On Designing Efficient Codecs for Bus-Invert Berger Code for Fully Asymmetric Communication

Stanisław J. Piestrak, Sebastien Pillement, and Olivier Sentieys

***Abstract*—Berger-invert code is a coding scheme proposed recently to protect communication channels against all asymmetric errors and to decrease power consumption. This paper proposes a modified encoding scheme of Berger-invert codes and a new design of encoding/decoding circuitry (a codec). Implementation results prove that the new approach reduces the codec area and power consumption respectively by up to 30% and 48% for a 32-bit bus and decreases the error rate by up to about 35% for a 6-bit bus.**

***Index Terms*—asymmetric channel, Berger codes, bus-invert, self-checking design, unidirectional error detection.**

I. Introduction

In modern VLSI circuits, bandwidth of data transfers between internal blocks have increased considerably. Interconnect wires and the associated circuitry are responsible for a high fraction of the energy consumption of an integrated circuit, which can reach up to 50% [1]. The possibility of increasing the bandwidth of interconnections is limited by the noise induced by simultaneous switchings. Several techniques that could be applied at various levels have been suggested to reduce noise and wire power consumption in interconnections. Those which concern research presented here include: bus-invert coding used for low power [2], [3] or noise reduction [4], low-weight coding used for noise reduction [4], [5] or for low power [6], and reduction of the voltage swing of the signal on the wire [7].

Unfortunately, some performance improvement and power reduction techniques (e.g., [7]) involve reduced noise margin that might result in increased error rate. Also, steadily shrinking transistor sizes and decreasing power supply voltage result in increasing reliability problems related to growing sensitivity to cosmic radiation [8]. This is because cosmic particles with less energy are more likely to flip memory bits or cause transient faults resulting in errors in low voltage circuits. Clearly, to ensure dependable operation of communication channels in low power VLSI circuits, the use of some concurrent error detecting or/and correcting techniques has become mandatory.

A binary asymmetric channel is a model of a communication system in which the error probability e.g. from 1 to 0 is much higher than the error probability from 0 to 1. Besides optical communication (a photon can vanish but can never be created), asymmetric errors have been observed in recent volatile memories like flash ROM memories [9], dynamic RAM memories [10], and various electronic devices designed for defect tolerance, soft error protection or leakage reduction [11]–[15]. Thus, the asymmetric error model is more appropriate for these systems. To deal with asymmetric errors, various error detecting or error correcting codes have been proposed [16]–[20], that generally require less redundancy and simpler encoding/decoding algorithms than equivalent multiple-bit symmetric error control codes.

The low-weight coding was considered for low power [6] and noise reduction [5], both applicable for a class of parallel terminated buses with pull-up terminators (e.g., Rambus), wherein the goal was to have as few 1's as possible, because only the 1 values are directly dissipative. On the other hand, bus-invert coding [3] was used to reduce power consumption, mainly through reducing the number of transitions, that required looking at the two consecutive bus transfers however. Recently, Huang [21] proposed to combine the low-weight bus-invert coding and error detecting Berger codes. The resulting *Berger-invert (BGI) code* was suggested not only to protect data transmitted over asymmetric channels but also to reduce the error rate and power consumption at the same time.

Here we shall propose a new modified BGI code whose encoding/decoding circuitry (a codec) is significantly simpler, and such that for various bus widths, the number of extra bus lines can be reduced by one. As a result, the area, power consumption, and error rate could significantly be reduced.

The remainder of the paper is organized as follows. In Section 2, the basic properties of coding techniques used here are presented. In Section 3, we analyze the properties of BGI encoding and propose a new BGI encoding method and a codec. Section 4 presents some simulation results and comparison regarding error rates, power consumption, and area of codec circuitry. Finally in Section 5, the main results of this paper are summarized.

II. Preliminaries

In this section, we present the basic properties of bus-invert, Berger, and Berger-invert coding techniques.

We shall use the following notation summarized in Table I.

A. Coding for Low Power and Noise Reduction

Bus-invert (BI) coding is a method intended to reduce the power consumption or switching noise in a bus [2]–[4]. It uses one extra bus line, called *bus-invert (BI)*, to inform the receiver side whether a current pattern is inverted or not. The signal BI can be generated as a function of either the *Hamming weight* of

S. J. Piestrak is with IRISA, 22300 Lannion, France, on leave from Univ. of Metz, 57070 Metz, France. O. Sentieys and S. Pillement are with Univ. of Rennes 1/IRISA, 22300 Lannion, France; email: piestrak@univ-metz.fr, Sebastien.Pillement@irisa.fr, Olivier.Sentieys@irisa.fr.

TABLE I
NOTATIONS AND DEFINITIONS APPLIED IN THIS PAPER

D	a data vector
C	check bits of D
I	the width of data vector D
K	the number of Berger code check bits, $K = \lceil \log_2(I+1) \rceil$
$\overline{X}$	bit-wise inversion of a binary vector X
$w(D)$	the number of 1's in D (the Hamming weight of D)
$w_0(D)$	the number of 0's in D
BI	the bus-invert bit
C_n^k	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$, k-combinations of size n
e	bit error rate of each bit transmitted over the bus
E	error rate of a total codeword in transmission

a pattern $w(D)$ or the *Hamming distance* between the present pattern and the last BI-coded pattern of the bus, including BI. If the Hamming distance (i.e., the number of bits in which two consecutive patterns differ and hence result in a transition) is larger than $I/2$, then $BI = 1$ and the present pattern is transmitted with each bit inverted (denoted $\overline{D}$); otherwise the pattern is left unchanged and $BI = 0$. The decoder needs only to remove the sign bit and re-invert the others when $BI = 1$; when $BI = 0$, the decoder needs only to remove it.

Note: only the first case of BI coding was taken into account in [21] and is of our interest here. It can be considered a special case of *limited-weight code* or *low-weight coding*, the concept suggested as a straightforward encoding strategy for noise reduction [4] and power minimization [6]: in either case, a code with few 1's must be used. By definition, an M-limited-weight code uses codewords with the Hamming weight upper bounded by M. For an $(I+1)$-bit BI-encoded data, $M = \lfloor (I+1)/2 \rfloor$, where $\lfloor A \rfloor$ is the integer part of A.

