



A Novel Multiply-Accumulator Unit Bus Encoding Architecture for Image Processing Applications

R. Samanth*, S. G. Nayak^{*(C.A.)}, and P. B. Nempu**

Abstract: In the CMOS circuit power dissipation is a major concern for VLSI functional units. With shrinking feature size, increased frequency and power dissipation on the data bus have become the most important factor compared to other parts of the functional units. One of the most important functional units in any processor is the Multiply-Accumulator unit (MAC). The current work focuses on the development of MAC unit bus encoders as well as the identification of an improved architecture for image processing applications. To reduce the power consumption in these functional units, two bus encoding architectures were developed by encoding data before it was sent on the data buses. One is MSB reference encoding, and another is Fourth and Fifth bit ANDing (FFA) without the need for an extra bus line with fewer transitions by using gray codes. The comparison of the proposed encoding architectures with the existing encoding architectures from the literature revealed an 8% to 36% significant improvement in power dissipation. The simulation was done with Xilinx ISE, and the Cadence RTL Compiler tool was utilized for the synthesis, which was done with the 180nm technology library. And also, the image filtering is analyzed using MATLAB.

Keywords: Data Bus Encoders, Fourth and Fifth Bit ANDing, Gaussian Filter, MSB Reference Encoding, Multiply-Accumulator Unit.

1 Introduction

IN every digital image and signal processor, the multiply-accumulator (MAC) unit is the fundamental computational block. The majority of image processing and signal processing operations including Fourier Transforms, filtering, sharpening, wavelet transforms, and feature detection mechanisms, require multiple multiplication and additions. While performing multiplication and addition there is a chance of high data traffic on the data bus due to technological advancements, resulting in a great quantity of switching

on the bus. This leads to a significant amount of power dissipation in MAC units. A multiplier, adder, and accumulator register comprise a standard MAC. Hence before sending the data on the data bus to the multiplier, adder, and accumulator to reduce the transitions there have been numerous studies conducted in the past. Power dissipation can be static or dynamic. The leakage current in transistors causes static power dissipation while switching power and short circuit power cause dynamic power dissipation. Switching power is dissipated when a transition occurs from 0 to 1 or 1 to 0. Switching activity refers to the likelihood of such a transition [1, 2]. These transitions are directly proportional to power consumption thus reducing the bus switching in an efficient way to reduce the bus power consumption [3].

In whole systems, buses are the major source of dynamic power dissipation. The global bus lines in digital circuits are typically loaded with huge capacitances on both the OFF-CHIP and ON-CHIP sides [4], [25]. Because of the huge amount of data transferred in today's VLSI circuits in the digital domain, bus capacitance and power losses compete with processor power. Sometimes overpowering it, and switching is a major contributor to this loss. The use of

Iranian Journal of Electrical and Electronic Engineering, 2023.
Paper first received 14 January 2022, revised 25 September 2022, and accepted 28 September 2022.

* The authors are with the Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Udupi, Karnataka, India.

E-mails: rashmi.samanth@gmail.com and gs.nayak@manipal.edu.

** The author is with the Department of Electrical and Electronics Engineering, Nitte Meenakshi Institute of Technology Bengaluru, Karnataka, India.

E-mail: pramod.bhat@nmit.ac.in.

Corresponding Author: S. G. Nayak.

<https://doi.org/10.22068/IJEEE.19.1.2391>

different number representation systems, counting sequences, coding methods, and data formats can all have an impact on the switching frequency of a design [5]. In terms of power dissipation, lowering switching frequency has the same effect as lowering capacitance. The charging and discharging capacitance and short circuit power are directly influenced by these switching actions. The amount of switching is reduced as a result of this feature as well as the system's reliability is improved.

The expression for dynamic power is given by:

$$P_{dynamic} = \alpha C_L V_{dd}^2 f_{clk} \quad (1)$$

where, α is the switching activity, C_L is physical capacitance at the output node, V_{dd} is the voltage supply and f_{clk} is the clock frequency [2].

The present work proposes to explore a novel multiply-accumulator unit bus encoding architecture for image processing applications. The main contribution of this work is to implement an efficient MAC unit with two bus encoding architectures by taking power dissipation constraints from the literature into account. The novel design uses MSB reference and FFA encoding approach to minimize the MAC architecture's computation power. The developed encoders are integrated into the MAC unit for image filtering. The paper is structured as the past research on various encoding data schemes is given in Section 2. Section 3 explains the theory as well as the proposed MAC encoder designs methodology and applications. The data analysis and the results and inferences are presented in Section 4. Finally, in Section 5, the conclusion is presented.

2 Previous Works

There are a variety of encoding and decoding techniques available to reduce switching activity on the data bus. The most popular techniques are BITS and BI (bus invert) [7, 8]. (Bus invert transition signaling). In order to use these techniques, a second bus line is needed. However, very few of the existing schemes consistently achieve significant reductions in data and multiplexed address bus transition activity. The bus inversion coding technique is a popular coding system for data buses. The Bus-Invert coding method for low power I/O was created by Stan *et al.* [9] to minimize switching activities in the data bus. This approach is universal, but it functions best with buses. This is beneficial because buses are likely to have big capacitances and, as a result, dissipate a lot of power.

In order to address the issue of power dissipation for intellectual property core processors, the low-power address bus encoding solution is presented by Aghaghiri *et al.* [10]. Since there are no redundant lines in the developed technique, it has an advantage over existing techniques. As a result, an additional expensive pin is saved. The drawback was they used T0 coding instead

of gray coding which works for all the data streams except for the streams containing all 1's. Singh *et al.* [11] used gray code and T0 code for memory address bus encoding. The amount of bit switching that takes place on address buses is decreased by bit encoding.

