Design and Implementation of Address cum Data bus Encoder for Low Power

Shankaranarayana Bhat M[1], Vivek Hegde N[2], Cletus Crasta[2], K. M. Balakrishna[3]

Abstract— **Power dissipation is a major concern in today's VLSI circuits and systems as it adversely impacts feasibility, portability, and reliability. The power dissipation in buses is significant in such systems and constitutes a substantial portion of the dynamic power consumed. Bus encoding techniques were developed for reducing bus power by way of encoding the data before they were sent on the buses. The encoding techniques will be of either redundant or irredundant type based on whether they employ extra bus lines for their operation or not. The paper focuses on design and implementation of an irredundant encoding technique called AMBITS, which is an adapted version of MBITS suitable for multiplexed data and address buses. The number of bit transitions and power dissipation in the case of AMBITS was found to be less than that of MBITS. However, the power and the area overhead for the encoder and decoder in the case of AMBITS were found slightly higher compared to the MBITS on the implementation of physical design using Cadence tool.**

Index Terms— **Bus encoding, Low power VLSI, Switching activity, Power dissipation.**

I. INTRODUCTION

Power consumption is one of the most important design constraints in VLSI based systems particularly in portable systems such as cell phones, personal digital assistants (PDA) and embedded systems. It is shown that a significant portion of overall dynamic power is dissipated while driving the off-chip bus, because of the large off-chip driver, the pad capacitance, and high off-chip capacitance [1]. Power consumed by off-chip driving becomes a dominant factor as the devices are scaled down, as off-chip capacitance depends more on the packaging and printed circuit board (PCB) technologies and less on the process technology [2]. One of the ways available to combat such power dissipation due to high capacitance is to reduce the amount of switching or the bit transitions on the bus, which in turn reduces the effective capacitance need to be driven. Bus encoding techniques were employed to reduce the bit transitions on the bus, which results in lower power consumption in a bus.

Various low power encoding methods are available focusing on the reduction of bit transitions on the buses. Such encoding techniques can broadly be classified into redundant and irredundant techniques based on whether they make use of extra bus line or not. An encoding technique is called redundant technique if it makes use of an extra bus line or lines for its functioning. Literature has a rich collection of redundant encoding techniques. Bus Invert encoding (BI), Bus Invert Transition Signaling (BITS), Universal rotate and invert, partial inversion are some of them [2]-[5].

The extra bus line or lines used in redundant encoding techniques such as in BI or BITS [2],[3] encoding makes it difficult to use in real circuit design as it requires modifications to the interface specification of the chip. Moreover, power reduction is obtained at the cost of the overhead of encoding and decoding circuits, which introduce delay, occupy the additional area and add to power dissipation. To overcome the need for such extra bus line and reduce the overhead of encoding and decoding circuits, the irredundant encoding techniques were proposed by the researchers [6],[7]. The irredundant encoding schemes, as the name suggests, do not make use of the extra bus line for their functioning and hence shun the need for modification in the processor architecture. This feature of irredundant encoding helps reducing area and other overheads, and thus, it results in a better optimization.

The irredundant schemes, in addition to getting rid of the extra bus line or lines, also have an impact on the end to end delay as well as the power dissipated in the bus lines. Approximate Bus Invert Transition Signaling (ABITS), Gray coding with MSB as a reference, Fourth and Fifth bit ANDing (FFA) and Modified Bus Invert Transition Signaling (MBITS) are some of the examples for the irredundant encoding [8]-[9]. Among these the MBITS was found to have the least power dissipation for random data [10].

The above encoding styles assume the random data on the buses and work reasonably well for such situations. However, in reality, the data on the bus can be random as well as sequential. For example, when the address is placed on the bus, it is predominantly sequential. Asymptotic Zero-Transition Encoding (T0) is a redundant encoding technique developed exclusively for the address bus [11]. In processor based systems, the buses are usually multiplexed for saving the number of external pins. In such a scenario, the data on the busses will be both random as well as sequential. To address this situation, the mixed encoding techniques are used. T0-BI, Dual-T0, and Dual T0-BI [12] are some of the

Manuscript received Oct, 2017.