B. Berger Codes

A *unidirectional error* is a multiple error such that all erroneous bits are of either $0 \to 1$ or $1 \to 0$ type, but not both at the same time. If the probability of unidirectional $\overline{z} \to z$ errors is extremely small compared to unidirectional $z \to \overline{z}$ errors, $z \in \{0, 1\}$, the errors are called *asymmetric* $z \to \overline{z}$ errors. Notice that if a code detects all asymmetric errors, it also detects all unidirectional errors [16].

Berger codes [16] are the optimal systematic unordered codes capable of detecting asymmetric and unidirectional errors of any multiplicity. Let I denote the number of data bits (the width of an unprotected bus). The *Berger code* [16] has two different encoding schemes of its $K = \lceil \log_2(I+1) \rceil$ check bits $C = (c_{K-1}, \ldots, c_1, c_0)$: (i) the check part of the B_0-type encoding scheme is $w_0(D)$, the binary representation of the number of 0's in the information part; and (ii) the check part of the B_1-type encoding scheme is $\overline{w(D)}$, the bit-by-bit complemented (1's complement) number of 1's in the information part. These two encoding schemes are equivalent and optimal in the sense that either of them uses the same number of check bits and detects all asymmetric and unidirectional errors, and that no other systematic code capable of detecting the same class of errors uses fewer bits. Notice that they are identical for $I = 2^K - 1$, because $w_0(D) = \overline{w(D)}$. In this paper, we shall use mainly the B_1-type encoding scheme.

TABLE II
BGI ENCODING FOR $I = 5$ FROM [21]

$w(D)$		$w_0(D)$	$w_0(D)+1$		BI	C
0	0 0 0	5	6	**1 1 0**	0	1 1 0
1	0 0 1	4	5	**1 0 1**	0	1 0 1
2	0 1 0	3	4	**1 0 0**	0	1 0 0
3	**0 1 1**	2	3	0 1 1	1	0 1 1
4	**1 0 0**	1	2	0 1 0	1	1 0 0
5	**1 0 1**	0	1	0 0 1	1	1 0 1

C. Berger-Invert Codes

The *Berger-invert (BGI) code*, proposed in [21], combines the BI encoding (used to reduce power consumption) and the Berger encoding (used to detect asymmetric errors). First, the bus-invert signal BI is determined according to

$$BI = \begin{cases} 0 & \text{if} \quad w(D) < w_0(D) \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Then, the BGI check part is generated according to

$$C = \begin{cases} w_0(D) + 1 & \text{if} \quad BI = 0 \\ w(D) & \text{if} \quad BI = 1. \end{cases} \tag{2}$$

For $BI = 0$, $w_0(D)$ is incremented by 1 to take into account $BI = 0$, that is also protected by BGI check part.

Table II shows a sample BGI encoding for $I = 5$ from [21]. The binary vectors marked in bold in the first and third columns are those selected as BGI check parts.

III. DESIGN OF CODECS FOR BERGER-INVERT CODES

In this section, we shall first present the codec by Huang [21] and then propose the new one for a modified BGI.

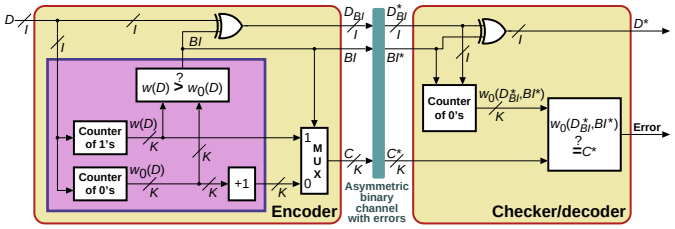

Fig. 1. BGI codec for $2^{K-1} - 1 \le I \le 2^K - 2$ from [21].

A. Previous Design

The encoder of Fig. 1 works as follows. First, the numbers of 1's and 0's in the data D are counted using two separate circuits (each can be built using e.g. $I - K$ full-adders and a few half-adders [22]). Next, their outputs are compared by the comparator of integers to determine the signal BI. Finally, two alternative check parts are generated and the appropriate one is selected by a MUX controlled by BI.

Because any input signal at the receiver side can be affected by $1 \to 0$ transmission errors, any such a signal will be denoted by an asterisk (D^*_{BI}, BI^*, C^*). The decoder inverts received data D^*_{BI} if $BI^* = 1$. The checker recomputes the check part $w_0(D^*_{BI}, BI^*)$, compares it against the received check part C^*, and activates the error signal if $w_0(D^*_{BI}, BI^*) > C^*$ (this inequality holds for any $1 \to 0$ transmission errors).

B. New BGI Code and Codec

Part of the BGI encoder by Huang, marked in Fig. 1, can be significantly simplified due to the following observations.

First, a counter of 1's suffices to realism the functions of two counters (of 1's and 0's), because $w_0(D)$ can be easily obtained using a counter of 1's, by taking advantage of 2's complement arithmetic properties of K-bit integers. Because $w(D) + w_0(D) = I$, therefore

$$w_0(D) = I - w(D) = \left[I + \overline{w(D)} + 1\right] \bmod 2^K, \quad (3)$$

i.e., $w_0(D)$ is the sum of $\overline{w(D)}$ and the constant $I+1$ with the carry out bit of weight 2^K discarded, the latter formally denoted as $\bmod 2^K$ operation.

Second, according to (2), both $w(D)$ and $w_0(D)$ are needed to generate BI. However, once I is known, we can use the following inequality

$$BI = \begin{cases} 0 & \text{if} \quad w(D) \le I/2 \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

The digital majority voter that implements (4) is significantly less complex than a comparator of integers [23]. It can also be implemented as an analog circuit [5], although with some limitations [3].

Now we shall show that using B_1-type Berger encoding would not only result in a simpler encoder, but that it would also allow to reduce the number of bus lines by one for all I but $I = 2^K - 1$ (in Huang's design, such a possibility is suggested for $I = 2^K - 2$ only).

