Microelectronics and Solid State Electronics 2013, 2(3): 45-51
DOI: 10.5923/j.msse.20130203.02

Low Power Bus Encoding Techniques for Memory Testing

Balwinder Singh[1,*], Arun Khosla[2], S. Bindra. Narang[3]

[1]Centre for Development of Advanced Computing (CDAC), Mohali, India
[2]ECE Department Dr. B .R. Ambedkar National Institute of Technology, Jalandhar , India
[3]Electronics Technology Department, Guru Nanak Dev University, Amritsar, India

Abstract Power reduction during testing is one of the most important challenges for the VLSI design and test engineers. Switching is one of the important contributors to the power consumption of on chip buses. This paper addresses the problem of minimizing power dissipation due to switching of the busses during testing. For memory testing, many read/write cycles are required (as per March algorithm), in which test patterns are sent through the data bus, according to the addressing sequences. The address switching is also controlled by address encoding schemes with reference data. The result shows the average power reduction of 30% and 74.8% in data bus and addresses buses encoding schemes respectively.

Keywords Low Power Encoding Scheme, T0-C code, Bus inverting Automatic Test Pattern Generator

1. Introduction

Power consumption has become one of the most critical issues during the design of battery operated portable systems. Design methodologies and related tools enabling the control on the power budget during the different phases of the design flow are mandatory. The main components of such portable systems are processing unit, the memory controller, memory chips. It is anticipated that the embedded memory will account for 95% of the chip area in the complex systems[1] and the communication channels which provide data transfer between the CPU and the memory. During the communicati on of these components bus power consumption is proportional to the switching activity. The switching activity of the buses is often responsible for a substantial fraction of the total power dissipation for large VLSI modules. As the compelxity of the system increase the number of data buses and address buses incresses. The large and long interconnects dissipates as dynamic energy dissipation for charging and discharging of internal node capacitances and inter-wire capacitances.

Testing of the memory sections is a tedious job because in memory testing the bunch of the data is written and read again and again for all the memory locations at very high speed. During memory testing power dissipation increase drastically due to excessive switching in test patterns. Therefore, it needs low power encoding techniques for communication between devices or on chip communication. To access a Memory Under Test (MUT) address buses are required to access memory locations and data buses for sending test patterns. Thus, the minimization of the transitions can provide significant savings on the overall power budget. To accomplish this job there is a need of very fast processors which are connected in parallel to provide the very high testing rate. SCAN based design are mostly preferred for memory testing[1]. During writing and scanning first, write 0s to the memory in ascending order of address. Second, read the memory (expect to read a 0) and then write a 1 in ascending order of address and third is to read and write a 0 in descending order of address. Finally Fourth, read the memory (expect to read a 0). However testing time of a scan system is proportional to both the number of test patterns and the number of flip-flops in the scan chain(s). In a modern VLSI circuit, the numbers of internal flip-flops are in the range of thousands or even higher, therefore, to reduce test application time and large amount power consumption become an important issue, that needs to be optimized along with speed and area. Power consumption is directly related to the number of times flip-flops toggle their states i.e. toggling rate, so intention is given to arrange the test vectors in such a manner that minimum toggling occur in a Memory Under Test (MUT). Thus, the minimization of the transitions can provide significant savings on the overall power budget. The relation between power and switching rate is related according to equation 1.

$$p = \left(\frac{V_{dd}{}^2}{2*clock\ period}\right) * \left(\sum_{for\ all\ gates\ g}^{n} \lfloor toggle(g) * C(g) \right. \quad (1)$$

Where V_{dd} represents biasing voltage, C(g) is the output capacitance of the nodes and toggle(g) is switching of the gates 'g'.

VLSI circuits with large power dissipation require

* Corresponding author:
balwinder.cdacmohali@gmail.com (Balwinder Singh)
Published online at http://journal.sapub.org/msse

expensive packaging to ensure appropriate heat dissipation. Excessive power dissipation results the overheating of the MUT, and probability of damage are increased at testing time. Thus most research efforts have focused on reducing the dynamic power consumption during memory writes and read cycle by reducing the transitions.

