VOL. 10, NO. 11, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

EMBEDDED SHIFT INVERT TRANSITION CODING FOR PARALLEL LINKS

B. Sivachandra, R. Ganesh, B. Parthasarathy and M. Dhinesh
Department of Electronics and Communication Engineering, Aarupadai Veedu Institute of Technology, Chennai, India
E-Mail: sivabala_82@yahoo.com

ABSTRACT

Power dissipation in a chip mostly depends only on switching activity in a chip (high to low vice versa).Various gating and Encoding schemes are introduced to reduce the unnecessary switching activity of a chip during serial data transmission in a network. But the Encoding methods results in increased bit size and reduces only 15% of the power dissipation (loss). This paper proposed to reduce the switching transitions in a wider Bus, Embedded Shift Invert Coding technique is designed and simulated and its performance is compared with the existing Bus Encoding Techniques. The proposed Embedded Shift Invert Coding technique is first designed and simulated by partitioning the Bus data into two equal width data partitions and applying conventional Shift Invert Coding technique to each of the data partitions. The analysis and simulation results indicate that the Proposed Coding scheme produces a low bit transition for different kinds of data patterns (INV, Left Shift, Right Shift and Normal).

Keywords: bus invert coding, coding efficiency, model sim parallel link, shift invert coding.

1. INTRODUCTION

Design of portable consumer electronic devices and other embedded systems are requiring low power consumption to maximize the battery life, reduce weight, size and improve reliability. Generally these power devices are having microprocessors as the processing unit and memories are the storage units. A data Bus is needed for the connection of the processing unit and the storage unit. Therefore the microprocessor will do all the processes on data at every clock pulse and the buses within the processor make the more power consumption due to the data Bus.

Although the power consumption of a system can be reduced at various phases of the design process from system level down to process level, optimization at higher level can provide more power saving. Among the architectural components at the system level, Buses that interconnect subsystems are an important component, which consumes a lot of power. Therefore, a considerable amount of power can be saved by reducing the power consumption of Bus. The major share in the total power dissipation of a Bus is contributed by the dynamic power dissipation. Such dynamic power dissipation in parallel data transmission is due to the charging and discharging of substrate capacitances (load capacitance) and inter-wire capacity (coupling capacitances) contributed by the transition or switching of signals on the Bus. This work mainly focuses on reducing the number of switching transitions in Data Bus by encoding the data that has to be transmitted through the Bus. Bus Invert Coding is a widely popular Bus Encoding technique for reducing the number of switching transitions in Data Bus. Shift Invert Coding technique is another popular Bus Encoding technique which is superior to Bus Invert Coding technique in reducing the number of switching transitions. The main disadvantage of these two Coding techniques is that its Coding efficiency degrades for wider data Buses. So for wider Buses Embedded Bus Invert Coding is used where the Bus data is embedded into two equal width data partitions and the conventional Bus Invert Coding technique is applied to each of the data partitions. Although this Embedded Bus Invert Coding is superior to Bus Invert and Shift Invert coding for wider Buses, here also too much reduction in the number of switching transitions cannot be achieved. Since the Shift Invert Coding technique is superior to Bus Invert Coding technique, so by combining the Shift Invert Coding technique and the concept of Bus data partitioning a large reduction in the number of switching transitions can be achieved for wider data Buses.

2. PROPOSED SYSTEM

The paper is aimed at combining the Shift Invert Coding and the scheme for partitioning the Bus data into several data partitions. Here the proposed Embedded Shift Invert Coding technique is first designed by partitioning the Bus data into two equal width data partitions and the conventional Shift Invert Coding is applied to each of the data partitions. Also the Bus data can be embedded not only into two equal width data partitions but also into any of the partition types. This paper also compares the performance of proposed Coding technique (Embedded Shift Invert Coding) with the conventional Data Busencoding Techniques like Bus Invert Coding, Shift generated data vectors. The performance will be compared by designing the Encoding and Decoding logics for 8 bit, 16bit, 32bit and 64 bit data Buses by applying the four Bus Encoding techniques.

