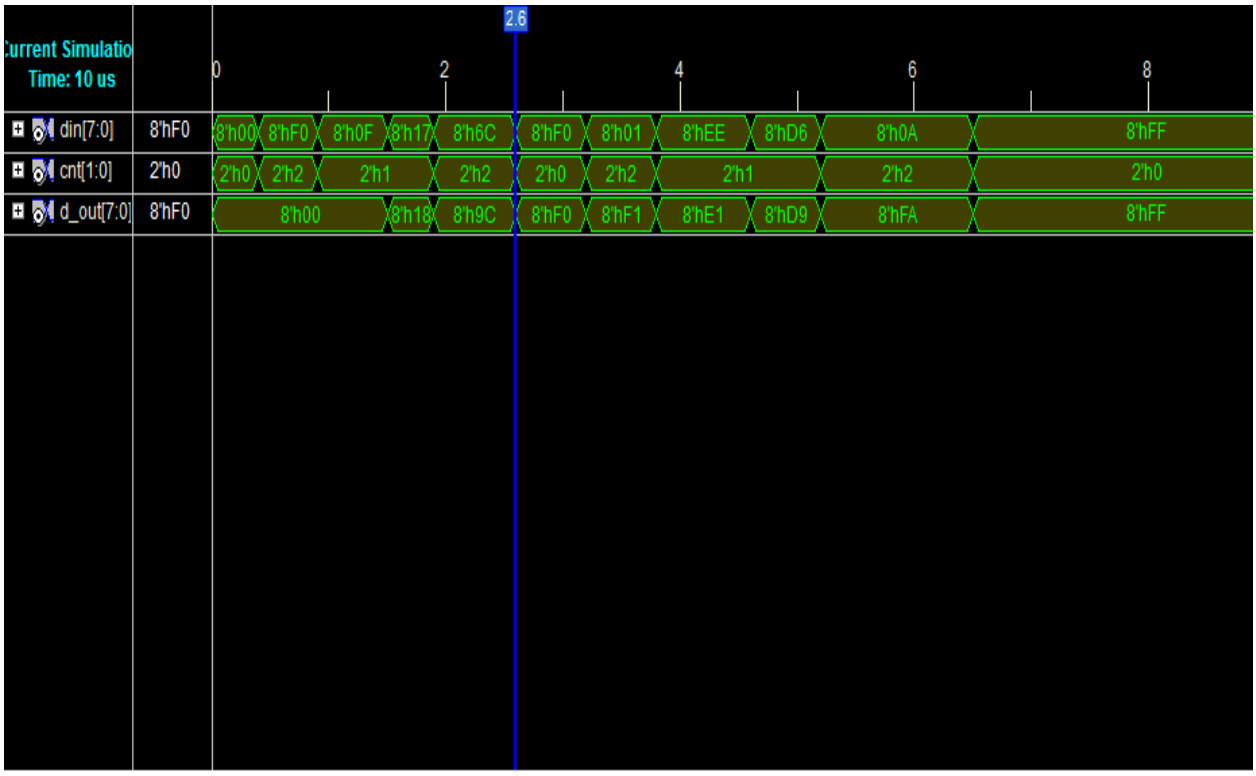


Fig 4.8: Waveforms for proposed bus encoder

4.3 Proposed pipelined Bus Encoder:

The pipelined proposed bus encoder has been designed using bus invert algorithm and pipelining concept in such manner that the delay has been reduced. We are giving the circuit for the proposed pipelined bus encoder along with its synthesis report, waveforms and FPGA implementations.

The proposed works over the pipelined bus invert encoder has given satisfactory results which are given below. The circuit for the pipelined bus invert encoder along with their FPGA implementation and waveforms have been obtained by using Xilinx 9.2i. The technology schematic of bus invert encoder is shown in figure 4.10. The simulation waveform is also shown in figure 4.12.

Where Din[7:0] is input data and Clk is input clock, D_out[7:0] is output data or we can say encoded data . We use behavior level modeling in designing in verilog.