The rest of the paper is organized as follows: In the next section, we reviewed the related work on the bus encoding schemes for low power. In section 3, the Bit Invert, Bit invert signaling, ABIT and Proposed Scheme for power reduction during the data are explained. Then the encoder and decoder are designed for these schemes and implemented on FPGA. In section 4, the address encoding schemes are discussed and implemented for the memory addressing. In section 5, we have compared these schemes on the basis of power, Area & energy efficiency for the reference data. In the last section conclusion is drawn.

2. Related Work

Luca benini et. al.[3] presents the low power address bus encoding technique to sort out the problem of power dissipation for intellectual proprietary core processor. Here reduction in bus activity is achieved with this technique. Yazdan Aghaghiri et. al.[18][4] in this T0 and Offset- XOR encoding techniques are discussed and irredundant bus encoding for low power is presented and its benefit over the other techniques are presented, and it is shown that this method of irredundant saves 83% of power. Prabhat K. Saraswat et. al.[13] uses gray code and T0 code for memory address bus encoding. Bits encoded n such way the bit switching activity is reduced on the address busses. The sequences of addresses are generated of specified sequentialty and evaluated the performance of both codec's.

Sathish et.al[10] suggested a Efficient Switching Activity Reduction Technique for Fault Tolerant Data Bus for Deep-submicron (DSM) systems. The proposed encoding technique reduces the switching activity by 18% to 22.5%. Its efficiency is 8% to 15% more compare to others encoding techniques. Youngsoo et.al[8] presents a partial bus-invert coding scheme for power optimization of system level bus. In the proposed scheme, they select the sub-group of bus lines for bus encoding to reduce the total number of bus transitions. It reduces the total bus transitions by 62.6% on the average, compared to that of the unencoded patterns.

Ghoneima et. al[22] proposed a low power on chip the serial-link bus (SLB) architecture which reduces the number of bus lines of the conventional parallel-line bus (PLB) architecture by multiplexing each m-bit onto a single line. In another paper , Sinha et.al[23] authors have proposed a bus encoding technique using Forbidden Transition Free Algorithm for Cross-Talk Reduction for On-chip VLSI Interconnect.

3. Data Bus Encoding Technique for Memory Testing

Bus-Invert Method: This technique is based on Hamming Distance between the consecutive test vectors. Test patterns generated by the Automatic Test Pattern Generation (ATPG) are encoded according to their hamming distance before it sends to Circuit Under Test (CUT). The encoding depends on the Hamming distance between the value of the encoded bus lines at time t-1 *(*also counting the redundant line at time t-1) and the value of the bus lines at time t.

The bus-invert encoding method can be expressed by the following equations[5]]2]:

$$(B(t), INV(t)) \quad (b(t), 0) \quad \text{if } H(t) = N/2$$
$$= \quad (b'(t), 1) \quad \text{if } t > 0. H(t) > N/2$$

Where B (t) is the value of the encoded bus lines at time t,
INV (t) is the additional bus line,
b (t) is the address value at time t,
$H(t) = (B(t-1) \mid INV(t-1), b(t) \mid 0)$ is the Hamming distance, and N is the bus width of b (t).

The corresponding decoding scheme is simply defined as:

$$b(t) = B(t) \quad \text{if } INV = 0$$
$$= B'(t) \quad \text{if } INV = 1$$

Major drawback of this approach are related to the required redundant bus line and the overhead due to the logic to implement the voter to decide whether the Hamming distance exceeds N / 2.

Bus Invert Transition Signaling: With transition signaling a logical 1 is represented by a *transition* (from HI to LO or LO to HI) while a 0 is represented by the lack of such a transition. Transition signaling will not reduce switching activity by itself[5]. If transition signaling is used and at the same time we reduce the number of 1's in the code words we can directly reduce the switching activity on the bus.

Transition signaling combined with bus-invert (BI) coding, is called BITS coding. BITS reduce the number of logical 1s transmitted over the bus,[4] that are responsible for all the high-to-low and low-to high transitions, while the logical *0s* are represented by the lack of transitions.

In BITS encoding, if the number of ones in the incoming data is larger than half the bus width, then each bit of incoming data is inverted and INV is set to one (1) and then transition-encoded. Otherwise, each bit of incoming data is transition-encoded without alteration. More precisely, for a pattern at time, (Xi), its BI-encoded version is given by:

$$Yi \mid I = Xi \mid 0 \text{ ; if } w(Xi) <= n/2$$
$$= \text{not } (Xi) \mid 1 \text{ ; otherwise}$$

where | denotes a concatenate operation, I denote the value at the invert line, w (Xi) denotes the number of 1's in Xi, and n denotes the bus width.

