International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 9
ISSN: 2454-4248
137 – 142

# Design and Implementation of Area and Power Efficient Low Power VLSI Circuits through Simple Byte Compression with Encoding Technique

Vithyalakshmi.N
Associate Professor , Electronics and Communication Engineering
Sri vidyanikethan Engineering College
Tirupati, India
vidhyavinoth@gmail.com

Vinoth.G.S
Project Lead
QuEST Global, Technopark
Trivandrum, India
vinothgs@gmail.com

***Abstract***— Transition activity is one of the major factors for power dissipation in Low power VLSI circuits due to charging and discharging of internal node capacitances. Power dissipation is reduced through minimizing the transition activity by using proper coding techniques. In this paper Multi coding technique is implemented to reduce the transition activity up to 58.26%. Speed of data transmission basically depends on the number of bits transmitted through bus. When handling data for large applications huge storage space is required for processing, storing and transferring information. Data compression is an algorithm to reduce the number of bits required to represent information in a compact form. Here simple byte compression technique is implemented to achieve a lossless data compression. This compression algorithm also reduces the encoder computational complexity when handling huge bits of information. Simple byte compression technique improves the compression ratio up to 62.5%. As a cumulative effort of Simple byte compression with Multi coding techniques minimize area and power dissipation in low power VLSI circuits.

***Keywords***— *Compression, delay, Interconnect, Power dissipation, Transition activity, multi coding.*

*****

## I. INTRODUCTION

Novel VLSI technology has resulted in more interconnects, increasingly limits the area, power dissipation and performance of new processors [1]. In all, electronic design increasing the data rate typically results in higher power dissipation and vice versa. The only option of decreasing the transition activity without increasing the energy dissipation is to decrease the output capacitance is achieved by decreasing the size of the devices. As a design rule, shrinking beyond 0.25 µm, interconnection dimensions should also be reduced. Interconnects have an increasing effect on the area, power dissipation of circuits and delay. Reduction in dimensions causes several effects like decrease of gate delays due to thinning of gate oxide and increase in interconnect resistance due to shrinking wire widths. Also the aspect ratios of interconnects have to be increased to compensate for increasing interconnect resistance. Similarly the lateral and fringing capacitance dominates the total capacitance of interconnects [2]. In deep sub-micron technology, the performance of a high speed circuit is largely dependent on interconnects which connect different macro cells within a VLSI chip [3] The main goal is to minimize the power consumption in VLSI interconnects while transmitting huge bits of information. If the interconnects are scaled down, it results in RC delays that begin to dominate the chip performance.

## II. BUS MODEL

The scaling of wire dimension not only affects delay time but also all related interconnect performance such as power dissipation, reliability and bandwidth for local and global interconnects. The ever decreasing interconnect cross section dimension gives rise to increase in resistance [4].VLSI chips are not just transistors alone, but they also have to be connected and also should be supplied with power and ground. For interconnect dimension analysis, the wire dimension is as equally as important as the transistor dimension. They affect speed, burn power and can wear out too. Till the 0.18 µm technology the wires were aluminum. But the modern process used copper. The resistivity of different materials is listed in Table 1. In today's technology wires form complex geometries that introduce parasitic elements like resistive, capacitive and inductive, which affect performance by increasing signal propagation delay, energy dissipation, power distribution and also introduces extra noise sources which affect circuit reliability. Bus energy model is classified in to lumped RC model and distributed RC model

Table. 1. Resistivity of different metals

| Sl. No | Metal | Bulk resistivity |
|---|---|---|
| 1 | Silver(Ag) | 1.6 |
| 2 | Copper(Cu) | 1.7 |
| 3 | Gold(Au) | 2.2 |
| 4 | Aluminum(Al) | 2.8 |
| 5 | Tungsten(W) | 5.3 |
| 6 | Molybdenum(Mo) | 5.3 |

### *A. Lumped RC Model*

Here the total resistance of the wire can be lumped in to single resistor 'R' and the global capacitance of the wire is combined in to a single capacitor 'C' is shown in Fig.1