Wei-Chung Cheng and M. Pedram [12] employed the quick and effective Bus Inversion (BI) technique, which can significantly reduce the activity factor. It had been reported in J. V. R. Ravindra, M. B. Srinivas [6] that, bus inversion technique can lower peak power by up to 50% and in Olivieri *et al.* [13] average power reduced up to 25% [13]. The inadequate power savings of BI in the average scenario prompted changes such as splitting data lines (depending on each line's activity factor) [13], [14]; arranging data lines into odd and even groups and applying BI separately to each group [15]. For system-level bus power optimization, Youngsoo *et al.* [16] developed a partial bus-invert coding method. Hong *et al.* [17] identified a subset of bus lines for bus encoding in the proposed system to reduce the overall number of bus transitions. When compared to unencoded patterns, it reduced total bus transitions by 62.6% on average code [18], T0 code [19] have been proposed for instruction buses, which simplify transitions while lowering power consumption. S. Vinay [20] MAC unit for filter design is developed using bit reversed encoding. Bit reversed refers to the fact that the ordering corresponds to what a binary counter would produce if the bits were taken in reverse order (that is the least significant bit first). The drawback of FFT-based MAC is that it either takes input or leaves outputs data in a jumbled sequence, which means that the data must be reorganized at some point.

Zhang *et al.* [21] introduced a novel bus coding technique where the data bus is separated into even and odd bit groups. The Hamming distance between even and odd subgroups, reversed data, and current data are related to the data on the bus. For decoding reasons, the data of the sub-group with the shortest Hamming distance is inverted and transmitted with two redundant control bits. Coupling transitions occur as a result of duplicated bits, which was one of the technique's drawbacks [22].

The data was encoded using an encoding method and sent with redundant bits to indicate whether the encoding is being used or not before being sent on the bus. The data on the bus can be sequential or random when compared to earlier data. Sequential data and prior data refer to the same thing; the only difference between the two values is the stride value which is developed by Naveen *et al.* [23]. Because the number of bits deleted during calculation is directly proportional to the quantity of energy dissipated in a functional unit. The number of transitions reduced is very similar to that of the original bus invert coding method. S. Saravanan *et al.* [24] developed a MAC unit using the Hybrid Encoded Reduced Transition Activity

Technique (HERTAT). The developed technique helped to reduce the switching activities as well as used for image processing applications.

Keeping the above facts, a methodology has been presented that reduces the switching activity on buses without the need for additional bus lines, as well as encoder and decoder hardware that is very compact and consumes very little power compared to conventional designs. And also, these encoders are integrated into the MAC unit for image processing applications.

3 Proposed Design

The basic data bus encoding architecture comprises encoder and decoder blocks on both the source and destination sides of this basic digital system have that bidirectional data lines. When the source wants to transfer data, the data is passed to the encoder, which then encodes the data to reduce the number of switching activities. The encoded data is sent through the data line and received by the decoder, which restores the original form of the data. At the receiver end, the original data is received with a significant reduction in switching power. In this paper, two novel MSB reference and Fourth and Fifth bit ANDing (FFA) encoder architectures are developed and evaluated which do not require an extra bit line. These encoders will reduce the switching activity in the MAC unit with the help of gray codes. The developed designs are then compared with the results of BI, BITS, and HERTAT-based conventional encoder architecture, MAC designs.

3.1 Binary Versus Gray Coding Counting

Gray code, is a binary sequence in which only one bit changes value when transitioning between neighboring states also known as reflected binary code. These codes are superior to other codes because they use half the power of binary codes. Consider the implementation of two N-bit counters using gray and binary code counting sequences. The binary counter has twice as many transitions as the gray counter when the number of transitions is significant. A gray code is often more power-efficient than a binary counter since power consumption is connected to switching activity. Gray code has reflective, cyclic, and unit distance features. Table 1 compares binary and gray code transitions using sequences and the number of toggles that correspond to

Table 1 The sequence of binary and gray codes [25].

Binary Code		Gray Code	
Sequence	Toggle count	Sequence	Toggle count
000	3	000	1
001	1	001	1
010	2	011	1
011	1	010	1
100	3	110	1
101	1	111	1
110	2	101	1
111	1	100	1

each transition. After 111, the binary sequence will return to 000, requiring three toggles, whereas gray code will only require one toggle state. The following Boolean expressions 2, 3, 4, and 5 will explain how the Binary to gray conversion takes place using XORing the bits. Where B is a 4-bit input binary value and G is a 4-bit gray output.

$$G(3) = B(3); \tag{2}$$

$$G(2) = B(3) \oplus B(2); \tag{3}$$

$$G(1) = B(2) \oplus B(1); \tag{4}$$

$$G(0) = B(1) \oplus B(0); \tag{5}$$

These codes save half of the power as compared to binary codes. Hence these codes are utilized in the proposed techniques to reduce the transitions.

3.2 MSB Reference Encoding Architecture

In this architecture, the given binary data is transformed into gray code, as seen in Fig. 1. The data bus receives the initial gray coded data in its original form. The MSB of the succeeding coded data is then checked. If the value is '0', an XOR operation is performed among the lower N-1 bits of the present coded bits and the lower N-1 bits of the preceding encoded bits. The MSB of the recent data, as well as the XOR output, is delivered in its original form.

If the value is '1', an XNOR operation is performed between the subordinate N-1 bits of the current gray coded data and the subordinate N-1 bits of the prior encoded data. The MSB of the current gray coded data is delivered completely, along with the XNOR output. If the MSB of the data received through the data bus is '0', the decoder performs an XOR operation between the lower N-1 bits of the previous data received and the lower N-1 bits of the current data through the data bus; otherwise, an XNOR operation is done. Finally, the gray coded information is transformed to binary.