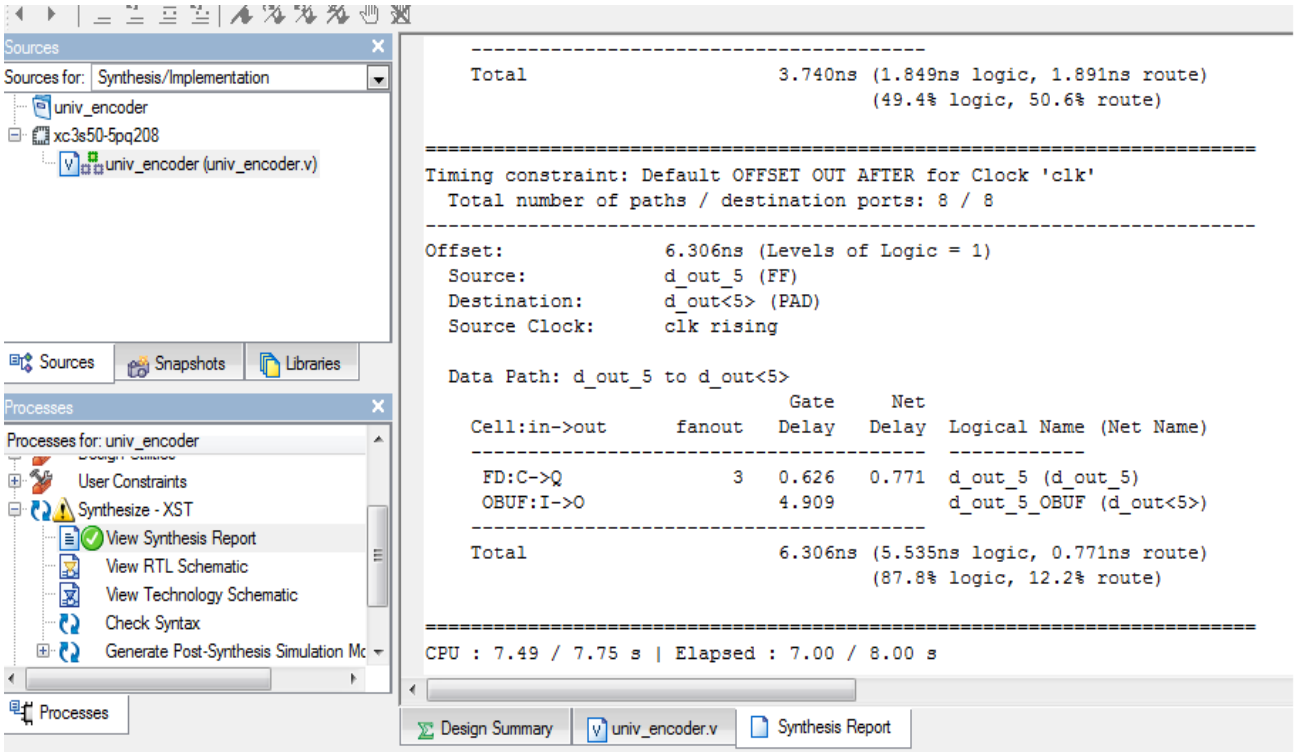


Fig4.9: Synthesis report for the proposed pipelined bus encoder

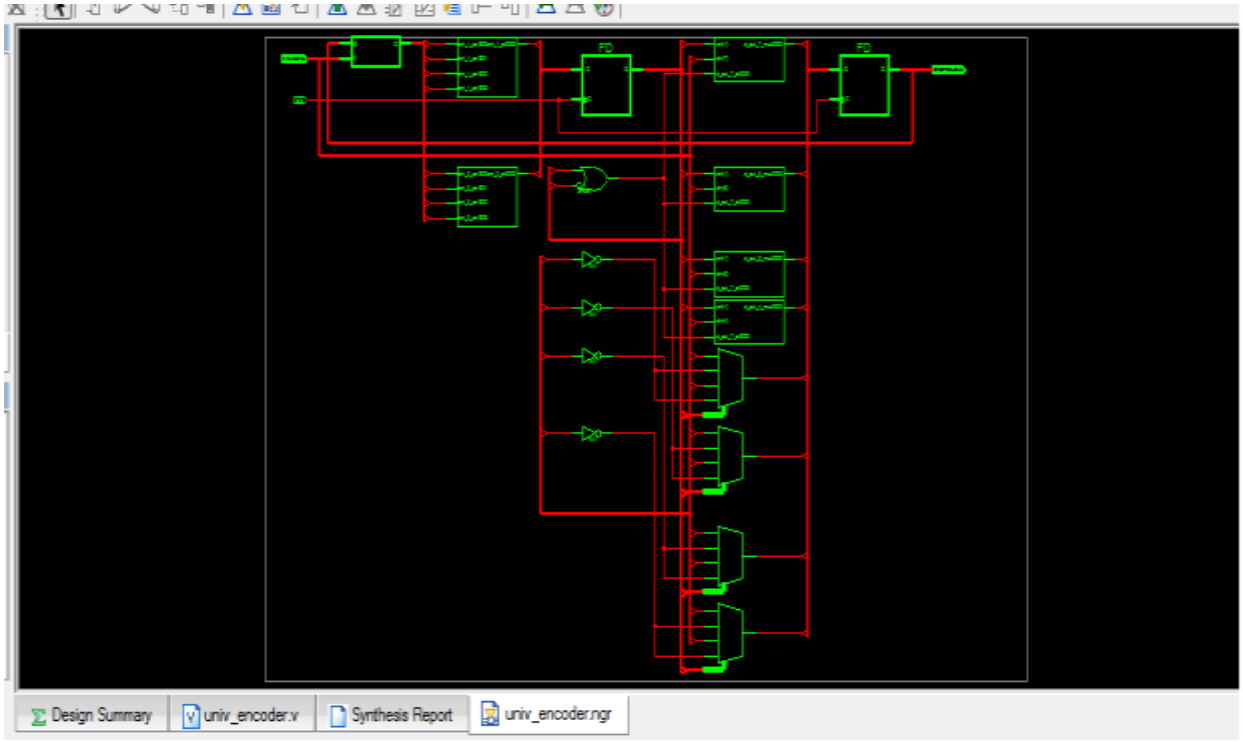


Fig4.10:RTL view of proposed pipelined bus encoder

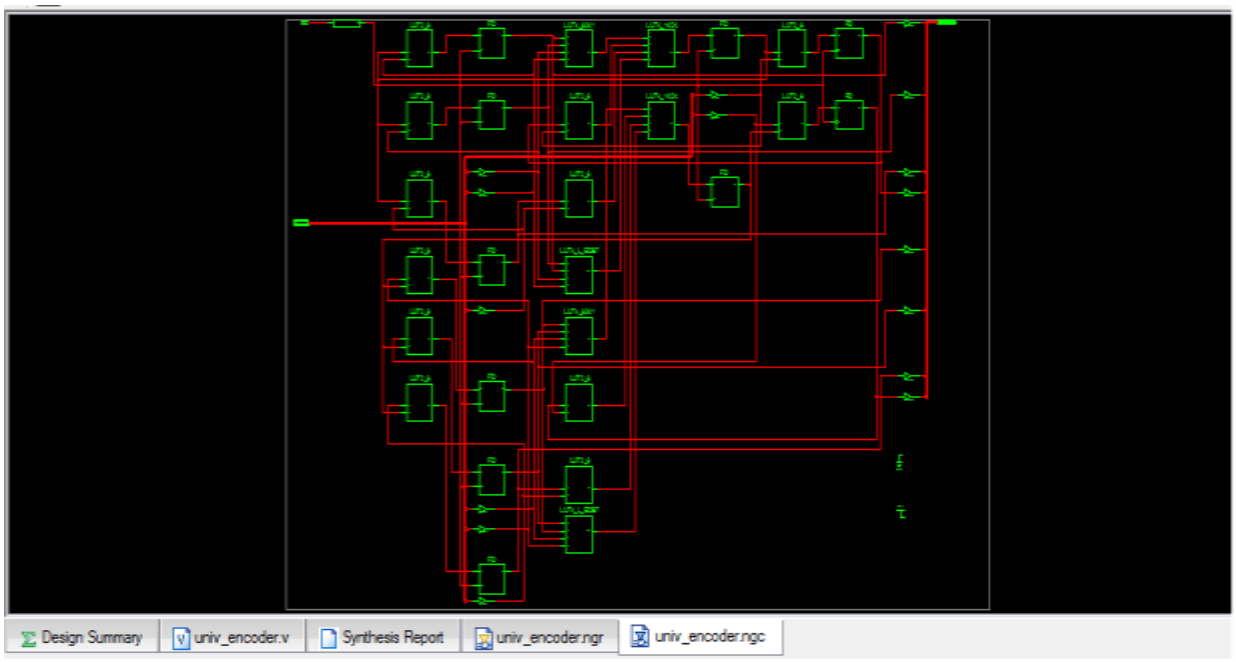