 ISSN: 2454-4248
137 – 142

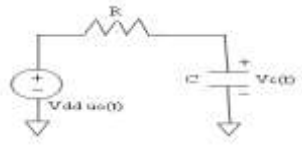

Fig.1. Lumped RC Model

$$I_{int} = C\frac{dv_c(t)}{dt} \quad (1)$$

$$\frac{v_{dd}u(t) - v_c(t)}{R} = C\frac{dv_c(t)}{dt} \quad (2)$$

By applying Laplace Transform we get,

$$Vdd\frac{u(t)}{R} - \frac{Vc(t)}{R} = C\frac{dVc(t)}{dt} \quad (3)$$

$$\frac{Vdd}{SR} - \frac{Vc(s)}{R} = CsVc(s) \quad (4)$$

(5)

$$Vc(s) = \frac{Vdd}{RC}\cdot\frac{1}{s\left(s+\frac{1}{RC}\right)}$$

$$Vc(s) = Vdd\left[\frac{1}{s} - \frac{1}{\left(s+\frac{1}{RC}\right)}\right] \quad (6)$$

$$Vc(t) = Vdd\left[1 - e^{-\frac{t}{RC}}\right]uo(t) \quad (7)$$

Time delay ( 50% and 90 %)

$$Vc(t) = Vdd\left[1 - e^{-\frac{t}{RC}}\right] \quad (8)$$

$t_{0.5} = 0.693RC$ (9)

$t_{0.9} = 2.3\ RC$ (10)

### *B. Distributed RC Model*

A wire can be represented in to several RC sections. The delay of the circuit is approximated as shown in Fig.2.

$$Td = \sum_{j=1}^{n}\frac{C}{N}\sum_{k=1}^{n}\frac{R}{N} \quad (11)$$

$$Td = \frac{C}{N}X\frac{R}{N}\left(\frac{N(N+1)}{2}\right) \quad (12)$$

$$Td = RC\left(\frac{N+1}{2N}\right) \quad (13)$$

$$Td = \frac{RC}{2}\ for N = \infty \quad (14)$$

N – is the number of sections

If N becomes large the sections reduces to

$$Td = \frac{RC}{2}\left(\frac{r1}{w}\right)(calw + 2Cp(l+w)) \quad (15)$$

$$Td = \frac{1}{2}rCal^2 \quad (16)$$

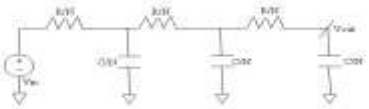

Fig.2. Distributed RC Model

r – Resistance per unit length

c – Capacitance per unit length

l – Wire length

Here RC effects dominate very long wires because the delay is proportional to the square of the length. Doubling the length of the wire will quadruple the delay.

## III. POWER ESTIMATION

Power estimation means, in general the technique used to estimate the average power dissipation of circuits. Power can be estimated in the below four ways

- Circuit Level
- Gate Level
- Architectural Level
- Behavioral Level

Circuit level is the most straight forward method of power estimation is circuit level, and here we estimate the average power. Architectural level is represented by functional blocks and the complexity of the design, at this level, is relatively low compared to the circuit and gate level. Functional blocks can be implemented by using gate count method, power factor approximation method and dual bit type model. Behavioral model describes the function of a system versus a set of inputs. The power estimation at the behavioral level relates the consumed energy to execution of algorithm. There are two techniques for power estimation at the gate level, probabilistic method and Event driven simulation. In probabilistic method the power dissipation can be analyzed using pattern independent. Event Driven technique uses a cell library that has been characterized by static and dynamic power dissipation.

### *A. Static Power*

It is the power consumed during standby mode and CMOS gates have sub threshold leakage current even when the gates are not turned on. The major issue of static power dissipation is drain to source leakage current. This dissipated power is a very negligible. In a usual chip 10% of the power consumed is leakage and 90% is dynamic power. So, clearly the major concern is dynamic power dissipation.

### *B. Dynamic Power dissipation*

A dynamic power vector describes an event in which power is dissipated due to a signal switching at the cell inputs during charging and discharging of load capacitance. The power dissipation can be estimated by the load capacitance $C_L$. This power loss is due to the charging and discharging of Load Capacitance $C_L$ is shown in Fig.3. The average dynamic power $P_D$ is required to charge and discharge a capacitance $C_L$ at a switching frequency $f_{sw}$