Then, the BITS-encoded version of *Xi* is given by

$$Zi \mid I = TS(Yi; Z i\_1) \mid I;$$

where TS(x; y) denotes a transition encoding of x with respect to y.

The decoding process can be carried out by

$$Xi = Yi ; \quad \text{if } I = 0$$
$$= \text{not } Yi; \text{ otherwise}$$

where $Yi = TS-1(Zi;Zi\_1)$.

Note that both TS(x; y) and its inverse TS-1 (x;y) can be implemented by XORing x and y

In **Approximate BITS encoding**[8] technique, numbers of transitions are reduced as compared to the simple and BI (bit inverse) method. Due to which the total power of the circuit is reduced. Its application to real circuit design is limited by the extra bus line, which calls for change in pin out and interface specification of the original chip. As the main overhead comes from a majority vote, that decides whether to encode 1 or 0 as having a transition, we eliminate the voter by moving to another technique which is ABITS. In ABITS method, the guess relies on the MSB of each input bus. This means that MSB takes over the function of invert line thus eliminating the need for an extra bus line. In ABITS method, the output function Y is the XORing or XNORing depending upon the MSB of input data.

$$Yi = \begin{cases} x_i^{n-1} | X_i(n-1), \\ x_i^{n-1} | \overline{X_i(n-1)}, \end{cases} \quad \text{if } x_i^{n-1} = 0, \text{otherwise}$$

$$Zi = y_i^{n-1} | TS(Yi(n-1), Z_{i-1}(n-1)$$

Although ABITS encoding obtains less transition reduction due to incorrect guess, the overall power consumption (including the power consumed by the encoder itself) is fairly comparable to that of BITS encoding because the ABITS encoder consumes less power than BITS encoder.

Proposed Technique for Encoding: The proposed technique has an edge over other existing techniques that it has no redundant line. Due to which an extra expensive pin is saved. This technique works for all data streams except for the stream containing all 1's.

Encoder

In this technique the encoder is used for encoding the transitions. It takes 8-bit input word as input bit stream and functions as:

$$Yi = \begin{cases} xi(n-1) \oplus xi(n) & \text{if } N_{11} < \frac{1}{2}BW \\ xi(n-1)\,! \oplus xi(n) & \text{if } N_{11} > \frac{1}{2}BW \\ xi(n-1) \oplus (!\,xi(n)) & \text{otherwise} \end{cases}.$$

Decoder

The decoder is used for decoding the output generated from the encoder. It takes transmitted word as input bit stream and then computes the original input word X(n). It functions as:

$$Xi = \begin{cases} yi(n-1) \oplus yi(n) & \text{if } N_{11} < \frac{1}{2}BW \\ yi(n-1)\,! \oplus yi(n) & \text{if } N_{11} > \frac{1}{2}BW \\ yi(n-1) \oplus (!\,yi(n)) & \text{otherwise} \end{cases}.$$

Encoding function should minimize the average number of ones at its output while the encoded value Y(n) can still be uniquely decoded by D.

4. Address Bus Encoding Technique for Memory Testing

In memory testing read /writes operations are performed according to the algorithms such as March. One address generator is also required for the addressing the memory locations. In this switching also takes place, causes the power dissipation. This switching can be reducing with various address encoding techniques. To encode the address bus for test with least switching activity various methods like gray encoding and T0-C encoding. By utilizing the sequential addressing instead of using random addressing reduces the power dissipation in the address line by reducing the toggling count. But after some clock cycles even the in the sequential addressing the toggling count is increased by the unity value. Due to that here we concentrate on T0, gray code base addressing and irredundant bus encoding. In gray encoding the toggling count is fixed to the unit value, which saves the power dissipation. To even reduce the toggling value beyond the unit value the technique preferred is the T0-C encoding. To save more power, instead of using gray coding T0-C encoding is used. In case of simple T0 encoding the process of sending address bit is shown in table1, below:

Table 1. T0 encoding

b(t)	B(t)
39	39
40	39
41	39
39	39

As when we reach the last row of the table, no valid code word can be generated for source word 39. If we use 39 as the code word, the receiver (decoder) cannot determine whether the source word was 39 (backward jump) or 42 (next sequential address). So the error appears when the data on the bus is equal to the branch target itself. That is why spatial redundancy was originally introduced into the T0 code.