In our design, the BGI check part C is generated as follows. If $w(D) \le I/2$, then $BI = 0$ and $C = \overline{w(D)}$, which is a 'classic' B_1-type Berger encoding. If $w(D) > I/2$, then $BI = 1$ and $C = w_0(D) + 1$ (the transmitted vector contains $w_0(D)$ 1's resulting from complementation of 0's and $BI = 1$. Due to (3), C can be written as

$$\begin{aligned} C = w_0(D) + 1 &= \left[I + \overline{w(D)} + 2\right] \bmod 2^K \\ &= \left[w(D) + (2^K - I - 2)\right] \bmod 2^K. \end{aligned} \quad (5)$$

For a given K, we shall consider two cases separately: (i) $I = 2^K - 1$; and (ii) $I \ne 2^K - 1$ (i.e., $2^K \le I \le 2^K - 2$).

(i) For $I = 2^K - 1$, all $2^{K-1} + 1$ K-bit combinations resulting from inverting the binary encodings from 0 to $(I+1)/2 = 2^{K-1}$, i.e., $\underbrace{(11...11)}_{K\ 1's}$, (11...10), ..., (01...11), are used as BGI check parts. By substituting I with $2^K - 1$ in (5) for $BI = 1$, we get the following general equation for the BGI check part

$$C = \begin{cases} \overline{w(D)} & \text{if} \quad BI = 0; \\ \left[w(D) + 2^K - 1\right] \bmod 2^K & \text{if} \quad BI = 1. \end{cases} \quad (6)$$

(ii) For $I \ne 2^K - 1$, all combinations resulting from inverting the binary encodings from 0 to $\lfloor I/2 \rfloor$ are used as BGI check parts. Their number is upper bounded by 2^{K-1} for $I = 2^K - 2$, because in this extreme case they are: (11...11), (11...10), ..., (10...00). Clearly, the most significant bit (MSB) carries a constant binary signal 1 that can be ignored for any $I \ne 2^K - 1$. Therefore, for $I \ne 2^K - 1$, the BGI check part can be generated according to

$$C = \begin{cases} \overline{w(D)} \bmod 2^{K-1} & \text{if } BI = 0; \\ \left[w(D) + 2^K - I - 2\right] \bmod 2^{K-1} & \text{if } BI = 1. \end{cases} \quad (7)$$

Equations (6) and (7) can be presented in the following unified form

$$C = \begin{cases} \overline{w(D)} \bmod 2^{K'} & \text{if} \quad BI = 0; \\ [w(D) + Const] \bmod 2^{K'} & \text{if} \quad BI = 1, \end{cases} \quad (8)$$

where: $K' = K$ and $Const = 2^K - 1$ if $I = 2^K - 1$, and $K' = K - 1$ and $Const = 2^K - I - 2$ otherwise.

As for the checker at the receiver's side, the following minor hardware saving modifications are also observed.

- Because for all I but $I = 2^K - 1$, the counter of 1's with the total of $K' = K - 1$ rather than $K' = K$ outputs is used, the comparator has two inputs less than in [21].
- The equality comparator is used, because it is simpler than the comparator of integers, suggested in [21].

All the above observations have led us to propose a new codec architecture shown in Fig. 2(A).

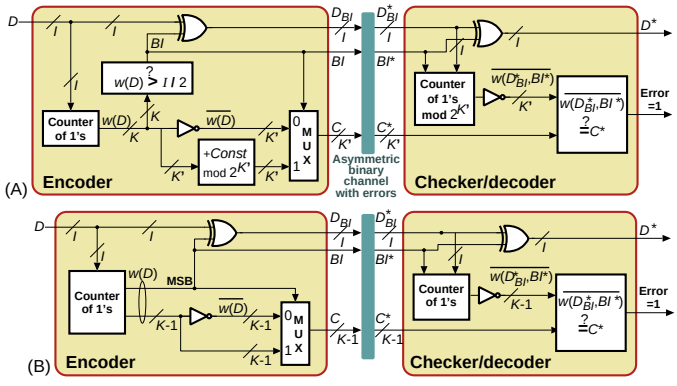

Fig. 2. New BGI codec: (A) for any I; and (B) for $I = 2^K - 2$.

TABLE III
NEW BGI ENCODING FOR $I = 5$

$w(D)$		$w(D)+1$		$\overline{w(D)}$	BI	C	
0	0 0 0	1	0 0 1	**1 1 1**	0	*1*	1 1
1	0 0 1	2	0 1 0	**1 1 0**	0	*1*	1 0
2	0 1 0	3	0 1 1	**1 0 1**	0	*1*	0 1
3	0 1 1	4	**1 0 0**	0 1 1	1	*1*	0 0
4	1 0 0	5	**1 0 1**	0 1 0	1	*1*	0 1
5	1 0 1	6	**1 1 0**	0 0 1	1	*1*	1 0

Table III shows the new BGI encoding for $I = 5$ derived from (8). As before, the binary vectors selected as BGI check parts are marked in bold. Two following observations, that can be shown valid for any $I = 2^K - 3$, allow to save some hardware: (i) because the MSB of the check part has the constant value for all data patterns ($c_2 = 1$), it can be removed to save one bus line compared to Huang's encoding (see the last column of Table II); and (ii) because the MSB of $w(D) + 1$ equals to BI, no comparator to implement (4) is needed.

TABLE IV
NEW BGI ENCODING FOR $I = 6$

$w(D)$		$\overline{w(D)}$	BI	C	
0	0 0 0	**1 1 1**	0	*1*	1 1
1	0 0 1	**1 1 0**	0	*1*	1 0
2	0 1 0	**1 0 1**	0	*1*	0 1
3	0 1 1	**1 0 0**	0	*1*	0 0
4	**1 0 0**	0 1 0	1	*1*	0 0
5	**1 0 1**	0 0 1	1	*1*	0 1
6	**1 1 0**	0 0 1	1	*1*	1 0

The case of $I = 2^K - 2$, for which $K' = K - 1$ and $Const = 2^K - I - 2 = 0$, also deserves special consideration. The inspection of sample BGI encoding shown in Table IV reveals that BI is identical to the MSB of $w(D)$ which means that the comparator of $w(D)$ against the constant $I/2$ is not needed. The resulting simplified version of the BGI codec for $I = 2^K - 2$ is shown in Fig. 2(B). (Actually, BI is identical to the MSB of $w(D)$ for $I = 2^K - 1$ as well, so that the comparator that implements (4) is also not needed.)