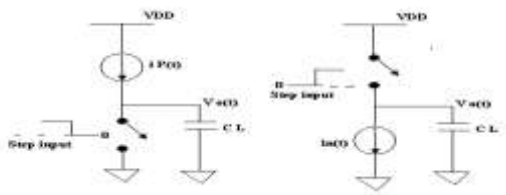


Fig.3. Equivalent circuit for dynamic power calculation

 ISSN: 2454-4248
137 – 142

$$P_D = f_{sw}\int_0^T io(t)Vo(t)dt \quad (17)$$

During charging cycle

$$i_p = C_L \cdot \frac{dvo}{dt} \quad (18)$$

During discharge cycle

$$i_n = -C_L \cdot \frac{dvo(t)}{dt} \quad (19)$$

$$P_D = f_{sw}[\int_0^{V_{DD}} C_L V_o dVo - \int_{V_{DD}}^0 C_L V_o dVo] \quad (20)$$

$$P_D = f_{sw}[C_L[\frac{Vo^2}{2}]_0^{VDD} - [\frac{Vo^2}{2}]_{VDD}^0] \quad (21)$$

$$P_D = f_{sw}[C_L[\frac{V_{DD}^2}{2} + \frac{V_{DD}^2}{2}]] \quad (22)$$

$$P_D = f_{sw} C_L V_{DD}^2 \quad (23)$$

Assuming a logic gate goes through one complete charge/discharge cycle for every clock cycle, suppose the system clock frequency is f.

Let $f_{sw} = E_f$ ; Where E is the energy transition activity factor

Most gates do not switch every clock cycle,

$$P_D = EC_L V_{DD}^2 f \quad (24)$$

If the signal is clock , E=1 it charges and discharges every clock cycle, but most data have a maximum Energy Transition activity factor E=0.5 because they transit only once every cycle

## IV. RELATED WORK

In recent years many research works are carried out in the power dissipation of VLSI circuits. There are two categories to lessen the power dissipation by switching activity. First category reduces the switching activity in address buses. A2BC [5], the beach solution and gray coding are some among them. These methods use the redundancy bits to reduce the power dissipation. The second category uses the reduced switching activity for minimizing power dissipation in data buses. Bus invert coding [6] is one of the best methods to reduce the self transitions. DSM bus invert coding [7], Bus regrouping [8], Low power bus coding techniques considering inter wire capacitances [9], and partitioned bus coding for energy reduction [10] are some of the efficient methods to reduce the coupling transitions. Coding for energy minimization in VLSI interconnects [11]. However, among all the above methods only bus coding technique is considered to reduce power dissipation. This paper considers simple byte compression with bus coding technique to reduce the transition activities and encoder complexity. These combined techniques effectively reduces the power dissipation and area in the on – chip data buses.

## V. PROPOSED CODING SCHEME

It is always useful to code information using less number of bits than the original representation. There are lot of applications in which information size would be critical. Simple byte compression consists of two components, an encoder that takes a message and generates a compressed data and an decoder that reconstructs the original message from the compressed representation [12] is shown in Fig.4. Lossless and Lossy compression are the most popular compression techniques. In lossless compression all the information are compressed and reconstruction is exact to original data. In lossy compression few of the information incorporate in the original data is irrecoverably lost. The reason data compression is needed that most of the information that are generated and used in digital format is in the form of numbers and represented by bytes of data. If the application is multimedia, the number of byte required to represent the data can be very large. This huge number of data is coded and transmitted as it is, we face encoder and decoder complexity. This will increase Transition activity, area, delay and power dissipation. To overcome these effects a simple byte compression algorithm with multi coding Technique has been implemented. Simple byte compression algorithm is originally implemented for text message and bulk data transmission. This algorithm compresses the 8 byte of information in to 5 byte information. Fig.5 shows block diagram of Encoder/Decoder. It consists of binary converter, byte compression algorithm, multi coding technique and Transition estimator.

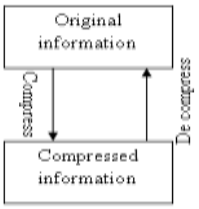


Fig.4. Simple compression principle

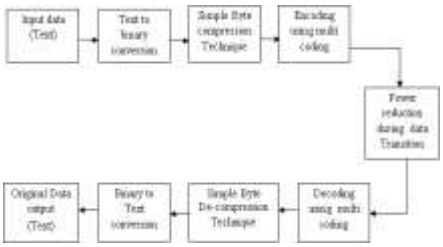

Fig.5. Block diagram for encoder / decoder

### A. *Binary Conversion*

The received inputs are random samples like simple text message. Foremost each character of random sample is assigned by a decimal equivalent value as shown in the following Table.2.Then each decimal value is converted into equivalent binary form.