In T0-C when such a case occurs, we set the code word to b(t-1) + S. The reason is that this is the only pattern that the receiver should not expect from the sender. Notice that when the receiver sees a value of b(t-1) + S, it knows that the sequential addressing has been stopped because the bus value has changed. On the other hand, when it computes the new jump address, it recognizes that this jump address is the same as the next sequential address. Therefore, if in fact a special case were not encountered, there would be no need for the sender to unfreeze the bus value. This special case is, of course, when the target of the backward jump is the same as the current value on the bus. The decoder is aware of this, and ambiguity is resolved. T0-C encoder works as follows:

if {b(t) == b(t-1) + S}
B(t) = B(t-1)
else if {B(t-1) != b(t)}
B(t) = b(t)
else
B(t) = b(t-1) + S

On the receiver side, when the b(t-1) + S value is received, the previous value on the bus is regarded as the branch target. For the previous example we will have:

Table 2. T0 –c encoding

b(t)	B(t)
39	39
40	39
41	39
39	42
40	42

Gray code address generator

The term Gray code or reflected binary code is referred to a binary sequence in which only a single bit changes value when transition between adjacent states occurs. A Gray code can be generated by changing the least significant bit that results in a new state. The next sequence creates a mirror image of the existing gray code below the original value. The binary values can be converted to gray codes by XORing the bits. Let us consider a 4-bit binary value B and 4-bit gray output G. B can be converted to G as:

G (3) =B (3);

G (2) =B (2) $\oplus$. (3);

G (1) =B (2) $\oplus$. B1);

G (0) =B (1) $\oplus$. (0);

The gray code counters are popular in digital system for a variety of applications, such as representing the state variables in state machines or acting as pointers in First-In First-Out (FIFO) memories, and also can be used as address counters. This is because in gray code counters only one output bit is ever toggling at a time in as opposed to multiple bits in a binary counter. These counters count the values in gray codes only. The advantage of Gray code counters over others is that they consume only half the power of an equivalent binary counter and correspondingly less noise.

5. Bus Power Model for Encoding Scheme

Let us assume, power P watt will be dissipate when switching takes place in any of the bus line. System bus switching can be calculate for n bus lines then the power dissipation is n * P watt and so to calculate the switching activity[15]. The power dissipation on the bus is calculated by applying all the encoding techniques on the reference data as taken in Table 5.

Power dissipation is given by:

$$P_{sw} = \alpha C_L V^2_{dd} f \quad (2)$$

α = Switching activity

V_{dd} =Supply voltage

f = Frequency

C_L= Load Capacitance

The encoding efficiency E is an important parameter for the comparison of various encoding techniques as given by equation[Reference], which is technology independent and gives only reduction in switching activity.

$$E = 1 - \alpha_{coded} / \alpha_{uncoded} \quad (3)$$

in order to determine the Power savings in overall power consumption, the self dissipation of encoder and decoder circuits has to be considered as well. The power savings are given by equation:

$$P_{saved} = P_{uncoded} - (P_{Coded} + P_{encoder} + P_{decoder} \quad (4)$$

$P_{uncoded}$ = power consumed for un-coded data streams

P_{coded} = power consumed for coded data streams

$P_{encoder}$ = power consumed by the Encoder

$P_{decoder}$ = power consumed by the Encoder

The percentage of bus power saved due to coding is given by the energy efficiency EP in equation

$$E_p = \frac{P_{saved}}{P_{uncoded}} \quad (5)$$

Energy E_p >0 then power is saved. For E_P<0 the total power consumption is increased by this encoding technique. To save Power, the Encoder and decoder has to be small and the power consumption of the encoded bus has to be sufficiently lower than the original un-coded bus.

6. Experimental Results

In this section, we present the details of our experimentation setup and discussion on the results obtained with various encoding techniques:

The I/O voltage was assumed to be 3.3v, Frequency of 10 MHz, and switching activity is estimated for the reference data taken in table no 6. The encoders and decoders are designed for the encoding schemes in HDL with the NC launch cadence tool. The power consumption and cell area is calculated for these schemes are shown in table 3. The number of transitions is calculated with transition counter for the reference data of table.