The actual MSB reference encoding architecture flow is represented in Fig. 2. Binary bits are sent to the encoder as binary input. This input will be translated to its gray counterpart. If the encoder is in a reset state, the input gray code is equal to the register value on the encoder side, and the register value on the decoder side

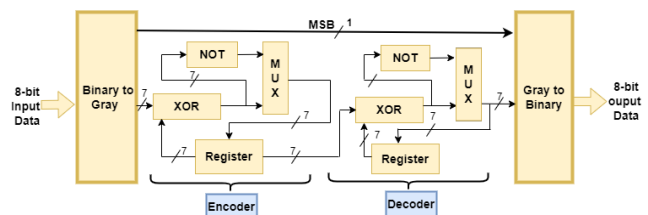


Fig. 1 MSB reference encoding architecture.

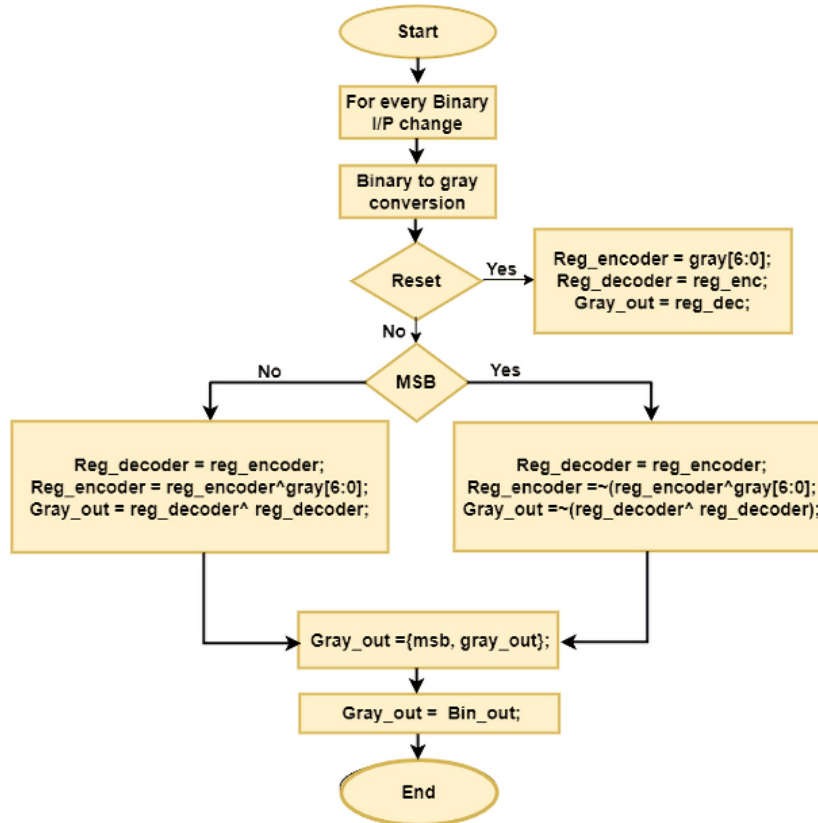


Fig. 2 Flow chart of MSB reference encoding.

is identical to the register value on the encoder side. Gray output is sent when the value in the decoder side register is zero. If MSB is 1 and reset is 0, the encoder side register is equal to the XNOR operation of the previous encoder register value and the current gray code value.

The gray output is equal to the value of the encoder and decoder registers in the XNOR operation. If MSB is '0', the encoder register is equal to the XOR operation of the previous encoder register value and the current gray code value. The XOR operation equals the value of the encoder register and the value of the decoder register. The MSB and gray outputs are combined to create the final gray output. The gray output is then transformed to binary.

3.3 Fourth and Fifth bit ANDing (FFA) Encoding Architecture

The binary input data is converted into gray code and sent across the data bus in this architecture, as shown in Fig. 3. Logical AND operation is conducted among the 4th and 5th bits of the following gray coded bits from an 8-bit data. If this operation returns a value of '0', an XOR operation is performed between the remaining six bits of gray coded data (lower three bits and upper three bits) and the previously encoded six bits (lower three bits and upper three bits). The XOR output, as well as the 4th and 5th bits, are sent to the data bus.

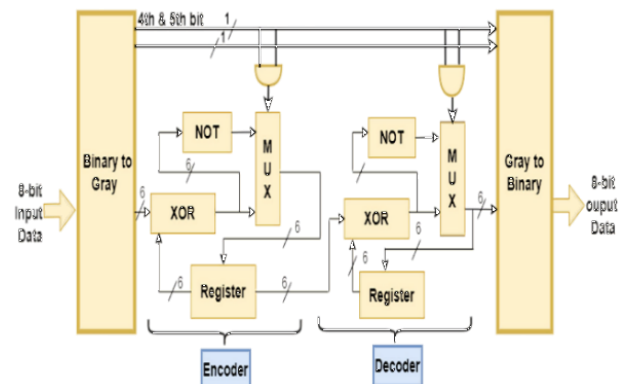


Fig. 3 FFA encoding architecture.

If '0' is returned, an XOR will be performed between the received six bits of data and the preceding received six bits of data. The received output will be converted back to binary.

The complete flow chart of the FFA encoding architecture is shown in Fig. 4. Every binary input will be translated to its gray counterpart. If the reset button is enabled, the fourth and fifth gray coded bits are ANDed, and the remaining 6 bits are concatenated and put in the encoder's register. Between the register encoded and decoded values, gray out is now equal to XNOR.