IV. PARAMETER ESTIMATION AND COMPARISON

Here we shall show some simulation results to prove that the new BGI encoding results not only in less hardware but also leads to reduced error rate and less power consumption.

A. Error Rate Evaluation

The probabilistic model used is an I-bit channel on which the 2^I data patterns are equally probable, i.e., the probability of occurrence of any I-bit data pattern is $1/2^I$. The probabilistic distribution of bit patterns that occur on extra lines (BI and Berger code check bits) depends directly on the probabilities of data patterns to which they are attached. For an asymmetric channel: (i) the probability of a $1 \to 0$ error on each data and check bit equals to e and it is independent for each bit, and (ii) the probability of a $0 \to 1$ error is 0. Consequently, the probability that a given bit has no error is $p = 1 - e$.

The error rates E and E_{BI} of simple and BI-coded I-bit data can be computed using the following equations taken from [21] (Eqns (1), (2), and (8)) with minor corrections ($i = 0$ is included now) and modifications:

$$E = \frac{1}{2^I} \sum_{i=0}^{I} C_I^i \cdot (1 - p^i) \approx \frac{Ie}{2} \quad (\text{if } Ie \ll 1), \tag{9}$$

$$E_{BI} = \frac{1}{2^I} \Big(\sum_{i=0}^{\lceil I/2 \rceil} C_I^i \cdot (1 - p^i) + \sum_{\lceil I/2 \rceil + 1}^{I} C_I^i \cdot (1 - p^{I-i+1}) \Big). \tag{10}$$

The following general formula can be used to determine the error rate of any Berger-encoded I-bit data

$$E_{BC} = \frac{1}{2^I} \sum_{i=0}^{I} C_I^i \cdot (1 - p^{i + w(C(i))}), \tag{11}$$

where $w(C(i))$ is the weight of the check part $C(i)$ that corresponds to data of weight i, whatever encoding is used. In particular, for the BGI encoding proposed here, (11) becomes

$$E_{BGI_N} = \frac{1}{2^I} \Big(\sum_{i=0}^{\lfloor I/2 \rfloor} C_I^i \cdot (1 - p^{i + w(2^K - 1 - i \bmod 2^{K'})}) + \sum_{i=\lfloor I/2 \rfloor + 1}^{I} C_I^i \cdot (1 - p^{I-i+1+w(C(i+Const))}) \Big). \tag{12}$$

Note: henceforth all parameters of the new and Huang's designs will be denoted with indices N and H, respectively.

TABLE V
ERROR RATE ESTIMATIONS FOR $I = \{2^K - 3, 2^K - 2\}$

K I	e	E	E_{BI}	E_{BC}	E_{BGI_N}	E_{Red}
3 5	E-06	2.50E-06	2.06E-06	3.81E-06	2.78E-06	27.0
	E-03	2.50E-03	2.06E-03	3.81E-03	2.78E-03	27.0
4 13	E-06	6.50E-06	5.53E-06	8.46E-06	6.46E-06	23.7
	E-03	6.48E-03	5.52E-03	8.43E-03	6.44E-03	23.6
5 29	E-06	1.45E-05	1.28E-05	1.70E-05	1.40E-05	18.1
	E-03	1.44E-02	1.28E-02	1.69E-02	1.39E-02	18.0
3 6	E-06	3.00E-06	2.41E-06	4.41E-06	2.88E-06	34.7
	E-03	3.00E-03	2.40E-03	4.40E-03	2.87E-03	34.8
4 14	E-06	7.00E-06	5.93E-06	8.98E-06	6.63E-06	26.2
	E-03	6.98E-03	5.91E-03	8.94E-03	6.61E-03	26.1
5 30	E-06	1.50E-05	1.33E-05	1.75E-05	1.42E-05	18.9
	E-03	1.49E-02	1.32E-02	1.74E-02	1.41E-02	19.0

The error rate estimations provided in Table V were computed according to (9)–(12) (to note that all error rates for $e = E$-10 have decimal parts identical to those for $e = E$-06). They show that in the asymmetric channel the BI coding alone results in up to 11%–20% error rate reduction compared to simple data transmission, although it does not allow to detect any transmission errors. All $1 \to 0$ transmission errors are detected by the Berger coding alone, although at the 17%–52.5% increase of the error rate E_{BC}, owing to extra check bits that are also susceptible to errors. It is seen that the BGI encoding is the best solution: it not only lowers the error rate E_{BGI_N} even below the level E of an unprotected channel (except for $I = 5$), but also allows to detect all $1 \to 0$ transmission errors. The last column shows that the error rate of the BGI encoding compared to Berger coding alone is still reduced by $E_{Red} = 100 \cdot (E_{BC} - E_{BGI_N})/E_{BC}$, which ranges from 18% to 34.7% at no cost of extra bus lines. The error rate reduction reported in Table II of [21] ranges from 12.3% (for $I = 30$) to 21.3% (for $I = 6$) which is less than here.

B. Implementation Results

The codecs by Huang [21] and the new ones were synthesized for selected bus widths using the Synopsys® Design Compiler® suite of tools for a STMicroelectronics 90nm 1.0V technology. Table VI presents the area estimations of two versions of encoders, decoders, and complete codecs (encoder + decoder). The area savings of our circuits compared to their Huang's versions are calculated according to $A_{Red} = 100 \cdot (A_{BGI_H} - A_{BGI_N})/A_{BGI_H}$ for each circuit considered. The figures enclosed confirm the advantages of all our modifications suggested for both encoders and decoders. The overall area reduction for codecs ranges from over 12% for small I to about 40% for large I.