Table 3. Power calculation for data encoding scheme

Parameters		Un-coded	BI	BITS	ABIT	Proposed
Cell Area		--	1043	1387	1378	1792
e		--	1	1	0	0
PPower	Encoder	--	72.4	93.85	18.14	41.22
	Decoder	--	31.4	33.81	27.22	26.74
	Bus Power	8027	6665	6257	7347.5	5576
	Total	8027	6768.8	6384.6	7393	5643.9
Transitions		118	98	92	108	80
Encoding Eff.		--	17	23	09	33
Power saved		--	1258.2	1642.4	634.28	2383.1
Energy Eff.		--	15.6	20.4	7	29.6

Table 4. Power calculation for Address encoding schemes

Parameters		Binary	Gray code	T0
Cell Area		534	129	874
Power	Leakage Power (mw)	2.12	0.4215	2.233
	Dynamic Power(mw)	7.22	9.175	5.5469
	Bus Power	2178	1088	544
	Total	2187.34	1097.21	551.79
Transitions		14	8	4
Encoding Efficiency		--	42.58	71.42
Power saved		--	1090.13	1636.8
Energy Efficiency		--	49.83	74.79

Table 5. Reference address locations & Hamming Distance Calculation of Various encoding scheme

Binary	Transitions		T0 coding	Transitions		Gray coding	Transitions	
00000100		0	00000100		0	00000110	0	
00000101	1	136	00000100	1	136	00000111	1	136
00000110	2	272	00000100	0	0	00000101	1	136
00000111	1	136	00000100	0	0	00000100	1	136
00001000	4	545	00000100	0	0	00001100	1	136
00000110	3	408	00000110	2	272	00000101	2	272
00000111	1	136	00000110	1	136	00000100	1	136
00001000	4	545	00000110	0	0	00001100	1	136
Total	14	2178	Total	4	544	Total	8	1088

Table 6. Reference Test Patterns & Hamming Distance Calculation of Various encoding scheme

Bit pattern used	Simple HAMMING DISTANCE		HAMMING DISTANCE for Bit inverse algorithm		HAMMING DISTANCE for BITS algorithm		HAMMING DISTANCE for ABITS algorithm		HAMMING DISTANCE for Proposed technique	
	Transitions	Power (mw)	Transitions	Power (mw)	Transitions	Power (mw)	Transitions	Power (mw)	Transitions	Power (mw)
0000_0000_0000_0000	0	0	0	0	0	0	0	0	0	0
0000_0010_0000_0010	2	136	2	136	2	136	2	136	2	136
0000_1111_0000_1111	6	408	6	408	6	408	8	544.5	8	544.5
0000_0001_0000_0001	6	408	6	408	6	408	2	136	2	136
0001_0101_0001_0101	4	272	4	272	4	272	6	408	6	408
1001_0100_1001_0100	4	272	4	272	4	272	8	544.5	6	408
1001_0010_1001_0010	4	272	4	272	4	272	10	680.6	6	408
1010_0000_1010_0000	6	408	6	408	6	408	9	612.56	4	272
1110_1011_1110_1011	8	544.5	8	544.5	8	544.5	4	272	4	272
0011_0100_0011_0100	14	952.8	2	136	4	272	7	476	6	408
0111_0111_0111_0111	6	408	6	408	4	272	8	544.5	4	272
1111_0110_1111_0110	4	272	4	272	4	272	5	340	4	272
0011_0000_0011_0000	8	544.5	8	544.5	6	408	5	340	4	272
0010_1000_0010_1000	4	272	4	272	4	272	4	272	4	272
1110_0101_1110_0101	10	680.6	6	408	6	408	7	476	6	408
1110_1011_1110_1011	6	408	6	408	4	272	4	272	4	272
0010_0000_0010_0000	10	680.6	6	408	4	272	3	204	2	136
1010_0001_1010_0001	4	272	4	272	4	272	11	748.69	6	408
0000_1000_0000_1000	8	544.5	8	544.5	8	544.5	3	204	2	136
0000_0100_0000_0100	4	272	4	272	4	272	2	136	2	136
Total	118	8027.5	98	6665.5	92	6257	118	7347.35	80	5576.5

As the results shows that proposed scheme number of transitions are reduced by 33% from un-coded data and 29.6% power is saved whereas in Bit Inverse and Bit inverse transition signaling techniques reduces power only 15.6 and 20.4 respectively and also uses an extra bit line for inverse logic indication. ABIT save this extra line but power saving is less than these schemes. The proposed scheme saves more power without the extra line overhead.