The encoder register value is transmitted to the decoder register if the reset input is '0' and the 4th and 5th ANDing result is '1'. The XNOR of the previous

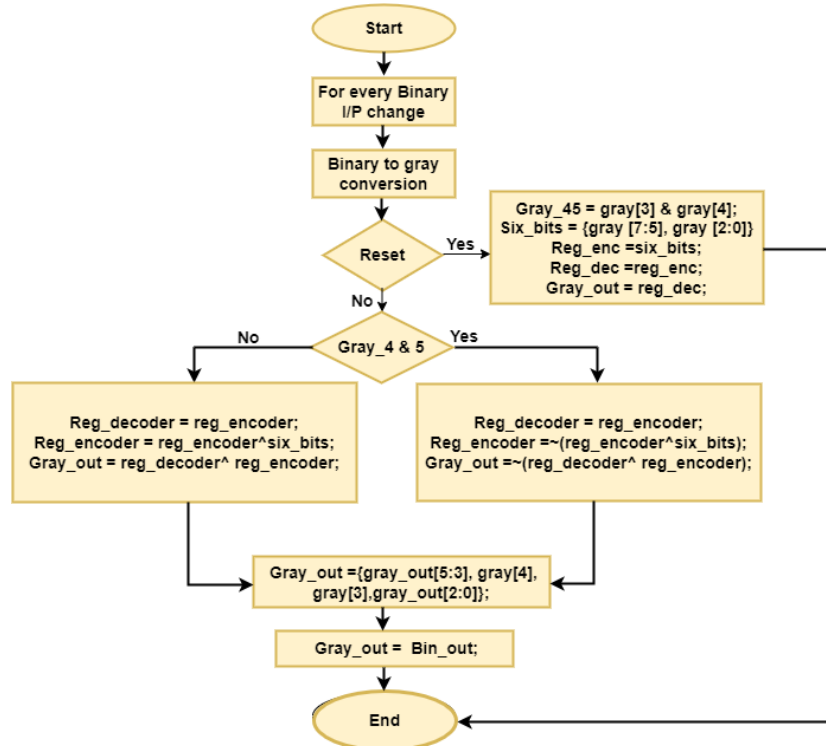


Fig. 4 Flowchart of FFA encoding.

encoder register value and the current gray code value will be stored in the encoder’s register. Now gray out is equal to XNOR between register encoded and decoded value. The encoder register value is transmitted to the decoder register if the reset input is ‘0’ and the 4th and 5th ANDing result is ‘0’ The XOR of the previous encoder register value and the current gray code value will be stored in the encoder’s register. Graying out is now comparable to XORing the register encoded and decoded values.

Finally, the gray value is equal to the concatenation of the gray out’s top 3 bits, the gray input’s fourth and fifth bits, and the gray output’s lower 3 bits. The final gray output is translated to its binary form.

These encoders and decoders were created to be utilized in situations when data is being sent from one location to another. This encoder will reduce the number of transitions by encoding the data before it is placed on the data bus, resulting in a reduction in switching power consumption. The data will be decoded into its original form by the opposite side decoder. The developed encoding architectures are employed in the MAC unit for image filtering applications.

3.4 Proposed MAC Unit for Image Filtering

Multiply, add, shift, and store are the main functions of the MAC unit. In the proposed MAC architecture main function is to encode the data before sending the data on the data buses, then the multiplication, addition, and accumulation process will be performed. The MAC unit performs 2D convolution for 8-bit image data in

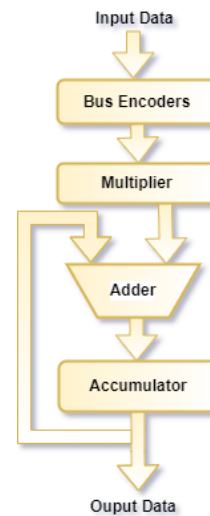


Fig. 5 Proposed MAC unit.

this work. For image filtering, a Gaussian filter is employed to analyze the image convolution. The blurring effect is tested for various σ values. The proposed MAC unit bus encoder architecture is shown Fig. 5.

A typical equation of such a system performing the convolution scheme in 2D array can be framed as:

$$Z(x, y) = Y(x, y) + \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} I(x+m, y+n) \cdot h(m, n) \quad (6)$$

where $I(x,y)$ represents input pixels, $h(m,n)$ is a kernel

with kernel lengths $m, n = 0 : K-1$ width of the image is $x = 0 : M-1, y = 0 : N-1$, and $Y(x,y)$ is accumulator register. $Z(x,y)$ represents the filtered image. The main idea of this unit is to move the window $h(m,n)$ plane over the image $I(x,y)$. The convolver uses K^2 MAC units over the image simultaneously at each clock cycle. Here the 8-bit signed integers operands are used as image pixels and kernels.

The simulation steps of the developed MAC unit in MATLAB:

- A specified filter kernel with an 8-bit fixed point or 8-bit integer is convolved with an image containing 8-bit signed integers.
- To keep the background pixels while performing the convolution, the image matrices are padded with zeros.
- The kernel and image matrices are translated to text as vectors h_i and I_j , with $i = 0$ to $K-1$ and $j = 0$ to $MN-1$, respectively. The image vectors should be in hexadecimal or binary format.

In Verilog:

- A hex file containing image pixels from MATLAB is imported to memory locations to the convolution test bench module.
- From the test bench, MAC units are invoked to convolve the image with the supplied kernels.
- The convolved results are written to a text file in either hexadecimal or binary format and kept in a separate memory block.

4 Results and Discussion

The developed MAC unit bus encoder architectures were implemented using fully-hierarchical Verilog HDL coding in RTL-to-netlist production through synthesis. The entire system was first conceptualized on paper, and then it was subdivided into hierarchical components, encoder, and decoder. These components were then subdivided into basic components.