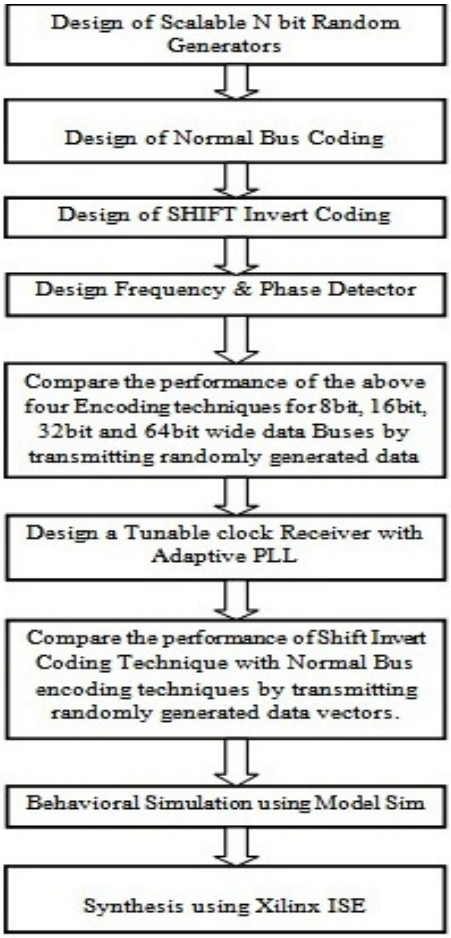

Figure-1. Proposed system flow diagram.

a) Shift invert coding

Shift Invert Coding technique is simple but efficient improvement of Bus Invert Coding technique. Shift Invert Coding technique is based on the principle that not only Inverting the data but Shifting the data **by one bit** position (either left Shifting or right Shifting) can also result in reduced number of transitions . So it is very clear that the Shift Invert Coding technique leads to more reduced number of transitions when compared to Bus Invert Coding technique. The rationale behind using the Shift operation is simple. By Shifting the next data that has to be put on the Bus it is possible that the bit values could match in more places with the existing data on the Bus than if the data bits were either Inverted or left unchanged. The matching of bits in more places implies fewer transitions, thereby giving a better solution. In shifting operation we will perform a circular left Shift or circular right Shift. This will guarantee that we do not lose any information from the original data. It is obvious that Shifting left or right will not always reduce the number of transitions. Depending on the values of current data on the Bus and next data to be send it is possible that either the Inverted next data or maybe even the unmodified/original next data gives the least transitions when sent on the Bus. So by Shift Invert Coding technique the next data that will be send through the Bus will be either the next data value itself or the Inverted next data value or the left Shifted next data value or the right Shifted next data value depending upon which will leads to minimum number of transitions with the current data on the Bus. Since there are four possible values two control bits will be required to indicate the Coding that was used. So if the Bus width is N then by Shift Invert Coding technique the resulting Bus width will be N+2.

b) Algorithm for embedded shift invert coding

Let $D^{(t)}$ be the current data on the Bus at time instant t and $b^{(t)}$ be the next data to be put on the Busat that time instant. Let $b^{INV(t)}$ be the Inverted next data $b^{(t)}$, $b^{LS(t)}$ be the left Shifted next data $b^{(t)}$ and $b^{RS(t)}$ be the right Shifted next data $b^{(t)}$.The number of transitions can be reduced by Coding as follows:

(1) Compute the hamming distance between the current Bus value and next data value. Let it be H.

(2) Compute the hamming distance between the current Bus value and Inverted next data value. Let it be H^{INV}.

(3) Compute the hamming distance between the current Bus value and left Shifted next data value. Let it be H^{LS}.

(4) Compute the hamming distance between the current Bus value and right Shifted next data value. Let it be H^{RS}.

(5) Find out the minimum hamming distance (H_{min}) among H, H^{INV}, H^{LS} and H^{RS}.

(6) The next data that will be put on the Bus ($B^{(t)}$) will be one among $b^{(t)}$, $b^{INV(t)}$, $b^{LS(t)}$ and $b^{RS(t)}$ depending upon the minimum among H, H^{INV}, H^{LS} and H^{RS}.

Since there are four possible Coding schemes for the data in Bus, two control bits will be required to indicate the scheme used at a particular time instant. The bit assignments for the four cases are given in Table-1.

Table-1. Representation of control BRRS.

Coding scheme	**Control bits** (C_1C_0)
Default (no encoding)	00
Invert	01
Left -shift	10
Right-shift	11

The arithmetic expression for the Shift Invert Coding is given in Equation (1).

$$(B^{(t)}, C_1^{(t)} C_0^{(t)}) = \begin{cases} (b^{(t)}, 00) & H_{min} = H \\ (b^{INV(t)}, 01) & H_{min} = H^{INV} \\ (b^{LS(t)}, 10) & H_{min} = H^{LS} \\ (b^{RS(t)}, 11) & H_{min} = H^{RS} \end{cases} \quad (1)$$

Here, $B^{(t)}$ stands for coded Bus value at time index t, $C_1^{(t)} C_0^{(t)}$ stands for the value of control line at time index t. .