TABLE VI
AREA ESTIMATION (IN μm^2) FOR VARIOUS BUS' WIDTHS I

K	I	A_{BGI_H}		A_{BGI_N}		A_{Red}[%]		
		Enc	Dec	Enc	Dec	Enc	Dec	Codec
3	6	317	285	283	244	10.72	14.31	12.51
4	7	406	334	347	285	14.59	14.47	14.53
	10	790	549	561	497	29.02	9.40	19.21
	14	1172	798	809	718	30.99	10.03	20.51
5	16	2613	1482	1503	938	42.50	36.66	39.58
	30	5132	2897	2879	1907	43.90	34.17	39.04
6	32	5958	3652	3651	3073	38.70	15.85	30.01

TABLE VII
POWER CONSUMPTION ESTIMATION (IN μW) FOR VARIOUS BUS' WIDTHS

K	I	P_{Simple}	P_{BGI_H}	P_{BGI_N}	P_{Red}[%]
3	6	72.88	153.20	144.08	6.2
4	7	90.57	196.78	191.51	2.8
	10	129.87	307.27	275.27	10.2
	14	175.76	487.22	415.36	14.8
5	16	203.93	1093.33	758.60	30.6
	30	380.16	3925.19	2059.03	47.6
6	32	403.70	4419.37	2296.49	48.1

To evaluate accurately the power consumption, some simulation experiments were performed on BGI-protected circuits (codec+bus) for various I, assuming the Metal-4 layer and 2-mm wire length of the bus. We use the classical Monte Carlo approach for power estimation in which we apply 10000 random input patterns in the simulations. This number is more than enough to guarantee that the measured power has converged close enough to the true average power [24]. As timing constraints for synthesis are set for 100 MHz frequency, we used the same constraints that were easily met for all circuits considered. The activity of the circuits was generated by Mentor Graphics® ModelSim SE Plus 6.4B simulation tools after synthesis for the cells in the nominal library (the most accurate approach available without actually generating the layout). The estimations of the power consumption of codecs and wires were obtained respectively using PrimeTime® B-2008.O6 version from Synopsys® and the high-level estimation CAD tool *Interconnect Explorer* [25].

The statistical results shown in Table VII take into account the bus wires and the codec except for a simple bus, which has no codec. Power consumed by both versions of BGI-encoded buses (P_{BGI_H} and P_{BGI_N}) is at least twice as large as for a simple bus. However, compared to Huang's designs, the new codecs reduce power consumption by $P_{Red} = 100 \cdot (P_{BGI_H} - P_{BGI_N})/P_{BGI_H}$% and their superiority grows with I.

V. CONCLUSION

The Berger-invert (BGI) code is a coding scheme proposed recently to protect communication channels against all asymmetric errors. It not only ensures that all transmission errors are detectable at the receiver side but also enjoys the advantages of decreased error rate and reduced power consumption. In this paper, a slightly modified BGI encoding scheme has been proposed that offers the same error protection as Huang's BGI encoding, but also reduces the error rate from 18 to about 35%. Owing to new encoding, one bus line is saved for all data widths I but for $I = 2^K - 1$ and the codecs (encoders and decoders/checkers) for these codes are significantly simpler. Implementation results of the sample bus systems confirm the reduction of both the area (from 12.5 to about 40%) and the power consumption (up to 48%). They also reveal that the cost of protecting the communication bus against errors, measured by the power consumption, is generally very high (at least two-fold).

REFERENCES

[1] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," *Proc. IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.

[2] r R. J. Fletcher, "Integrated circuit having outputs configured for reduced state changes," U.S. Patent no. 4 667 337, May 1987.

[3] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 49–58, Mar. 1995.

[4] J. F. Tabor, "Noise reduction using low weight and constant weight coding techniques," M.Sc. Thesis, Tech. Rep. 1232, Artif. Intell. Lab., MIT, Cambridge, MA, 1990.

[5] K. Nakamura, and M. A. Horowitz, "A 50% noise reduction interface using low-weight coding," in *Digest of Technical Papers of Symp. on VLSI Circuits*, 1996, pp. 144–145.

[6] M. R. Stan and W. P. Burleson, "Coding a terminated bus for low power," in *Proc. Fifth Great Lakes Symp. on VLSI (GLSVLSI'95)*, 1995, pp. 70–73.

[7] H. Zhang, V. George, and J. M. Rabaey, "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Trans. VLSI Systems*, vol. 8, pp. 264–272, June 2000.

[8] C. R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device and Materials Reliab.*, vol. 5, pp. 305–316, Sept. 2005.

[9] Y. Cassuto *et al.*, "Codes for limited magnitude asymmetric errors in flash memories," US Patent 20080168320, July 2008.

[10] P. G. Emma, R. K. Montoye, and W. R. Reohr, "Dynamic memory architecture employing passive expiration of data," US Patent 7 290 203, Oct. 30, 2007.

[11] H. T. Weaver, W. T. Corbett, and J. M. Pimbley, "Soft error protection using asymmetric response latches," *IEEE Trans. Electron. Devices*, vol. 38, pp. 1555–1557, June 1991.

[12] N. Azizi, A. Moshovos, and F. N. Najm, "Low-leakage asymmetric-cell SRAM," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, 2002, pp. 48–51.

[13] H. Deogun *et al.*, "Leakage and crosstalk-aware bus encoding for total power reduction," in *Proc. 41th ACM/IEEE Design Automation Conf. (DAC)*, 2004, pp. 779–782.

[14] F. Nemati, M. R. Kasnavi, and R. H. Igehy, "Memory cell error recovery," US Patent 6785169, Aug. 2004.

[15] A. Abdollahi, F. Fallah, and M. Pedram, "An effective power mode transition technique in MTCMOS circuits," in *Proc. of Design Autom. Conf. (DAC)*, 2005, pp. 37–42.

[16] J. M. Berger, "A note on error detection codes for asymmetric binary channels," *Inform. Contr.*, vol. 4, pp. 68–73, Mar. 1961.

[17] C. V. Freiman, "Optimal error detection codes for completely asymmetric binary channels," *Inform. Contr.*, vol. 5, pp. 64–71, Mar. 1962.

[18] J. E. Smith, "The design of totally self-checking check circuits for a class of unordered codes," *J. Des. Autom. Fault-Tolerant Comput.*, vol. 2, pp. 321–342, Oct. 1977.

[19] J. E. Smith, "On separable unordered codes," *IEEE Trans. Comput.*, vol. C-33, pp. 741–743, Aug. 1984.

[20] J. H. Weber, "Asymptotic results on codes for symmetric, unidirectional, and asymmetric error control," *IEEE Trans. Inform. Theory*, vol. 40, pp. 2073–2075, Nov. 1994.

[21] T.-C. Huang, "A low-power dependable Berger code for fully asymmetric communication," *IEEE Commun. Lett.*, vol. 12, pp. 773–775, Oct. 2008.