The encoder and decoder in this case are for 8-bit data (8-bit image data). Fig. 6 shows the RTL Schematic of the MSB reference encoder, which includes all of the encoder and decoder blocks that are built for 8-bit data

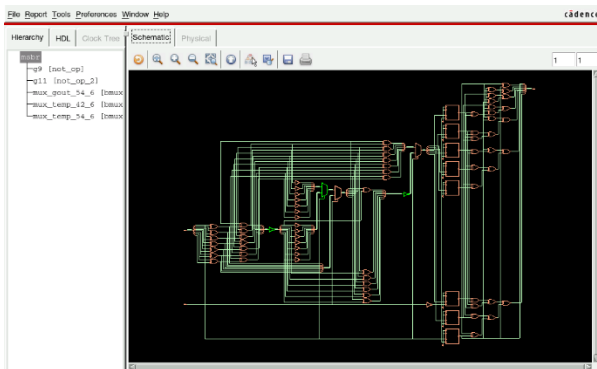


Fig. 6 RTL schematic of MSB encoding architecture.

input and output. Any number of input bits can be added to the same design. The Xilinx ISE (Integrated Software Environment) Design Suit was used to validate functionality and perform power analysis. The performance parameters of these two developed approaches, such as power consumption, area, and delay evaluated using the Cadence EDA tool. For the same techniques Cadence encounter RTL compiler is used to check the RTL schematic.

Fig. 7 shows the timing waveform of the MSB encoder architecture. Here, Input and output values are represented in hexadecimal representation for easy representation. The converted binary bits and the converted gray codes are shown in the timing waveform in hexadecimal form.

Initially, converted gray code data bits are directed as it is, over the data bus. Then the MSB of the second set of gray-coded bits is checked. If it is '0', then the XOR operation is performed. Similarly, as explained the MSB of the present gray-coded data is sent as it is, along with the output of the XOR. If it is '1', then the XNOR operation will be performed. The MSB of the present gray-coded data is sent as it is, along with the output of the XNOR. In the same manner, the decoder will perform depending on the status of the MSB of the data received through the data bus. Then the data is sent to the MAC unit for further computation.

Fig. 8 shows the power and area parameters of the MSB reference encoder which is obtained in Cadence with 180 nm technology library for normal Vt. Power is expressed in nW (nano Watts) and the area for the particular blocks is given in detail.

Fig. 9 shows the Critical path and worst-case delay of the same encoding architecture. The detailed timing report and delay are expressed in ps (Pico seconds) using the Cadence RTL compiler. For high threshold voltage, the subthreshold leakage current is less. For low threshold voltages, subthreshold currents are high and now by increasing the threshold voltage the subthreshold leakages are decreased. Delay is directly proportional to the threshold voltage. Hence, the delay is increased for high threshold voltages.

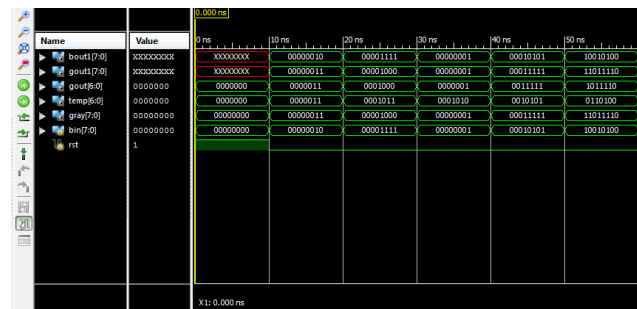


Fig. 7 Timing waveform of the MSB encoder architecture.

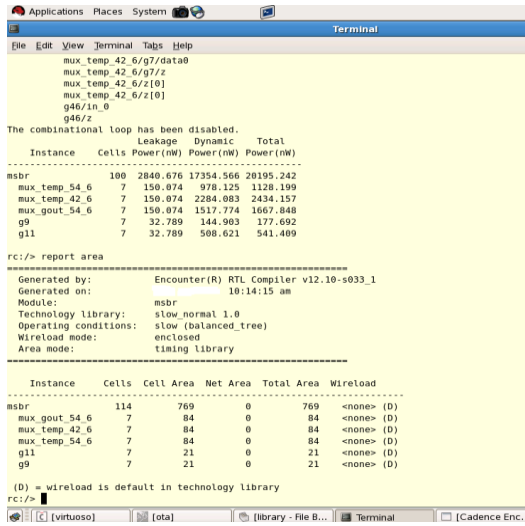


Fig. 8 Power and area of MSB reference encoder.

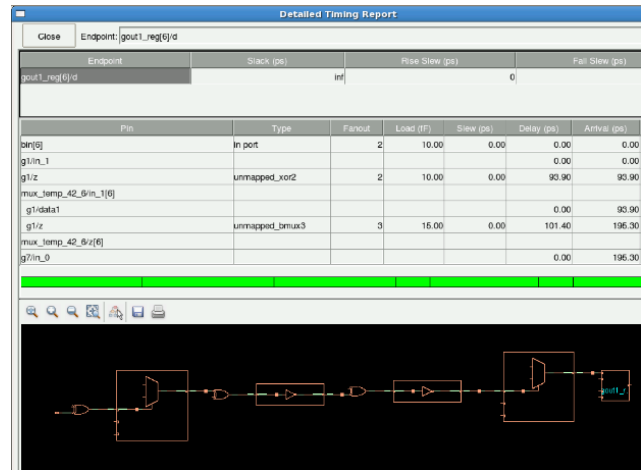


Fig. 9 Delay path of MSB reference encoder.