TABLE. 2. LOOKUP TABLE FOR TEXT CHARACTERS

| Character | Value | Character | Value | Character | Value | Character | Value |
|---|---|---|---|---|---|---|---|
| a | 1 | a | 15 | C | 29 | Q | 43 |
| b | 2 | p | 16 | D | 30 | R | 44 |
| c | 3 | q | 17 | E | 31 | S | 45 |
| d | 4 | r | 18 | F | 32 | T | 46 |
| e | 5 | s | 19 | G | 33 | U | 47 |
| f | 6 | t | 20 | H | 34 | V | 48 |
| g | 7 | u | 21 | I | 35 | W | 49 |
| h | 8 | v | 22 | J | 36 | X | 50 |
| i | 9 | w | 23 | K | 37 | Y | 51 |
| j | 10 | x | 24 | L | 38 | Z | 52 |
| k | 11 | y | 25 | M | 39 | - | 0 |
| l | 12 | z | 26 | N | 40 | - | 53 |
| m | 13 | A | 27 | O | 41 | - | 54 |
| n | 14 | B | 28 | P | 42 | - | - |

 ISSN: 2454-4248
137 – 142

### B. *Simple byte compression algorithm*

Assume random sample with 8 Characters long, each character in the random sample need 8 bits to represent its binary form. Exclusively 64 bits are required to represent the full random sample. By using simple byte compression algorithm 5 bits are enough to represent a character instead of using 8 bits. Finally 40 bits are enough to represent the same 8 character random sample. The following steps will explain how the compression is achieved effectively.

Step 1: Assign characters value from Table 2

Step 2: Assign 8 bit binary value to corresponding number

$K_E$=/$k_0$,$k_1$,$k_2$,$k_3$,$k_4$,$k_5$,$k_6$,$k_7$/$k_8$$k_9$,$k_{10}$,$k_{11}$,$k_{12}$,$k_{13}$,$k_{14}$,$k_{15}$/$k_{16}$,$k_{17}$,$k_{18}$, $k_{19}$,$k_{20}$,$k_{21}$,$k_{22}$,$k_{23}$ / $k_{24}$, $k_{25}$,$k_{26}$—$k_n$

Where n = 64 bit

K = Input data

Step 3 : Then remove each byte from the position of 3rd bit from left side and isolate 5 least significant bits

$K_E$=/$k_3$,$k_4$,$k_5$,$k_6$,$k_7$/$k_{11}$,$k_{12}$,$k_{13}$,$k_{14}$,$k_{15}$/$k_{19}$,$k_{20}$,$k_{21}$,$k_{22}$,$k_{23}$/$k_{27}$,$k_{28}$, $k_{29}$,$k_{30}$,$k_{31}$/$k_{35}$,$k_{36}$,$k_{37}$,$k_{38}$,$k_{39}$ / ----$k_{64}$ /

Step 4 : Then rearrange the isolated 5 bits properly in an array of bytes, finally the 8 byte data is reduced in to 5 byte.

$K_E$=/$k_3$,$k_4$,$k_5$,$k_6$,$k_7$,$k_{11}$,$k_{12}$,$k_{13}$/$k_{14}$,$k_{15}$,$k_{19}$,$k_{20}$,$k_{21}$,$k_{22}$,$k_{23}$,$k_{27}$/ $k_{28}$,$k_{29}$,$k_{30}$,$k_{31}$,$k_{35}$,$k_{36}$,$k_{37}$,$k_{38}$ / $k_{39}$------- / ----- $k_{64}$ /

### C. *Multi Coding Technique*

In this section the compressed input data are coded in eight different ways such as Invert, Circular Left Shift, Invert even position, Swap, Invert odd position, Rotate left with invert, Circular Right Shift and Rotate Right with invert. Then 3 bit control signals are added to coding techniques in order to recover the original information at the decoder. Data Transition During data transmission via buses, the input data changes from logic '0' to logic '1' and vice versa is called data transition. Number of data transition is calculated by using hamming distance estimator. Where hamming distance'd' is equivalent to data transition. The hamming distance estimator estimate hamming distance'd' between the coded input data with reference data. Similarly hamming distance is calculated for all possible input data .Among all least hamming distance'd' is transmitted to the decoder.