3. SYSTEM DESIGN REQUIREMENTS

In this work, the design entry is modeled using VHDL in Xilinx ISE Design Suite 12.1 with the project targeted to Spartan 3E Starter Kit Board, to obtain the synthesis report. The simulation of the design is performed using ModelSim PE 10.1a from Mentor Graphics to validate the functionality of the design.

a) VHDL: The language

VHDL stands for VHSIC (Very High Speed Integrated Circuits) Hardware Description Language.. In the mid-1980's the U.S. Department of Defense and the IEEE sponsored the development of this hardware description language with the goal to develop very high-speed integrated circuit. It has become now one of industry's standard languages used to describe and simulate complex digital systems. A hardware description language is inherently parallel, i.e. commands, which correspond to logic gates, are executed (computed) in parallel, as soon as a new input arrives. VHDL can be used to model a digital system at many levels of abstraction, ranging from the algorithmic level to the gate level. The complexity of digital system being modeled could vary from that of a simple gate to a complete digital electronic system, or anything in between. The digital system can also be described hierarchically. Timing can also be explicitly modeled in the same description.

b) Model Sim PE 10.1a

Model Sim is a powerful HDL simulation tool provided by Mentor Graphics that allows us to simulate the inputs of our modules and view both outputs and internal signals. Model Sim simulator support for both VHDL and Verilog designs. The combination of industry-leading performance and capacity with the best integrated debug and analysis environment make ModelSim the simulator of choice for both ASIC and FPGA design. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows. . ModelSim now offers two series of simulators: ModelSim PE for Block/Small System Simulation which works only in Windows and ModelSim DE for Quality Critical Designs which works both in Windows and Linux.

4. IMPLEMENTATION RESULTS

The performance of proposed Embedded Shift Invert Coding technique (with two equal width data partitions) are compared with the existing Coding techniques for transmitting randomly generated data vectors. First the data vectors are transmitted without any Coding and then transmitted by applying the four Encoding techniques. Total transitions in an 8 bit wide Bus for duration of 1ms.

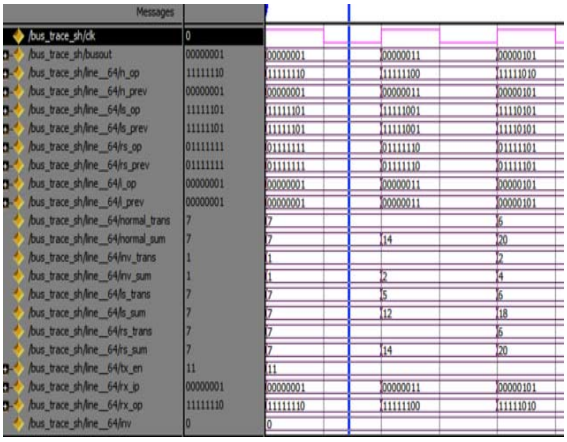

Figure-2. Simulated result to find data vectors with and without coding.

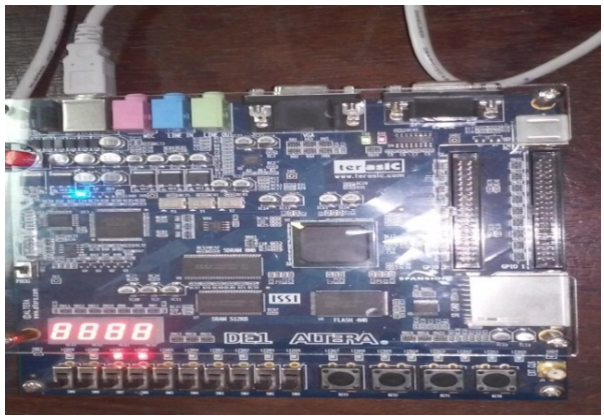

Figure-3. ALTERA cyclone II FPGA implementation.

Here it is clear that by transmitting the data vectors using Embedded Shift Invert Coding large reduction in the number of switching transitions is achieved when compared with transmitting the data vectors using the existing Coding techniques. The performance of all the four Coding techniques is then compared for 16 bit, 32 bit and 64 bit wide data Buses by transmitting randomly generated data vectors. The total transitions in the Bus for duration of 1 ms by applying the four Encoding techniques for each case is tabulated in below Table-4. Coding efficiency of each Encoding technique is then calculated for each Bus width. Coding efficiency indicates how many transitions are reduced after Encoding with respect to total transitions in the normal data. Percentage of Coding efficiency is calculated by using Equation (2).

$$\text{Coding Efficiency in \% encoding} = \frac{\text{No. of transition without encoding} - \text{No. of transitions after encoding}}{\text{No. of transition without}} * 100 \quad (2)$$