[22] S. J. Piestrak, *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*, Scientific Papers of Inst. of Techn. Cybern. of Techn. Univ. of Wrocław, no. 92, Ser.: Monographs no. 24, Wrocław 1995.

[23] S. J. Piestrak, "Efficient Hamming weight comparators of binary vectors," *Electron. Lett.*, vol. 43, pp. 611–612, 24 May 2007.

[24] R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte Carlo approach for power estimation," *IEEE Trans. Very Large Scale Integr. (VLSI) Systems*, vol. 1, no. 1, pp. 63–71, Mar. 1993.

[25] A. Courtay *et al.*, "High-level interconnect delay and power estimation," *J. of Low Power Electron.*, vol. 4, pp. 21–33, Apr. 2008.

This is the previous version:

TCAS-II 6788

Stanislaw J. Piestrak, Sebastien Pillement, and Olivier Sentieys

"On Designing Efficient Codecs for Bus-Invert Berger Code for Fully Asymmetric Communication"

On Designing Efficient Codecs for Bus-Invert Berger Code for Fully Asymmetric Communication

Stanisław J. Piestrak, Sebastien Pillement, and Olivier Sentieys

***Abstract*—Berger-invert code is a coding scheme proposed recently to protect communication channels against all asymmetric errors and to decrease power consumption. This paper proposes a slightly modified encoding scheme of Berger-invert codes and a new design of encoding/decoding circuitry (a codec). Implementation results prove that the new approach leads to significant hardware savings and results in reduced error rate and power consumption.**

***Index Terms*—Berger codes, bus-invert, asymmetric channel, self-checking design, unidirectional error detection.**

I. Introduction

IN modern VLSI circuits, high performance, power consumption and dependable operation belong to the most important design and implementation aspects. Communication systems are one class of basic blocks used in any VLSI circuit wherein these issues are particularly critical. On one hand, the possibility of increasing the bandwidth of interconnections is limited by the noise induced by simultaneous switchings. On the other hand, interconnect wires and the associated circuitry are responsible for a high fraction of the energy consumption of an integrated circuit, which can reach up to 50%. Several techniques that could be applied at various levels have been suggested to reduce noise and wire power consumption in interconnections. Those which concern research presented here include: bus-invert coding used for low power [1], [2] or noise reduction [3], low-weight coding used for noise reduction [3] (called *starvation coding*), [4] or for low power [5], and reduction of the voltage swing of the signal on the wire [6].

Unfortunately, some of performance improvement and power reduction techniques (e.g. [6]) involve reduced noise margin which might result in increased error rate. Also, steadily shrinking transistor sizes and decreasing power supply voltage result in increasing reliability problems related to growing sensitivity to cosmic radiation [7]. This is because cosmic particles with less energy are more likely to flip memory bits or cause transient faults resulting in errors in low voltage circuits. Clearly, to ensure dependable operation of communication channels in low power VLSI circuits, the use of some concurrent error detecting or/and correcting techniques has become mandatory.

A binary asymmetric channel is a model of a communication system in which the error probability e.g. from 1 to 0 is much higher than the error probability from 0 to 1. Besides optical communication (a photon can vanish but can never be created), asymmetric errors have been observed in recent volatile memories like flash ROM memories [8], dynamic RAM memories [9], and various electronic devices designed for defect tolerance, soft error protection or leakage reduction [10]–[14]. Thus, the asymmetric error model is more appropriate for these systems. To deal with asymmetric errors, various error detecting or error correcting codes have been proposed [15]–[19], which generally require less redundancy and simpler encoding/decoding algorithms than equivalent multiple-bit symmetric error control codes.

The low-weight coding was considered for low power [5] and noise reduction [4], both applicable for a class of parallel terminated buses with pull-up terminators (e.g. Rambus), wherein the goal was to have as few 1's as possible, since only the 1 values are directly dissipative. On the other hand, bus invert coding [2] was used to reduce power consumption mainly through reducing the number of transitions, which required looking at the two consecutive bus transfers however. Recently, Huang [20] proposed to combine the low-weight bus invert coding and error detecting Berger codes. The resulting *Berger invert (BGI) code* was suggested not only to protect data transmitted over asymmetric channels but also to reduce the error rate and power consumption at the same time.

Here we shall propose a new modified BGI code whose encoding/decoding circuitry (a codec) is significantly simpler, and such that for various bus widths, the number of extra bus lines can be reduced by one. As a result, the area, power consumption, and error rate could significantly be reduced.

The remainder of the paper is organised as follows. In Section 2, the basic properties of bus-invert and Berger codes are presented. In Section 3, we analyse the properties of BGI encoding and propose a new BGI encoding method and a codec. Section 4 presents some experimental results and comparison regarding error rates, power consumption, and area of encoding and checking/decoding circuitry. Finally in Section 5, the main results of this paper are summarised.

II. Preliminaries

In this section we present the basic properties of bus invert coding and of Berger codes.

We shall use the following notation summarised in Table I.

A. Coding for Low Power and Noise Reduction

Bus-invert (BI) coding is a method intended to reduce the power consumption or switching noise in a bus [1]–[3]. It uses

The authors are with IRISA/INRIA — University of Rennes 1, ENSSAT, 6, rue de Kerampont, 22300 Lannion, France; email: {name}@irisa.fr.

S. J. Piestrak is on leave from LICM, University of Metz, 7 rue Marconi, 57070 Metz, France

TABLE I
NOTATIONS AND DEFINITIONS APPLIED IN THIS PAPER

D	a data vector
C	check bits of D
I	the width of data vector D
K	the number of Berger code check bits, $K = \lceil \log_2(I+1) \rceil$
$\overline{X}$	bit-wise inversion of a binary vector X
$w(D)$	the number of 1's in D (the Hamming weight of D)
$w_0(D)$	the number of 0's in D
BI	the bus invert bit
C_n^k	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$, k-combinations of size n
e	bit error rate of each bit transmitted over the bus
E	error rate of a total codeword in transmission

one extra bus line, called *bus-invert (BI)*, to inform the receiver side whether a current pattern is inverted or not. The signal BI can be generated in two ways: as a function of either the *Hamming weight* of a pattern $w(D)$ or the *Hamming distance* between the present pattern and the last BI-coded pattern of the bus, including BI (the number of bits in which they differ and hence result in a transition). If the Hamming distance is larger than $I/2$ then $BI = 1$ and the present pattern is transmitted with each bit inverted (denoted $\overline{D}$); otherwise the pattern is left unchanged and $BI = 0$. The decoder needs only to remove the sign bit and re-invert the others when $BI = 1$; when $BI = 0$, the decoder needs only to remove it.