In memory testing read /writes operations are perform according to the algorithms such as March. One address generator is also required for addressing the memory locations. In this switching is also take place, which causes the power dissipation. This switching can be reduced with various address encoding techniques. The power consumption and cell area is calculated for address encoding schemes for reference address locations are shown in table 4.

7. Conclusions

The low power bus encoding techniques play an important role in reducing the switching activity during the on chip communication. Testing of the memory is very important because 90% of the chip area is occupied by memory. During memory testing many read /writes operations needs to be test according to the March algorithms) to achieve 100% fault coverage. In this paper we have implemented bus encoding schemes for memory BIST. The encoding schemes are applied to the data bus and address bus by taking the reference data. The power consumption of various encoding schemes is shown in figure 1. It is observed from the results of proposed scheme for data encoding is 33% less switching activity occurs and about 30% of average power is saved. In case of address encoding that 71.2 % less switching activity occurs and about 74.8 % of average power is saved is shown in figure 2

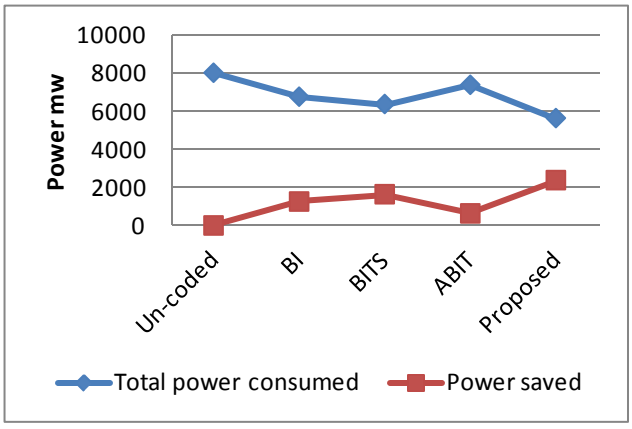

Figure 1. Power consumption of Various Data Encoding Schemes

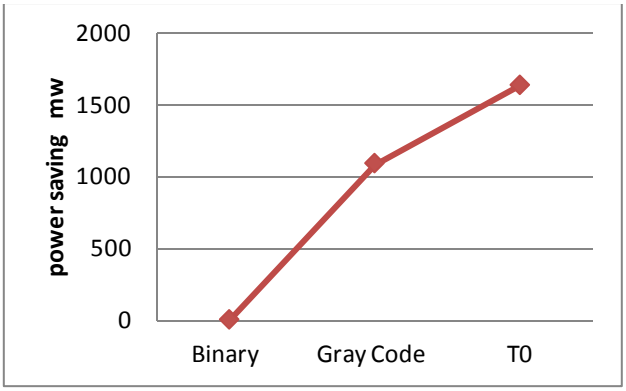

Figure 2. Power consumption of Various Address Encoding Schemes

REFERENCES

[1] International Technology Roadmap for Semiconductors (ITRS)," 2007.[Online]. Available: http://public.itrs.net

[2] Kuen-Jong Lee; Jih-Jeen Chen; Cheng-Hua Huang; , "Broadcasting test patterns to multiple circuits," Computer - Aided Design of Integrated Circuits and Systems, IEEE Transactions on , vol.18, no.12, pp.1793-1802, Dec 1999

[3] Luca Benini, Alberto Macii, Enrico Macii, "Architectures and Synthesis Algorithms for Power-Efficient Bus Interfaces", IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 19, NO. 9,pp.969-979, September 2000

[4] Yazdan Aghaghiri, Farzan Fallah, Massoud Pedram, "Irredundant Address Bus Encoding for Low Power", '01, August 6-7, 2001, Huntington Beach, CA.

[5] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 3, pp. 49–58, Mar. 1995

[6] Luca Benini, , Enrico Macii,, G.D. Macii,D.Scuito Asymptotic Zero-Transition Activity Encoding for Address Busses in Low-Power Microprocessor-Based Systems, IEEE/ ACM Great Lakes Symposium on VLSI, pp. 77–82, March 1997.