### D. *Decoder*

The received data are decoded properly based on the 3 bit control word and effectively decompressed by simple byte decompression algorithm. The value got by decompression is then mapped from look up table and thus the original message is reconstructed.


Fig. 6. Output of text data compression

### E. *Simple byte decompression algorithm*

Step1: After decoding we will receive 5 byte of information.

$K_D$=/$k_3$,$k_4$,$k_5$,$k_6$,$k_7$,$k_{11}$,$k_{12}$,$k_{13}$/$k_{14}$,$k_{15}$,$k_{19}$,$k_{20}$,$k_{21}$,$k_{22}$,$k_{23}$,$k_{27}$/ $k_{28}$,$k_{29}$,$k_{30}$,$k_{31}$,$k_{35}$,$k_{36}$,$k_{37}$,$k_{38}$ / $k_{39}$------- / ----- $k_{64}$ /

Step2: Each byte can be split in to a set of 5 bits

$K_D$=/$k_3$,$k_4$,$k_5$,$k_6$,$k_7$,$k_{11}$,$k_{12}$,$k_{13}$/$k_{14}$,$k_{15}$,$k_{19}$,$k_{20}$,$k_{21}$,$k_{22}$,$k_{23}$,$k_{27}$/ $k_{28}$,$k_{29}$,$k_{30}$,$k_{31}$,$k_{35}$,$k_{36}$,$k_{37}$,$k_{38}$ / $k_{39}$-------/-- --- $k_{64}$ /

$K_D$=/$k_3$,$k_4$,$k_5$,$k_6$,$k_7$ / $k_{11}$,$k_{12}$,$k_{13}$,$k_{14}$,$k_{15}$ / $k_{19}$,$k_{20}$,$k_{21}$,$k_{22}$,$k_{23}$ / $k_{27}$,$k_{28}$,$k_{29}$,$k_{30}$,$k_{31}$ / $k_{35}$,$k_{36}$,$k_{37}$,$k_{38}$,$k_{39}$ / ----$k_{64}$ /

Step 3 : Then the 5 bit binary value will be converted in to decimal value

Step 4 : Based on the decimal value the character will be assigned from the lookup table shown in Table 2.Finally get the original Text without any loss is shown in Fig. 6.

## VI. SIMULATION RESULT

The Hardware model is simulated using a Xilinx simulator. Output is verified by using random input samples. The compression ratio also calculated by using equation (25). In this section the performance of four different Lossless compression techniques like Run length encoding (RLE), Huffman coding, Arithmetic coding and Simple byte compression (SBC) algorithms and multi coding techniques are analyzed and tested with eight different text files. Each has different file sizes and contents. The input file sizes are 256 bytes, 240 bytes, 592 bytes,, 640 bytes, 816 bytes, 232 bytes, 192 bytes, 1232 bytes. To evaluate the compression and coding effectiveness Bits per character, transition activity and power dissipation are analyzed and listed below from Table 3 to 5 and Fig. 7 to 10 for eight different text files.

K – Input data

$K_C$ – Compressed data

Compression ratio ($C_R$) = Bits in K / Bits in $K_c$ (25)

$C_R$ = 64 bits / 40 bits X 100 (26)

$C_R$ = 62.5

The overall performance is analyzed in terms of average bits per character (BPC) for four different lossless compression

TABLE.3. ANALYSIS OF BPC FOR DIFFERENT COMPRESSION METHODS

| File Name | Original Size | RLE | Huffman | Arithmetic | SBC |
|---|---|---|---|---|---|
| | | BPC | BPC | BPC | BPC |
| Text 1 | 256 | 7.00 | 5.60 | 6.31 | 5.00 |
| Text 2 | 240 | 6.40 | 5.93 | 5.65 | 5.03 |
| Text 3 | 640 | 6.8 | 5.24 | 6.19 | 5.00 |
| Text 4 | 816 | 15.56 | 6.23 | 5.83 | 5.01 |
| Text 5 | 232 | 6.06 | 5.76 | 5.49 | 4.96 |
| Text 6 | 192 | 4.66 | 6.45 | 5.97 | 5.00 |
| Text 7 | 592 | 6.05 | 5.82 | 5.36 | 5.08 |
| Text 8 | 1232 | 13.29 | 6.07 | 6.79 | 5.01 |
| **Average BPC** | | **8.227** | **5.887** | **5.948** | **5.011** |