Note: only the first case of BI coding was taken into account in [20] and is of our interest here. It can be considered a special case of *limited-weight code* or *low-weight coding (LWC)*, the concept suggested as a straightforward encoding strategy for noise reduction [3] and power minimisation [5]: in either case a code with few 1's must be used. By definition an M-limited-weight code uses codewords with the Hamming weight upper bounded by M. For an $(I+1)$-bit BI-encoded data, $M = \lfloor (I+1)/2 \rfloor$, where $\lfloor A \rfloor$ is the integer part of A.

B. Berger Codes

A *unidirectional error* is a multiple error such that all erroneous bits are of either $0 \to 1$ or $1 \to 0$ type, but not both at the same time. If the probability of unidirectional $\overline{z} \to z$ errors is extremely small compared to unidirectional $z \to \overline{z}$ errors, $z \in \{0, 1\}$, the errors are called *asymmetric* $z \to \overline{z}$ errors. If a code detects all asymmetric errors, it also detects all unidirectional errors [15] (to note that similar does not hold for asymmetric error correcting codes however [19]).

Berger codes [15] are the optimal systematic unordered codes capable of detecting asymmetric and unidirectional errors of any multiplicity. Let I denote the number of data bits (the width of an unprotected bus). The *Berger code* [15] has two different encoding schemes of its $K = \lceil \log_2(I+1) \rceil$ check bits $C = (c_{K-1}, \ldots, c_1, c_0)$: (i) the B_0-type encoding scheme that uses $w_0(D)$, the binary representation of the number of 0's in the information part, as the check part; and (ii) the B_1-type encoding scheme that uses $\overline{w(D)}$, the bit-by-bit complemented (1's complement) number of 1's in the information part, as the check part. These two encoding schemes are equivalent and optimal in the sense that either uses the same number of check bits and detects all asymmetric and unidirectional errors and no other systematic code capable of detecting the same class of errors uses fewer bits. To note that they are identical for $I = 2^K - 1$, i.e. $w_0(D) = \overline{w(D)}$. In this paper, we shall use mainly the B_1-type encoding scheme.

TABLE II
BGI ENCODING FOR $I = 5$ FROM [20]

$w(D)$		$w_0(D)$	$w_0(D)+1$		BI	C
0	0 0 0	5	6	**1 1 0**	0	1 1 0
1	0 0 1	4	5	**1 0 1**	0	1 0 1
2	0 1 0	3	4	**1 0 0**	0	1 0 0
3	**0 1 1**	2	3	0 1 1	1	0 1 1
4	**1 0 0**	1	2	0 1 0	1	1 0 0
5	**1 0 1**	0	1	0 0 1	1	1 0 1

III. DESIGN OF CODECS FOR BERGER-INVERTED CODES

In this section we shall present the codec by Huang [20] and then propose the new one for a slightly modified BGI.

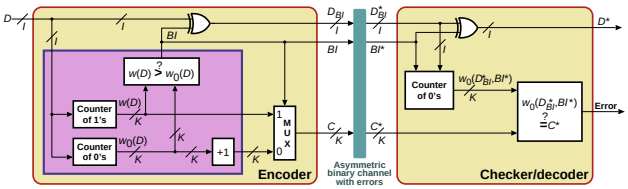

Fig. 1. BGI codec for $2^{K-1} - 1 \le I \le 2^K - 2$ from [20].

A. Previous Design

The encoder of Fig. 1 works as follows. First, the numbers of 1's and 0's in the data D are counted using two separate circuits (each can be built e.g. using $I - K$ FAs and a few HAs). Next, their outputs are compared by the comparator of integers to determine the bus invert signal

$$BI = \begin{cases} 0 & \text{if} \quad w(D) < w_0(D) \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Finally, two alternative check parts are generated and the appropriate one is selected by a MUX controlled by BI

$$C = \begin{cases} w_0(D) + 1 & \text{if} \quad BI = 0 \\ w(D) & \text{if} \quad BI = 1. \end{cases} \tag{2}$$

Since any input signal at the receiver side can be affected by $1 \to 0$ transmission errors, any such a signal will be denoted by an asterisk (D^*_{BI}, BI^*, C^*). The decoder inverts received data D^*_{BI} if $BI^* = 1$. The checker recomputes the check part $w_0(D^*_{BI}, BI^*)$, compares it against the received check part C^*, and activates the error signal if $w_0(D^*_{BI}, BI^*) > C^*$ (this inequality holds for any $1 \to 0$ transmission errors).

Table II shows a sample BGI encoding for $I = 5$ from [20]. The binary vectors marked in bold in the first and third columns are those selected as BGI check parts.

B. New BGI Code and Codec

Part of the BGI encoder by Huang, marked in Fig. 1, can be significantly simplified due to the following observations.

Firstly, the functions realised by two counters (of 1's and 0's) can be realised by a single circuit — a counter of 1's. This is because $w_0(D)$ can easily be obtained using a counter of 1's by taking advantage of 2's complement arithmetic properties of K-bit integers. Since $w(D) + w_0(D) = I$, therefore

$$w_0(D) = I - w(D) = \left[I + \overline{w(D)} + 1\right] \bmod 2^K, \quad (3)$$

i.e., $w_0(D)$ is the sum of $\overline{w(D)}$ and the constant $I+1$ with the carry out bit of weight 2^K discarded.

Secondly, to generate BI, unlike in (2) which requires both $w(D)$ and $w_0(D)$ to be known, once I is known, we can use the following inequality

$$BI = \begin{cases} 0 & \text{if} \quad w(D) \le I/2 \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

The digital majority voter that implements (4) is significantly less complex than a comparator of integers [22]. It can also be implemented as an analog circuit [4] with some limitations however [2].