[7] Y.Shin, S.I.Chae and K.Choi, "Partial Bus-Invert Coding for Power Optimization of Application-Specific Systems", IEEE Trans. On VLSI Systems, April 2001, vol. 9, pp377-383.

[8] Youngsoo Shin; Kiyoung Choi , "Narrow bus encoding for low power systems", Design Automation Conference, 2000. Proceedings of the ASP-DAC 2000. Asia and South Pacific , 2000 , Page(s): 217 -220

[9] Sangkwon Na, Sung Yang, and Chong-Min Kyung Low - Power Bus Architecture Composition for AMBA AXI Journal of Semiconductor Technology and Science June. 2009, Vol.9, No.2 pp 75-79

[10] A. Sathish ,M. Madhavi Latha, K. Lal kishore Efficient Switching Activity Reduction Technique for Fault Tolerant Data Bus International Journal of Computer Applications Volume 36– No.12, December 2011 pp 7-11

[11] P. K. Saraswat, G. Haghani, and A. K. Bernard, "A low power design of gray and T0 codecs for the address bus encoding for system level power optimization," 2005

[12] Rao, R.R.; Deogun, H.S.; Blaauw, D.; Sylvester, D.; , "Bus encoding for total power reduction using a leakage-aware buffer configuration," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.13, no.12, pp.1376-1383, Dec. 2005

[13] J. Gu and H. Gui, "An Efficient Segmental Bus-Invert Coding Method for Instruction Memory Data Bus Switching Reduction," EURASIP Journal on Embedded Systems, vol.2009, article ID 973976, 2009.Pages 1-10

[14] Ramprasad, S.; Shanbhag, N.R.; Hajj, I.N. , "A coding framework for Low-power address and data busses ", Very Large Scale Integration (VLSI) Systems, IEEE Transactions on Volume: 7 2, June 1999, Page(s): 212 -221

[15] M. Nourani, M. Tehranipoor and N. Ahmed, "Low-Transition Test Pattern Generation for BIST-Based Applications," IEEE Transactions on Computers, vol. 57, no. 3, pp. 303-315, March 2008.

[16] Ioannis Voyiatzis, "Embedding test patterns into Low-Power BIST sequences," 13th IEEE International On-Line Testing Symposium IOLTS 2007.

[17] Y. Aghaghiri, F. Fallah, M. Pedram, "ALBORZ: Address Level Bus Power Optimization," International Symposium on Quality Electronic Design (ISQED), California, Mar. 2002. Also in International Workshop on Logic and Synthesis (IWLS), California, June 2001.

[18] Y. Aghaghiri, F. Fallah, M. Pedram, "EZ: A Class of Irredundant Low Power Codes for Data-Address and Multiplexed Address Buses," Design Automation and Test in Europe (DATE), France, Mar. 2002.

[19] Rochit, R.: System-on-a-Chip: Design and Test, pp. 155–177. Artech house. Inc.,Norwood (2000)

[20] Khan, Z.; Arslan, T.; Erdogan, A.T.; "Low power system on chip bus encoding scheme with crosstalk noise reduction capability," *Computers and Digital Techniques, IEE Proceedings* - , vol.153, no.2, pp. 101- 108, 6 March 2006

[21] Hadi Parandeh-Afshar, Mohsen Saneei, Ali Afzali-Kusha, Massoud Pedram: Fast INC-XOR codec for low-power address buses. IET Computers & Digital Techniques 1(5): 625-626 (2007)

[22] Ghoneima, M., Ismail, Y., Khellah, M.M., Tschanz, J. and De, V.(2009). Serial-Link Bus: A Low-Power On-Chip Bus Architecture. IEEE Transactions on Circuits and Systems I. 56(9):2020-2032

[23] Sinha, S., Kar, R. and Bhattacharjee, A.K. Bus Encoding Technique Using Forbidden Transition Free Algorithm for Cross-Talk Reduction for On-chip VLSI Interconnect. Proceedings of International Conference on Advances in Computer Engineering (ACE).pp.256-258. 2010.

[24] Rao, J.V.; Rao, P.S., "Design and implementation of efficient CODECs for on-chip buses to reduce both crosstalk delay and power dissipation," *Computing and Communication Systems (NCCCS),* pp.1,5, 21-22 Nov. 2012