Shankaranarayana Bhat M*, Associate Professor, Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal University, Manipal, Udupi, Karnataka, India*

Vivek Hegde N, *Student, Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal University, Manipal, Udupi, Krnataka, India.*

Cletus Crasta*, Student, Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal University, Manipal, Udupi, Krnataka, India.*

K. M. Balakrishna*, Professor, Department of Physics, Mangalore University, Mangalagangothri, Konaje, Mangalore, Karnataka, India.*

examples. They switch between the two encoding styles based on whether the data or the address is placed on the bus in addition to employing extra bus lines.

This paper is based on the work carried out with the primary objective to improve the MBITS encoding scheme, which works well for the random data in such a way that it can be adapted to handle sequential data as well and still result in lower power dissipation.

II. Encoding Techniques

A. Bus Invert Encoding Technique

The very initial encoding technique employed for low power buses was Bus Invert Encoding (BI) technique [3]. It uses one extra bus line, say, INV, to indicate whether the data is sent with inversion or not to facilitate the decoder to reproduce the original data at the destination. If the Hamming distance between the current data on the bus and the next data to be sent is greater than N/2, (where 'N' is the width of the bus), then the data is inverted (bit level complementation) and transmitted over the bus by setting INV line to '1'. On the other hand, if the Hamming distance is less than or equal to N/2, then the data value is transmitted as it is along with INV=0. BI technique is simple and easy to implement with an upper bound on the power reduction is limited to 25%. Moreover, BI technique uses less overhead, performs well for narrow buses [8] and in general, applicable for both random and sequential data. However, for purely sequential data, such as in the case of instruction address bus, there are other better encoding techniques available, which perform better than BI coding. Gray code, T0 coding, etc., are more appropriate for such applications and achieve a higher reduction of power [8].

B. Bus Invert Transition Signaling Technique

In this technique, if the number of 1s in the input data is more than N/2, then each bit is inverted and encoded using the transition signaling. Else, each bit of the data is encoded using transition signaling without any modification. Even in this case, an extra bus line is used to indicate whether the data is inverted or not [2].

The encoding techniques discussed above are some examples of the redundant encoding techniques as they require an additional bus line for carrying the information regarding the alteration of the data to the receiver to help decoding of the data [2],[3].

C. Modified BITS Encoding Technique

This technique is an example of irredundant encoding as it does not require extra bus line. In this technique, three bits (4^{th}, 5^{th}, and 6^{th}) are used as the reference lines and sent to the destination without any modification. In the design of encoder, the 4th and 5th bits are ORed and then ANDed with the 6th bit. If it results in logic '0' upon ANDing, then the previous output is XORed with the present input excluding the reference bits (4^{th}, 5^{th}, and 6^{th}). If the result of ANDing is '1', then the previous output is XNORed with the present input excluding the reference bits. The data obtained with such operation is concatenated with the three reference bits to form the encoded data and will be placed on the data bus as shown in Fig.1.

In the decoder, the original data is recovered based on the following operation. If the reference line bit is '0', the previous output is XORed with the present input. On the other hand, if the reference bit is '1', the previous output is XNORed with the present input. In both the cases, the three reference bits are excluded [9].

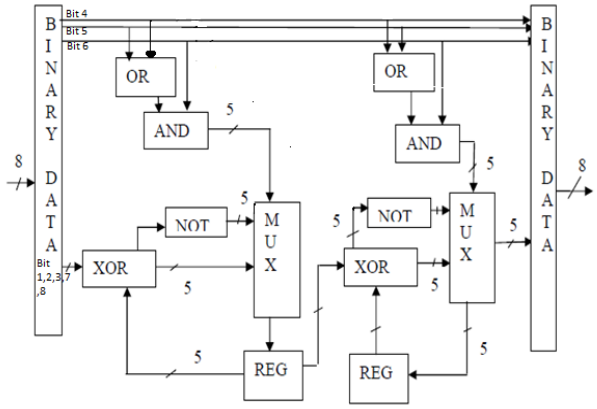

Figure 1: MBITS encoder and decoder

III. Adapted MBITS Technique

In this technique, the input is identified as the data and the address in the very beginning itself depending on the status of the A/D_bar line and processed differentially by the encoder. If the input is a random data, the operation of the existing MBITS circuit is employed with 4^{th}, 5^{th}, and the 6^{th} lines being utilized to generate the select line for deciding XNOR or XOR operation. On the other hand, if the input is an address (sequential data), the circuit for AMBITS was implemented with 6^{th}, 7^{th}, and 8^{th} lines being employed to generate the select line and hence is an adapted version of MBITS scheme. The encoder and decoder architecture is shown in Fig.2.