Fig 4.11: FPGA implementation of proposed pipelined bus encoder

Proposed pipelined waveforms:

The following are the waveforms obtained for pipelined proposed bus encoder:

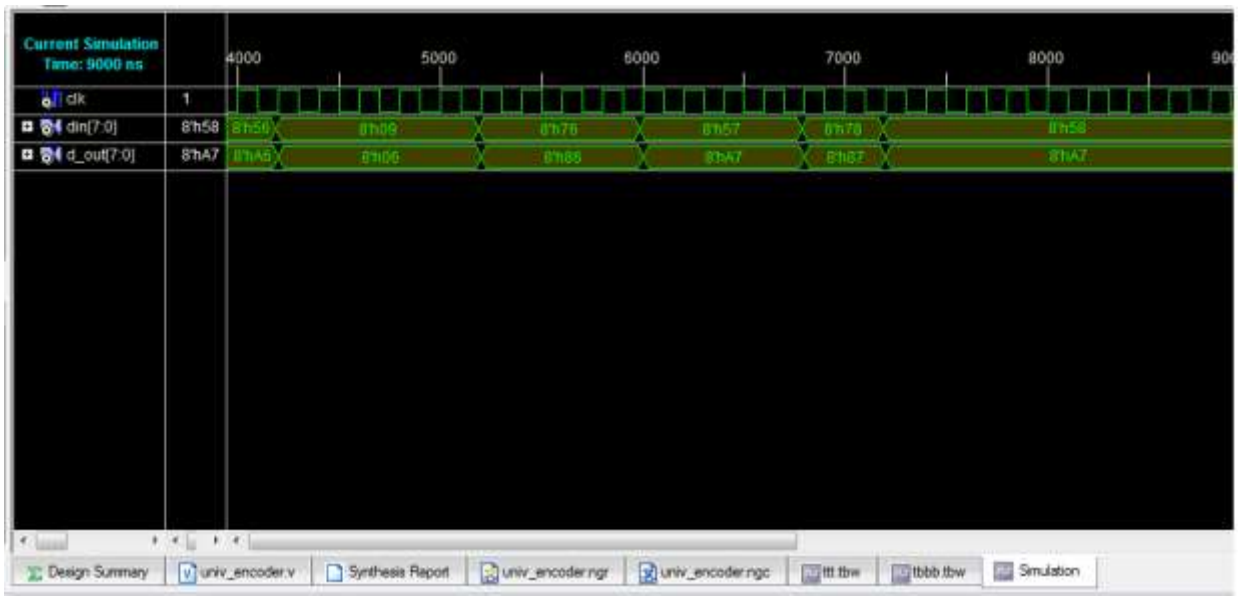


Fig4.12: Waveforms for proposed pipelined bus encoder

4.4 Comparison table:

The Switching Comparison of bus invert encoder and Proposed Bus invert encoder is shown in table 1. The Delay Comparison of bus invert encoder and Proposed pipelined Bus invert encoder is shown in table 1.