 ISSN: 2454-4248
137 – 142

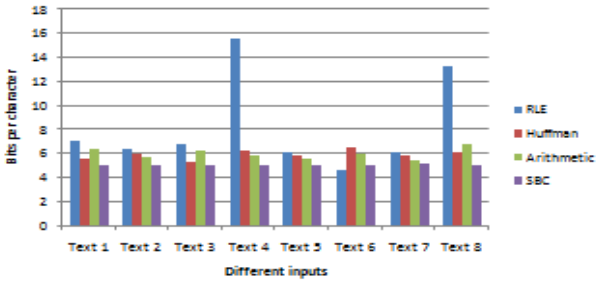


Fig. 7 .Analysis of BPC for different compression methods

TABLE.4: PARAMETER ANALYSIS FOR WITH AND WITHOUT COMPRESSION

| SL. No | Coding methods | % Transition Activity | Power dissipation (mw) |
|---|---|---|---|
| 1 | MC method | 52.3 | 0.624 |
| 2 | MC method with Compression | 58.26 | 0.431 |

TABLE.5: TRANSITION ACTIVITY FOR VARIOUS CODING TECHNIQUE

| SL. No | Coding methods | % Transition Activity |
|---|---|---|
| 1 | BI method | 27.34 |
| 2 | Shift inv | 31.4 |
| 3 | Rotate | 48.89 |
| 4 | NDSMBC | 21 |
| 5 | NBCMEI | 26 |
| 6 | MC method | 52.3 |
| 7 | MC method with Compression | 58.26 |

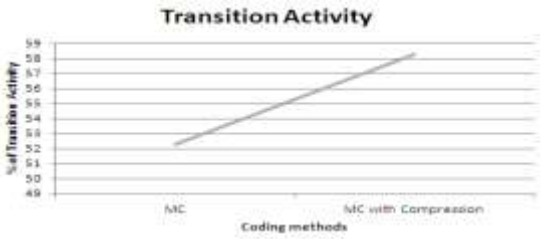


Fig.9. Power dissipation for Multi coding Technique with and without Compression

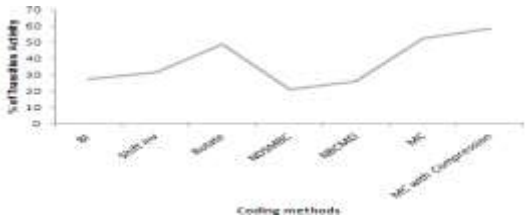


Fig.10. Transition Activity for various coding Techniques

techniques are shown in Table 5 & Fig.8. All text files contain English language of different sizes. Here dictionary compression technique is not considered here, because for large file size, larger dictionary is needed for compression and decompression process, also lots of computer resources required to process and overflow problem is also occurred. From Table 5 except Text file 6 all text files produced larger BPC for RLE technique. Mostly the RLE algorithm generates large compressed file than the original input file. This happens if the number of runs is less in the text file. Run length encoding functions are well when the total number of run is larger in the text files. The average BPC obtained is 8.227 which are higher compared to Huffman, arithmetic and SBC. The average BPC for Huffman coding is 5.887 which is greater than SBC and lesser than RLE and arithmetic. The average BPC for arithmetic coding is 5.948 which is lesser than RLE and larger than Huffman and SBC. Due to underflow problem the result given by arithmetic is not accurate. The average BPC for SBC is 5.011 which are less compared to all lossless compression techniques considered here. It is a best suitable method for larger and smaller files. There is no loss of characters during compression and decompression process, less computational complexity.

## VII. CONCLUSION

A simple byte compression algorithm with multi coding technique has been proposed to reduce the Transition Activity and power dissipation for random text samples. By using this technique compression ratio up to 62.5 % can be achieved, so that the encoder complexity can be reduced. With multi coding technique the compressed data are effectively coded in various coding principles and has improved transition activity up to 58.26% compared to other existing methods. The results are analyzed and verified using Xilinx.

## REFERENCES

[1] Ajay Joshi,Gerald G.Lopez , Jeffrey A.Davis. “Design and optimization Of on-chip interconnects using wave pipe lined multiplexed routing. VLSI systems”, IEEE Transaction on ,2007;.15, 9 : 990-1002.