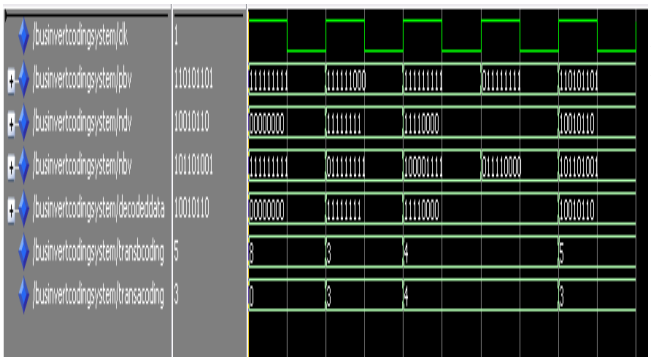

Figure-4. Simulation of Bus invert coding technique.

Table-2. Results from bus invert coding.

Clock Cycle	1	2	3	4	5
Pbv	111111111	111111000	111111111	011111111	110101101
Ndv	00000000	11111111	11110000	11110000	10010110
Trans b Coding	8	3	4	4	5
Nbv	111111111	011111111	100001111	011110000	101101001
Trans a Coding	0	3	4	4	3
Decoded data	00000000	11111111	11110000	11110000	10010110

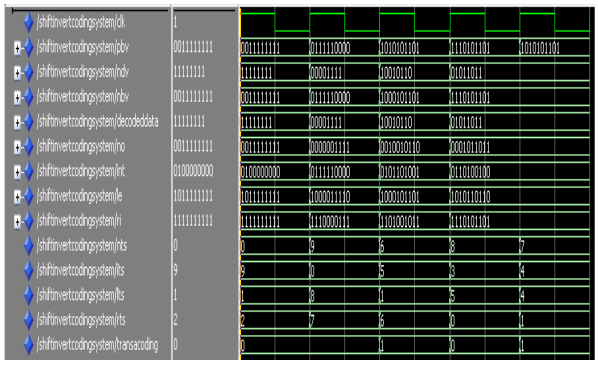

Figure-5. Simulation of embedded shift invert coding technique.

Table-3. Results from embedded shift invert coding.

Clock Cycle	1	2	3	4	5
Pbv	0011111111	0111110000	1010101101	1110101101	1010101101
Ndv	11111111	00001111	10010110	01011011	01011011
Nts	0	9	6	8	7
Its	9	0	5	3	4
Lts	1	8	1	5	4
Rts	2	7	6	0	1
Nbv	0011111111	0111110000	1000101101	1110101101	1110101101
transaCoding	0	0	1	0	1
decodeddata	11111111	00001111	10010110	01011011	01011011

Coding efficiency in % of each encoding technique is compared for all available bus encoding techniques.

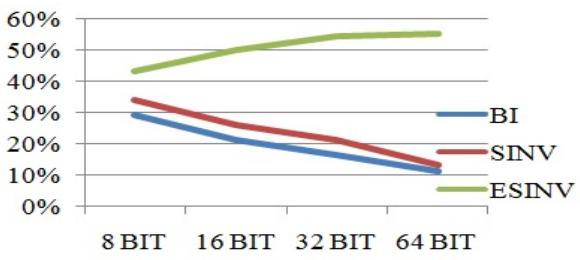

Figure-6. Coding efficiency comparison plots of different Encoding techniques.

5. SYNTHESIS RESULTS

The synthesis of Bus Invert Encoding system, Shift Invert Encoding System, and Embedded Shift Transition Invert Encoding System are done using Quartus II 10.1 Design Suite. In this paper, we modify the bus-invert coding method to maximize the power consumption reduction of data bus. Unlike the conventional scheme in which the whole bus lines are considered for bus-invert coding, our scheme partitions the bus lines into several sub-buses and each partitioned sub-bus is coded independently by bus-invert coding method.

Table-4. Comparison of coding efficiency in %.

Bus encoding techniques	Bus width			
	8 Bit	16 Bit	32 Bit	64 Bit
Bus invert coding	29%	21%	16%	11%
Shift invert coding	34%	26%	21%	13%
Embedded shift invert coding (with two equal width bus data partitions)	43%	50%	54%	55%

Table-5. Bus invert coding power summary.

Power analyzer summary	
Total thermal power dissipation	29.75mW
Core dynamic thermal power dissipation	0.00 mW
Core static thermal power dissipation	18.00mW
I/O Thermal power dissipation	11.75mW

Table-6. Shift invert coding power summary.