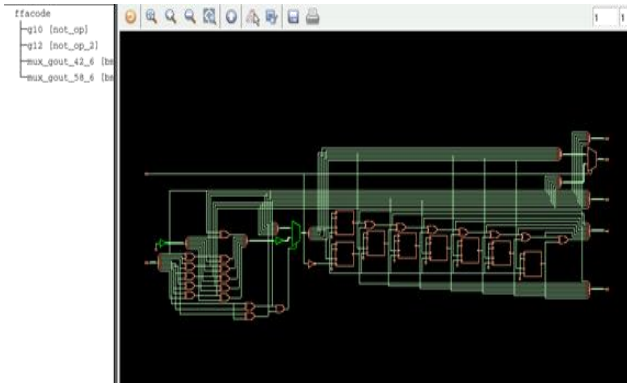


Fig. 10 Power and area of MSB reference encoder.

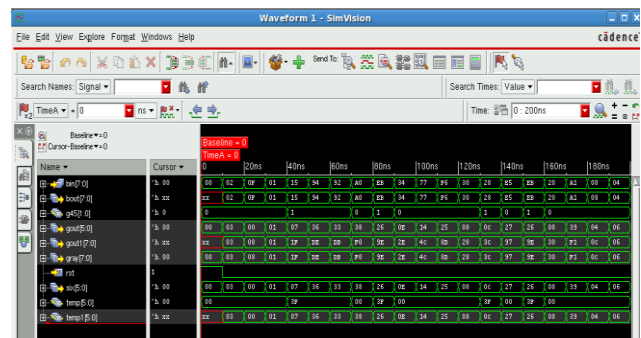


Fig. 11 Timing waveforms of FFA encoder.

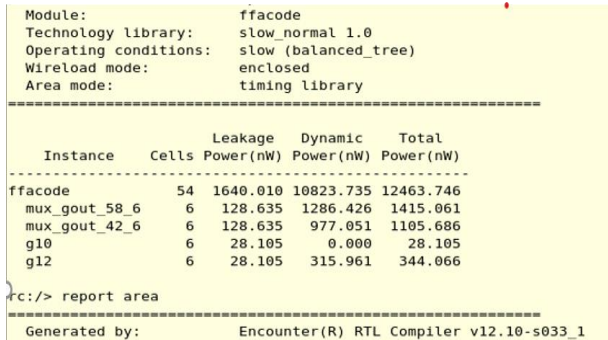


Fig. 12 Power and area of FFA encoder.

Similarly, Fig. 10 depicts the power and area of MSB reference encoder and Fig. 11 represents the simulation result of the FFA encoding architecture for the 8-bit data using the Cadence RTL compiler. The logical operation is performed between the inputs based on the developed encoder.

Fig. 12 demonstrates of power and area parameters of the FFA encoding obtained in Cadence with 180 nm technology library for normal V_t .

Initially, random binary inputs are used to verify the design functionality of the suggested designs and the conventional designs from the literature review. Then 8-bit image data is taken as input to the bus encoders. Finally, the encoded image input image data is fed to

Table 2 Power dissipation comparison of different encoders.

S. No.	Technique name	Encoder power [nW]	Decoder power [nW]
1	BI (Bus invert) [11]	555,956.55	173,342.55
2	TI (Transition inversion) [19]	588,732.90	411,383.57
3	HERTAT [24]	213,349.20	366,369.37
4	Proposed MSB reference	386,547.35	255,112.73
5	Proposed FFA	390,203.12	273,067.29

the MAC unit.

Along with the proposed designs, conventional designs are implemented and verified in the same 180nm technology Cadence platform for performance evaluation. The power is tabulated and compared with other bus encoding designs which are presented in Table 2.

The above table gives the inference that the proposed techniques for low-power data bus encoding and decoding are power efficient and better compared to the existing techniques.

For further evaluation, the developed techniques performance parameters such as power, area, and delay are compared and tabulated in Table 3.

Fig. 13 depicts the MSB reference and FFA with respect to power (nW).

These encoder architectures are better than the existing architecture because they do not need the extra bus line from the encoder to the decoder to indicate the decoder the data is encoded and decoded.

Fig. 14 represents the area and delay comparison of the two proposed techniques. FFA scheme has low power dissipation and less area overhead compared to the technique, whereas the performance of the scheme is better because of less delay. These encoder architectures are integrated into the MAC unit for image filtering. Then developed MAC unit function is analyzed and validated in both Verilog and MATLAB platforms.

4.1 Results of Proposed MAC Unit for Image Filtering

The developed MAC is designed to perform 2D convolution. The PSNR for developed MAC unit bus encoder architecture is evaluated using Gaussian blurring. A normal distribution is used to construct the Gaussian kernel $h(x,y)$ where x and y denote the coordinates of kernel pixels.

$$h(x, y) = \frac{1}{\sigma^2 2\pi} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{7}$$

Table 4 shows the kernels used were Gaussian filters with varying σ measure from 0.5 to 2.

$$PSNR = 10 \log_{10} \frac{N^2}{MSE} \tag{8}$$

where N is the maximum pixel image value. Since an example of an 8-bit integer image was used in this

study, the measure is calculated using the following equation:

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [I_o(x, y) - I_R(x, y)]^2 \tag{9}$$

where I_o represents the original image and I_R represents the filtered image. The PSNR in dB of processed images retrieved from the Verilog $PSNR_V$ and MATLAB $PSNR_M$ platforms is compared to the error values using the following expression:

$$PSNR_{error} = PSNR_M - PSNR_V \tag{10}$$

The PSNR (dB) values were found to have a minimum error E_M-E_V of 0.005 to 0.1 for varying σ of the kernels that are evaluated using MATLAB and Verilog platforms which is shown in Fig. 15.