4.4.1 Switching Comparison

Sr. No.	Bus encoder	No. of switching = 4	No. of switching = 3
1.	Bus invert encoder	5	4
2.	Proposed Bus invert encoder	1	**2**

The chances of four switching in conventional Bus invert encoder is 5 and The chances of four switching in proposed Bus invert encoder is 1. The proposed Bus invert encoder has 80% less chances of four switching than conventional Bus invert encoder.

The chances of three switching in conventional Bus invert encoder is 4 and The chances of three switching in proposed Bus invert encoder is 2. The proposed Bus invert encoder has 50% less chances of four switching than conventional Bus invert encoder.

We can say that proposed bus encoder is more efficient than conventional Bus invert encoder in the term of switching activities as well as in power consumption.

4.4.2 Delay Comparison

Sr. No.	Bus encoder	delay
1.	Bus invert encoder	7.760ns
2.	Proposed pipelined Bus invert encoder	6.306ns

The propagation delay in conventional Bus invert encoder is 7.760ns and propagation delay in Proposed pipelined Bus invert encoder is 6.306. The proposed pipelined Bus invert encoder has 18.75% less propagation delay than conventional Bus invert encoder.

We can say that proposed pipelined bus encoder is more efficient than conventional Bus invert encoder in the term of propagation delay as well as in power consumption.

Experimental results also demonstrate that proposed bus encoders are more efficient than conventional Bus invert encoder.

CHAPTER-5

Conclusion and future scope

5.1 Conclusion:

We have designed a new pipelined bus invert encoder which is surely better than the previous bus invert encoder, in terms of speed, delay and power consumption aspects, which are the major factors responsible for certain system's performance and proven to be the key parameters especially in case when we are talking in terms of VLSI/ULSI technologies which are basically the micron and deep sub-micron technologies of circuit designing respectively.

The proposed pipelined Bus invert encoder has 18.75% less propagation delay than conventional Bus invert encoder.

5.2 Future Scope:

Here, in our proposed work, we have used the basic concepts of 'bus invert coding' and 'pipelining' to design a newer and better partitioned bus invert encoder, which is surely better than the previous ones in a number of circuit performance aspects. Although, the 'bus invert method' has been described in accordance with the dynamic power dissipation, which can further be applied in similar cases where large capacitances are included. This concept can also be applied for larger data buses as 16-bit, 32-bit, 64-bit and so on. We have proposed a better design for 8-bit data bus. In the end, we can say that our work is better than the previous works and also efficient in terms of power consumption. This whole concept can further be used to attain more better results in future. We can also use state machine approach to increase the speed . state machine is a sequential circuit which can reduced delay path.

For power reduction of bus encoder we can use following techniques-

Clock gating technique: For decreasing the dynamic power dissipation in a number of synchronous systems, clock gating is well known technique in field of digital systems. In order to prune the clock tree, some amount of the power is conserved by this technique of clock gating by the addition of more logics in certain circuit.

A part of the circuitory is inactivated by this pruning of the clock tree, so that switching of the states by the flip-flops is not required anymore. And transition or we can say switching definitely

consumes certain amount of power in the circuit. And when there is no switching happened, the power consumption due to that switching activity also becomes zero and only amount of leakage currents are observed from the circuit.

Reversible logic technique: Following are the properties which are incurred by the gate named as “reversible gate”:

- The input vector can be observed with the help of output vector uniquely.
- One to one correspondence scheme is to be followed between the assigned values of input and output respectively.
- Fan out of more than one, is the necessary condition. As it has to be there in the logic circuit.
- No feedback is recommended in the reversible logics, there is no feedback.

No energy is being dissipated by the circuits which are designed with the help of reversible logic circuit as there is no internal power, the internal power value is zero.