Power analyzer summary	
Total thermal power dissipation	67.67mW
Core dynamic thermal power dissipation	0.00 mW
Core static thermal power dissipation	47.35mW
I/O Thermal power dissipation	20.32mW

Table-7. Embedded shift invert coding power summary.

Power analyzer summary	
Total thermal power dissipation	29.53mW
Core dynamic thermal power dissipation	0.00 mW
Core static thermal power dissipation	18.00mW
I/O Thermal power dissipation	11.53mW

6. CONCLUSIONS

In this paper, to reduce the switching transitions in a wider Bus, Embedded Shift Invert Coding technique is designed and simulated and its performance is compared with the existing Bus Encoding Techniques like Bus Invert Coding, Shift Invert Coding and Embedded Bus Invert Coding. First the Encoding and Decoding logics of all the three existing Bus Encoding techniques are designed and simulated for 8 bit, 16 bit, 32 bit and64 bit wide data Buses. The proposed Embedded Shift Invert Coding technique is first designed and simulated by partitioning the Bus data into two equal width data partitions and applying conventional Shift Invert Coding technique to each of the data partitions. Then its performance is compared with the existing Bus Encoding techniques for 8 bit, 16 bit, 32 bit and 64 bit wide data Buses by transmitting randomly generated data vectors. In each case first the data vectors are transmitted without any Coding and then they are transmitted by applying the four Encoding techniques. The transition count and the Coding efficiency are evaluated in each case. The Coding efficiency of all the existing Encoding techniques decreases as the Bus width increases. But the Coding efficiency of proposed Coding technique increases with Bus width. The proposed Embedded Shift Invert Coding technique with two equal width Bus data partitions has got more than 50% efficiency for wider Buses and it is more than 40% efficient.

REFERENCES

[1] M. B. Abdelhalim, A. E. Salama and S. E. –D. Habib. 2006. “Hardware software partitioning Using Particle Swarm Optimization Technique,” The Sixth International Workshop on System on Chip for Real time Applications, pp. 189-194.

[2] Bill Moyer.2001. “Low Power Design for Embedded Processors”, in: Proceedings of the IEEE, Vol. 89, No. 11.

[3] H Guo and Y Zhou. 2009. “A Segmental Bus Invert Coding Method for Instruction Memory Data Bus Power Efficiency,” Proceeding of the 2009 IEEE International Symposium on Circuits and Systems (ISCAS2009 Taipei, Taiwan, pp. 137- 140.

[4] Jayapreetha Natesan and Damu Radhakrishnan. 2004. “Shift Invert Coding (SINV) for Low Power VLSI”, in: Proceeding of Digital System Design, pp. 190-194.

[5] Junkai Sun and Anping Jiang. 2011. “Embedded Bus-Invert Coding for Power Consumption Optimization of Data Bus”, IEEE Transactions on VLSI Systems, Vol.18, No.5, pp.452-455.

[6] Mehdi Kamal, Somayyeh Koohi and Shaahin Hessabi. 2011. “GPH: A group-based partitioning scheme for reducing total power consumption of parallel Buses” in Microprocessors and Microsystems 35, pp. 68–80.

[7] A. Sathish, M. Madhavi Latha and K. Lalkishor. 2011. “An efficient Switching Activity Reduction Technique for On-Chip Data Bus”, in: IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No. 2.

[8] Y. Shin, S. Chae and K. Choi. 2000. “Partial Bus-Invert Coding for power optimization of application specific system,” IEEE Transaction on VLSI System, Vo1.9, No.2, pp.377-383.

[9] M.R. Stan and W.P. Burleson. 1995. “Bus-Invert Coding for low-power I/O,”IEEE Transactions on VLSI Systems, Vol.3, No.1, pp.49-58.

[10] C. L. Su, C. Y. Tsui and A. M. Despain. 1994. “Saving Power in the Control Path of Embedded Processors”, IEEE Design and Test of Computers, Vol. 11, No.4, pp. 24-30.

[11] D. Vijendra Babu, P. Subramanian and N. Ravikannan. 2008. “Micro Blaze and UCLINUX Based Data Acquisition on SPARTAN 3E”, Proceedings of International Conference on VLSI & Embedded Systems ‘08(IC VLSI ‘08), February, pp. 1-4, ISBN: 81-8424-300-6.

[12] D. Vijendra Babu and N. R. 2010. Alamelu, “Implementation of Energy Efficient Wavelet Transform in Spartan 3E FPGA”, International Journal on Computer Applications (IJCA), Vol.12, Issue No.9, February 2010.ISSN No.: 0975-8887. DOI-10.5120/263-422.