5 Conclusion

In the present work, a novel multiply-accumulator unit bus encoding architecture is developed for image blurring. Moreover, the proposed data bus encoding architectures do not need the extra bus line from the encoder to the decoder to indicate to the decoder that data is encoded or decoded. Gray codes are utilized to avoid multiple-bit transition problems as well as synchronization problems when the values are passing from one clock domain to another. In comparison with the existing designs, the developed design demonstrated an overall 8% to 36% significant improvement in power. These bus encoder architectures are useful for reducing the power in various VLSI functional units.

Table 3 Power Consumption of Developed encoder architectures.

Encoders type	MSB reference	FFA	Difference between two encoders	Change [%]
Leakage power [nW]	2,840.676	1,640.010	1,200.666	42.26
Dynamic power [nW]	17,354.566	10,823.735	6,530.831	37.63
Total Power (nW)	20,195.242	12,463.746	7,731.496	38.28
Area [μm^2]	769	405	364	47.30
Delay [ps]	93.90	178.30	84.4	89.88

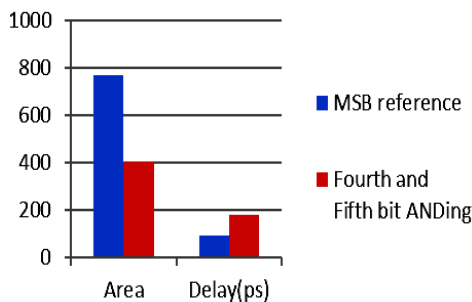


Fig. 14 Area and delay comparison of proposed bus encoders.

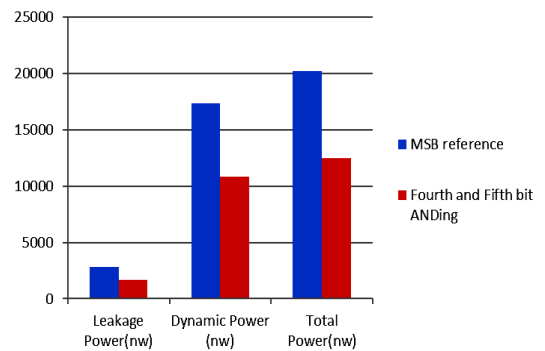


Fig. 13 Power comparison of MSB reference and FFA.

Table 4 PSNR of 2D Gaussian filter.

σ	$PSNR_M$ [db]	$PSNR_V$ [db]	E_M-E_V [db]
0.5	23.94	24.04	0.1
1	25.09	25.14	0.05
1.5	25.27	25.30	0.03
2	25.32	25.35	0.03

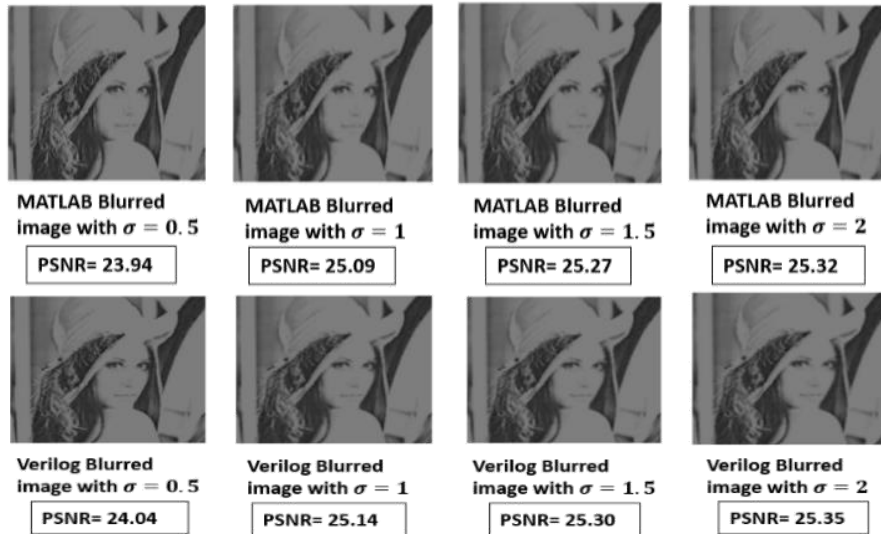


Fig. 15 PSNR evaluation of image filtering.

Intellectual Property

The authors confirm that they have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property.

Funding

No funding was received for this work.

CRedit Authorship Contribution Statement

R. Samanth: Idea & conceptualization, Research & investigation, Analysis, Methodology, Software and simulation, Original draft preparation. **S. G. Nayak:** Supervision, Verification, Revise & editing. **P. B. Nempu:** Analysis, Software Simulation & Result Discussions.

Declaration of Competing Interest

The authors hereby confirm that the submitted manuscript is an original work and has not been published so far, is not under consideration for publication by any other journal and will not be submitted to any other journal until the decision will be made by this journal. All authors have approved the manuscript and agree with its submission to "Iranian Journal of Electrical and Electronic Engineering".

References

[1] N. S. K. Chakravarthy, O. Vignesh, and J. N. Swaminathan, "High speed and low power buffer based parallel multiplier for computer arithmetic," *Lecture Notes in Networks and Systems Inventive Communication and Computational Technologies*, pp. 407–416, 2020.

[2] I. Ratković, N. Bežanić, O. S. Ünsal, A. Cristal, and V. Milutinović, "An overview of architecture-level power- and energy-efficient design techniques," *Advances in Computers*, pp. 1–57, 2015.

[3] N. Chintaiah and G. Reddy, "Low-power sector-based transition reduction bus encoding technique in SOC interconnects," *International Journal of Computer Aided Engineering and Technology*, Vol. 15, No. 23, p. 281, 2021.