[2] M.T.Bohr “Interconnect scaling – The real limiter to high performance ULSI”. Proceedings of IEEE international .Electron devices meeting, , 1995: pp. 241-244.

[3] G.Nagendra Babu, Deepika Agarwal, B.K.Kaushik., S.K.Manhas “Bus Encoder for crosstalk avoidance in RLC modeled interconnects”. International journal of VLSI design & Communication systems, 2012;.3, 2.

[4] Arti Joshi, Rishika Sethi, Purushottam Kumawat. “Review of Electromagnetic failure, optimization techniques and stress prediction in Interconnect”. IOSR Journal of VLSI and Signal Processing, 2014; 4, 2 : 23-28.

[5] Jorg Henkel , Haris Lekatsas . “Adaptive Address bus coding for low power deep sub-micron designs”, DAC ,2001.

[6] M.R.Stan , W.P.Burleson. “Bus-Invert coding for low- power I/O”. IEEE Transaction on VLSI, 1995;. 3 : 49-58.

[7] Naveen K. Samala, Damu Radhakrishnan , Baback Izadi.” A Novel Deep Sub – micron Bus Coding for Low Energy”. Proceedings of the international conference on Embedded systems and applications ,ESA , 2004.

[8] A.Sathish , T.Subba Rao. “Bus regrouping method to optimize power in DSM Technology”. Proceedings of IEEE international conference on signal processing communication and networking, 2008, PP 432-436.

 ISSN: 2454-4248
137 – 142

[9] P.P.Sotiriadis, A.Chandrakasan. "Low power bus coding techniques considering inter-wire capacitances". Proceedings of IEEE Custom Integrated Circuits Conference,2000, pp. 507-510.

[10] Lin Xie , Peiliang Qiu , Qinru Qiu. "Partitioned Bus Coding for Energy Reduction". Asia and south pacific Design Automation conference – ASP-DAC,2005, pp.1280-1283.

[11] K.S.Sainarayana, J.V.R.Ravindra, Kiran .T. Nath , M.B.Srinivas." Coding for Minimizing Energy in VLSI Interconnects". The 18th International Conference on Microelectronics (ICM) 2006, pp. 166-169.

[12] Zirra peter Buba , Gregory maksha wajiga. "Radical data compression algorithm using factorization". International journal of computer science and security,2011;5,2:221-226.

[13] Jayapreetha Natesan , Damu Radhakrishnan. "Shift Invert Coding (SINV) for Low Power VLSI". Proceedings of the EUROMICRO Systems on Digital System Design ,2004; 190-194.

[14] MI.Shankaranarayana Bhat , D.Yogitha Jahnavi. "Universal Rotate Invert Bus Encoding for Low Power VLSI". International Journal of VLSI design & Communication Systems (VLSICS) , 2012;.3, 4.

[15] Youngsoo Shin, Soo-Ik Chae, Kiyoung Choi. "Partial Bus-Invert Coding for Power optimization of Application-Specific Systems". IEEE Transaction on VLSI, 2001; 9:. 377-383.

[16] Massimo Alioto, Gaetano Palumbo , Massimo Poli. "Energy Consumption in RC Tree Circuits". IEEE Transaction on Very Large Scale Integration (VLSI) Systems,2006;14, 5: 452-460.

[17] S.Ramprasad , N.R.Shanbhag. J.N. Haji . "Information theoretic bounds on average signal Transition activity". IEEE Transaction on VLSI, 1999 ; 1-10.

[18] Anantha P. Chandrakasan , Robert W. Brodersen. "Minimizing Power Consumption in CMOS Circuits".Department of EECS University of California at Berkeley, pp.1-10.

[19] Aravind Kumar , Anil Kumar Goyal. "Study of various Full Adders using Tanner EDA Tool". International journal of Computer Science And Technology, 2012;.3:.581-584.

[20] Yan Zhang, John Lach, Kevin skadran , Mircea R.Stan." Odd / Even bus invert with Two-phase transfer for buses with coupling". ISLPED, 2002; pp. 80-83.

[21] Anchula sathish, .M.Madhavi latha, K.Lalkishore." High performance data bus encoding technique in DSM Technology". ACEEE International journal .on communications, 2011;3: 486-489.