APPENDIX A

A.1 Implementation of previous bus encoder:

```
module bus_paper(din, cnt, d_out);

  input [7:0] din;

  output reg cnt;

  output reg[7:0] d_out;

        reg [7:0] m;

        reg [4:0] sum , cout;

        reg t1 ,t2,t3,y;


initial d_out = 8'd0;

initial cnt= 1'b0;


always@(*)

begin

 m = d_out^din;


sum[0]= (m[0]^m[1])^m[2];

cout[0]= ((m[0]&m[1])|(m[1]&m[2])|(m[2]&m[1]));
```

```
sum[1]= (m[3]^m[4])^m[5];

cout[1]= ((m[3]&m[4])|(m[4]&m[5])|(m[5]&m[3]));


sum[2]= (m[6]^m[7]);

cout[2]= (m[6]&m[7]);


  sum[3]= (sum[0]^sum[1])^sum[2];

  cout[3]= ((sum[0]&sum[1])|(sum[1]&sum[2])|(sum[2]&sum[1]));


   sum[4]= (cout[0]^cout[1])^cout[2];

  cout[4]= ((cout[0]&cout[1])|(cout[1]&cout[2])|(cout[2]&cout[1]));


            t1 = sum[3]&cout[3]&sum[4];

    t2 = sum[3]|cout[3]|sum[4];

    t3 = cout[4]&t2 ;

    y = t1|t3 ;

cnt =y;


if(cnt)

d_out = ~din;

else
```

```
 d_out = din;
end
endmodule
```

A.2 Proposed bus encoder:

```
module bus_encoder_new(din, d_out, cnt);
    input [7:0] din;
    output reg [7:0] d_out;
    output reg [1:0] cnt;
            initial d_out = 8'd0;
            reg [7:0] m;
 always@(*)
begin
 m[0] = din[0]^d_out[0];
 m[1] = din[1]^d_out[1];
 m[2] = din[2]^d_out[2];
 m[3] = din[3]^d_out[3];
 m[4] = din[4]^d_out[4];
 m[5] = din[5]^d_out[5];
 m[6] = din[6]^d_out[6];
 m[7] = din[7]^d_out[7];
```

```
cnt[0] = ((m[1]&m[0]) & (m[2] | m[3]) ) | ((m[3]&m[2]) & (m[1] | m[0]) ) ;

cnt[1] = ((m[5]&m[4]) & (m[6] | m[7]) ) | ((m[7]&m[6]) & (m[5] | m[4]) ) ;

if(cnt==2'b00)

begin

        d_out[7:4] = din[7:4];

                d_out[3:0]= din[3:0];

end


else if(cnt==2'b01)


begin

        d_out[7:4] = din[7:4];

                d_out[3:0]= ~din[3:0];

end

else if(cnt==2'b10)

begin

        d_out[7:4] = ~din[7:4];

                d_out[3:0]= din[3:0];

end

else
```

```
begin
        d_out[7:4] = ~din[7:4];
                d_out[3:0]= ~din[3:0];
end
end
endmodule
```

A.3 Proposed pipelined bus encoder:

```
module univ_encoder(din,clk, d_out);
   input [7:0] din;
   output reg [7:0] d_out;
    reg [1:0] cnt;
          input clk;
           initial d_out = 8'd0;
           reg [7:0] m;


always@(posedge clk )
begin
 m[0] = din[0]^d_out[0];
 m[1] = din[1]^d_out[1];
 m[2] = din[2]^d_out[2];
 m[3] = din[3]^d_out[3];
 m[4] = din[4]^d_out[4];
 m[5] = din[5]^d_out[5];
 m[6] = din[6]^d_out[6];
 m[7] = din[7]^d_out[7];
```

```
cnt[0] = ((m[1]&m[0]) & (m[2] | m[3]) ) | ((m[3]&m[2]) & (m[1] | m[0]) ) ;
cnt[1] = ((m[5]&m[4]) & (m[6] | m[7]) ) | ((m[7]&m[6]) & (m[5] | m[4]) ) ;
end

always@(posedge clk )
begin
if(cnt==2'b00)
begin
        d_out[7:4] = din[7:4];
                d_out[3:0]= din[3:0];
end

else if(cnt==2'b01)

begin
        d_out[7:4] = din[7:4];
                d_out[3:0]= ~din[3:0];
end
else if(cnt==2'b10)

begin
        d_out[7:4] = ~din[7:4];
                d_out[3:0]= din[3:0];
end

else
begin
        d_out[7:4] = ~din[7:4];
                d_out[3:0]= ~din[3:0];
end
```

```
end
endmodule
```