[4] D. Suresh, B. Agrawal, J. Yang, and W. Najjar, "Energy-efficient encoding techniques for off-chip data buses," *ACM Transactions on Embedded Computing Systems*, Vol. 8, No. 2, pp. 1–23, 2009.

[5] K. Mehta, "A review on strategies and methodologies of dynamic power reduction on low power system design," *International Journal of Computer Science & Communication*, Vol. 7, 2015.

[6] J. V. R. Ravindra and M. B. Srinivas. "Delay and energy efficient coding techniques for capacitive interconnects," *Journal of Circuits, Systems, and Computers*, Vol. 16, No. 6, pp.929–942, 2011.

[7] R. B. Lin, "Inter-wire coupling reduction analysis of bus-invert coding," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 55, No. 7, pp. 1911–1920, 2008.

[8] E. Maragkoudaki and V. F. Pavlidis, "Energy-efficient time-based adaptive encoding for off-chip communication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 28, No. 12, pp. 2551–2562, 2020.

[9] 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, No. 1, pp. 49–58, 1995.

- [10] 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*, Vol. 11, No. 05, pp. 445–457, 2002.
- [11] B. Singh, A. Khosla, and S. B. Narang, "Low power bus encoding techniques for memory testing," *Microelectron Solid State Electron*, Vol. 2, No. 3, pp. 45–51, 2013.
- [12] W. C. Cheng and M. Pedram, "Power-optimal encoding for DRAM address bus," in *Proceedings of the 2000 International Symposium on Low Power Electronics and Design (Cat. No.00TH8514)*, pp. 250–252, 2000.
- [13] M. Olivieri, F. Pappalardo, and G. Visalli, "Bus-switch coding for reducing power dissipation in off-chip buses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 12, pp. 1374–1377, 2004.
- [14] M. Alamgir, I. I. Basith, T. Supon, and R. Rashidzadeh, "Improved bus-shift coding for low-power I/O," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2940–2943, 2015.
- [15] R. G. Yang and C. Zhang, "Frequent value encoding for low power data buses," *ACM Transactions on Design Automation of Electronic Systems*, Vol. 9, No. 3, pp. 354–384, 2004.
- [16] Y. Shin, S. I. Chae, and K. Choi, "Partial bus-invert coding for power optimization of system level bus," in *Proceedings of International Symposium on Low Power Electronics and Design (IEEE Cat. No.98TH8379)*, pp. 127–129, 1998.
- [17] S. Hong, U. Narayanan, K. S. Chung, and T. Kim, "Bus-invert coding for low-power I/O - A decomposition approach," in *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems (Cat.No.CH37144)*, Vol. 2, pp. 750–753, 2000.
- [18] S. Parunandi and P. Anitha. "Transition inversion based low power data coding scheme for buffered data transfer," *International Journal of Engineering Research & Technology (IJERT)*, Vol. 9, No. 11, pp. 501–506, Nov. 2020.
- [19] R. Abinesh, R. Bharghava, S. Purini, and G. Regeti, "Transition inversion based low power data coding scheme for buffered data transfer," in *23rd International Conference on VLSI Design*, pp. 164–169, 2010.
- [20] S. Vinay, "DSP architectures for system design," in *Tech. Credit Seminar Report, Electronics Systems Group, EE dept, Bombay*, 2002.
- [21] Y. Zhang, X. Hu, X. Feng, Y. Hu, and X. Tang, "An analysis of power dissipation analysis and power dissipation optimization methods in digital chip layout design," *IEEE 19th International Conference on Communication Technology (ICCT)*, pp. 1468–1471, 2019.
- [22] N. H. E. Weste and D. Harris, *CMOS VLSI design: A circuits and systems perspective*, Pearson Education India, 2005.
- [23] N. K. Samala, D. Radhakrishnan, and B. Izadi. "A novel deep submicron bus coding for low energy," in *Proceedings of the International Conference on Embedded Systems and Applications*, pp. 25–30, 2004.
- [24] M. Madheswaran and S. Saravanan, "Modified multiply and accumulate unit with hybrid encoded reduced transition activity technique equipped multiplier and low power 0.13 μ m adder for image processing applications," *International Journal of Computer Applications*, Vol. 1, No. 9, pp. 57–62, 2010.
- [25] G. K. Yeap, *Practical low power digital VLSI design*. Springer Science & Business Media, 2012.



R. Samanth is currently working as a Research Scholar with the Department of Electronics and Communication Engineering, Manipal Institute of Technology. She received the B.E. degree in Electronics and Communication Engineering and a Master's degree in Microelectronics from the Manipal Institute of Technology, Manipal. Her research interests include electronics, digital systems, VLSI, and low-power VLSI.



S. G. Nayak is currently working as a Professor and Head of the Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal. He received his B.E. degree in Electronics and Communication Engineering, M.Tech. degree in Biomedical Engineering, and Ph.D. degree in Electrical and Electronics Engineering. His research interests include digital systems and processor architecture design and applications. Further information is available on his homepage: <https://manipal.edu/mit/departement-faculty/faculty-list/g-subramanya-nayak>.



P. B. Nempu is presently working as an Assistant Professor at the Department of Electrical and Electronics Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, Karnataka, India. He received B.E. degree in Electrical and Electronics Engineering from Moodlakatte Institute of Technology, Kundapura, India and M.Tech. in Energy Management, auditing and lighting from Manipal Institute of Technology, India respectively. He obtained Ph.D. from Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. His research areas include renewable energy based distributed generation systems and power electronics.



© 2023 by the authors. Licensee IUST, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license (<https://creativecommons.org/licenses/by-nc/4.0/>).