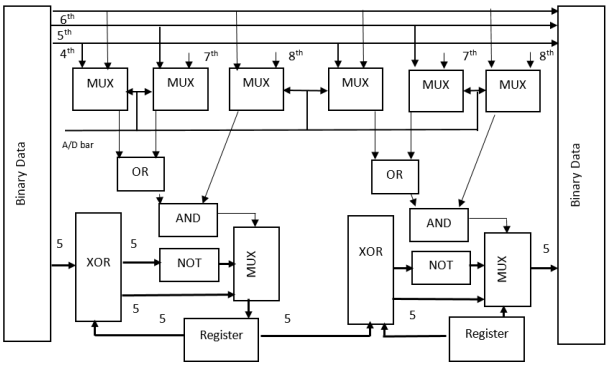

Figure 2: AMBITS encoder and decoder

IV. Methods

The architectures for encoders and decoders of MBITS and AMBITS were designed using Verilog. Subsequently, the schematics of the respective architectures were constructed using Simvision, and their functionality was verified in real

time using the same software platform. The analysis of power dissipation was performed and calculated using Cadence Encounter tool.

For the purpose of simulation of the circuits, four different sets of input data stream consisting of both the sequential and the random data were applied and the corresponding outputs were analyzed for verifying the functionality. Random data had been generated by Leaner Feedback Shift Register (LFSR) using a c++ program, and various combinations of sequential data were taken to keep the results unbiased.

The architecture of AMBITS was implemented using Cadence Incisive tools. The physical design of the MBITS and Adapted MBITS (AMBITS) was carried out using Cadence and analyzed for power and area. In an n-bit bus, the power dissipation is n times the power dissipation occurring in one single line. To calculate the switching activity on the bus lines, we need to have a common reference data, which can be applied to various encoding schemes. The number of transitions between the successive states before and after encoding is calculated. Being all other parameters are same during the experiment, except the switching activity, the power dissipation can be measured based on the change in the switching activity on the bus lines. A set of data was used as a reference for calculation of the power dissipation on the bus lines for the different encoding schemes. The number of transitions between the successive states before and after encoding was then calculated, which in turn showed the change in switching activity and hence the change in power dissipation.

V. Results and Discussion

The MBITS encoder worked satisfactorily for random data input. However, when the same circuit was tested for the sequential data, the results were quite different. In general, in a program, typically, about 15% of the instructions are of program transfer type or branching instructions. This means that, on the instruction address bus, 85% of the time there will be sequential accesses and for the remaining 15% of the time the data on the bus will be non-sequential. The AMBITS encoding treats the input data in a differential way based on the status of the A/D_bar line. If the data were random, the AMBITS technique reduces to MBITS, and if the data were address, the AMBITS feature comes into effect, thereby working for both type of inputs.

The physical design of encoder and decoder circuits for both the MBITS and the AMBITS was done using the Cadence Encounter tool in 45nm technology. Further, the analysis for area and power for the encoders and decoders was carried out, and the reports are generated (Table 1). It can be seen that the encoder and the decoder circuits of AMBITS occupied 29.33% more area and dissipated 15.97% more power when compared to the MBITS encoding. This is because the AMBITS requires extra hardware as compared to MBITS encoding technique.

Table 1: Area and power dissipation details

Parameter	**MBITS**	**AMBITS**
Area	150	194
Design Power (nW)	4155.570	4819.649

The design had been tested with different sets of data. Four sets of data were employed for testing the design. The first three sets were a different combination of the sequential data. SET 1 and 2 had the 7th and 8th bits as 00 and 11 while SET 3 had a mix of 01 and 10 (50% of the data has 7th and 8th bits as 01 and the rest with 10). The SET 4 was a random data generated using LFSR. For each of the inputs the bit transition was analyzed and recorded (Table 2). The corresponding power dissipation was calculated and recorded as given in Table 3.

Table 2: Number of bit transitions for different data sets

Data Sets	**Number of bit Transitions**	
	MBITS	**AMBITS**
SET 1	97	77
SET 2	113	78
SET 3	81	76
SET 4	121	121

Table 3: Power dissipation for different data sets

Data Sets	**Power Dissipated in nW**	
	MBITS	**AMBITS**
SET 1	5667.731	5420.229
SET 2	5931.070	5072.435
SET 3	5787.429	5376.173
SET 4	6526.559	7109.873

The results clearly showed that the AMBITS bus encoding scheme helped reduction of switching activity on bus lines for the first three sets of data, which in turn resulted in a reduction of power dissipation. This fact established the reduction in bus power, which is in line with the reduction of bit transitions due to the AMBITS encoding performed on the input data. The data SET 4 being random data, the AMBITS did not show any saving in switching activity, instead, it matched with that of the MBITS. However, the circuits of the AMBITS dissipated 8.93% higher power, which was due to the requirement of the additional hardware.

VI. Conclusion

The work focused on design, implementation, and evaluation of irredundant encoding technique AMBITS. The design and the implementation of both the MBITS and the AMBITS encoding were done, and the analysis for power

dissipation and area requirements was carried out. The physical design of encoders and decoders for both the MBITS and the AMBITS was done using Cadence tool. It was observed that the number of bit transitions and power dissipation in the case of AMBITS was less than that of MBITS when the test data was applied as the inputs making it suitable for multiplexed data and address bus. However, the area overhead and the power dissipation in the encoder and decoder part of AMBITS were slightly higher compared to the MBITS.

REFERENCES

[1] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," IEEE Journal of Solid-State Circuits, vol. 29, pp. 663-670, 1994.
[2] Y. Shin, K. Choi, and Y.-H. Chang, "Narrow bus encoding for low-power DSP systems," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 9, pp. 656-660, 2001.
[3] 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, 1995.
[4] S. M. Bhat and D. Y. Jahnavi, "Universal rotate invert bus encoding for low power VLSI," International Journal of VLSI Design & Communication Systems, vol. 3, p. 97-109, 2012.
[5] Y. Shin, S.-I. Chae, and K. Choi, "Partial bus-invert coding for power optimization of application-specific systems," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 9, pp. 377-383, 2001.
[6] Y. Aghaghiri, F. Fallah, and M. Pedram, "Irredundant address bus encoding for low power," in Proceedings of the 2001 international symposium on Low power electronics and design, pp. 182-187, 2001.
[7] S. Komatsu, M. Ikeda, and K. Asada, "Bus data encoding with coupling-driven adaptive code-book method for low power data transmission," in Solid-State Circuits Conference, 2001. ESSCIRC 2001. Proceedings of the 27th European, pp. 297-300, 2001.
[8] Y. Shin and K. Choi, "Narrow bus encoding for low power systems," in Design Automation Conference, 2000. Proceedings of the ASP-DAC 2000. Asia and South Pacific, 2000, pp. 217-220.
[9] C. Goyal and I. Sood, "Low Power Data Bus Encoding & Decoding Schemes," ed: July, 2007.
[10] M. Shankaranarayana Bhat, S. Agiwal, D. B. Fernandes, and K. Balakrishna, "Implementation and Evaluation of Redundant and Irredundant Bus Encoding Techniques for Low Power VLSI," International Journal of Current Trends in Science and Technology, vol. 7, pp. 20336-20343, 2017.
[11] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, "Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems," in VLSI, 1997. Proceedings. Seventh Great Lakes Symposium on, 1997, pp. 77-82.
[12] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, "Address bus encoding techniques for system-level power optimization," in Proceedings of the conference on Design, automation and test in Europe, 1998, pp. 861-867.