Now we shall show that using B_1-type Berger encoding would not only result in a simpler encoder, but it would also allow to reduce the number of bus lines by one for all I but $I = 2^K - 1$ (in Huang's design, such a possibility is suggested for $I = 2^K - 2$ only).

In our design the BGI check part C is generated as follows. If $w(D) \le I/2$, then $BI = 0$ and $C = \overline{w(D)}$, which is a 'classic' B_1-type Berger encoding. If $w(D) > I/2$, then $BI = 1$ and $C = w_0(D) + 1$ (the transmitted vector consists of $w_0(D)$ complemented 0's and $BI = 1$) which, due to (3), can be written as

$$\begin{aligned} C = w_0(D) + 1 &= \left[I + \overline{w(D)} + 2\right] \bmod 2^K \\ &= \left[w(D) + (2^K - I - 2)\right] \bmod 2^K. \end{aligned} \quad (5)$$

For a given K, we shall consider two cases separately: (i) $I = 2^K - 1$; and (ii) $I \ne 2^K - 1$, i.e. for $2^K \le I \le 2^K - 2$.

(i) For $I = 2^K - 1$, the BGI check parts are all $2^{K-1}+1$ K-bit combinations resulting from inverting the binary encodings from 0 to $(I+1)/2 = 2^{K-1}$, i.e., $\underbrace{(11...11)}_{K\ 1's}$, (11...10), ..., (01...11). By substituting I with $2^K - 1$ in (5) for $BI = 1$, we get the following general equation for the BGI check part

$$C = \begin{cases} \overline{w(D)} & \text{if} \quad BI = 0; \\ \left[w(D) + 2^K - 1\right] \bmod 2^K & \text{if} \quad BI = 1. \end{cases} \quad (6)$$

(ii) For $I \ne 2^K - 1$, all combinations resulting from inverting the binary encodings from 0 to $\lfloor I/2 \rfloor$ are used as the BGI check parts C. Their number is upper-bounded by 2^{K-1} for $I = 2^K - 2$, i.e., in the extreme case they are: (11...11), (11...10), ..., (10...00). Clearly, the most significant bit (MSB) carries a constant binary signal 1 which hence can be ignored for any $I \ne 2^K - 1$. Therefore for $I \ne 2^K - 1$ the BGI check part can be generated according to

$$C = \begin{cases} \overline{w(D)} \bmod 2^{K-1} & \text{if } BI = 0; \\ \left[w(D) + 2^K - I - 2\right] \bmod 2^{K-1} & \text{if } BI = 1. \end{cases} \quad (7)$$

TABLE III
NEW BGI ENCODING FOR $I = 5$

$w(D)$		$w(D)+1$		$\overline{w(D)}$	BI	C	
0	0 0 0	1	0 0 1	**1 1 1**	0	*1*	1 1
1	0 0 1	2	0 1 0	**1 1 0**	0	*1*	1 0
2	0 1 0	3	0 1 1	**1 0 1**	0	*1*	0 1
3	0 1 1	4	**1 0 0**	0 1 1	1	*1*	0 0
4	1 0 0	5	**1 0 1**	0 1 0	1	*1*	0 1
5	1 0 1	6	**1 1 0**	0 0 1	1	*1*	1 0

Recall that the operation $\bmod 2^{K-1}$ relies on discarding the MSBs of weight 2^{K-1} and higher.

Equations (6) and (7) can be presented in the following unified form

$$C = \begin{cases} \overline{w(D)} \bmod 2^{K'} & \text{if} \quad BI = 0; \\ \left[w(D) + Const\right] \bmod 2^{K'} & \text{if} \quad BI = 1, \end{cases} \quad (8)$$

where: $K' = K$ and $Const = 2^K - 1$ if $I = 2^K - 1$, and otherwise $K' = K - 1$ and $Const = 2^K - I - 2$.

As for the checker at the receiver's side, the following minor hardware saving modifications are also observed.

- Since for all I but $I = 2^K - 1$, the counter of 1's with the total of $K' = K - 1$ rather than $K' = K$ outputs is used, the comparator has two inputs less than in [20].
- The equality comparator is used, since it is simpler than the comparator of integers, suggested in [20].

All the above observations have led us to propose a new codec architecture, shown in Fig. 2(A).

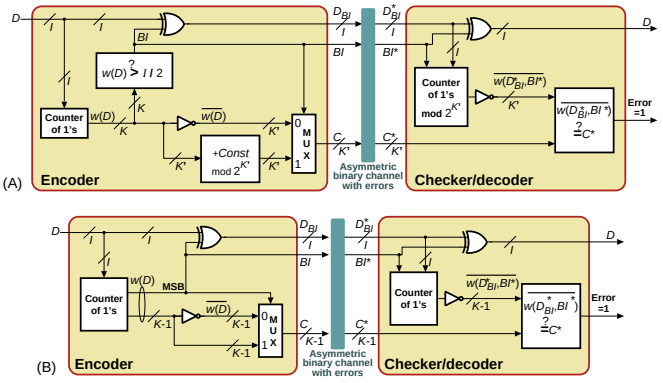

Fig. 2. New BGI codec: (A) for any I; and (B) for $I = 2^K - 2$.

Table III shows the new BGI encoding for $I = 5$, derived from (8). As before, the binary vectors selected as BGI check parts are marked in bold. Two following observations, which can be shown valid for any $I = 2^K - 3$, allow to save some hardware: (i) since the MSB of the check part has the constant value for all data patterns ($c_2 = 1$), it can be removed to save one bus line compared to Huang's encoding (see the last column of Table II); and (ii) since the MSB of $w(D) + 1$ equals to BI, no comparator to implement (4) is needed.

The case of $I = 2^K - 2$, for which $K' = K - 1$ and $Const = 2^K - I - 2 = 0$, also deserves special consideration. The inspection of sample BGI encoding shown in Table IV reveals that BI is identical to the MSB of $w(D)$, which means that the comparator of $w(D)$ against the constant $I/2$ is not

TABLE IV
NEW BGI ENCODING FOR $I = 6$

$w(D)$		$\overline{w(D)}$	BI	C	
0	0 0 0	**1 1 1**	0	*1*	1 1
1	0 0 1	**1 1 0**	0	*1*	1 0
2	0 1 0	**1 0 1**	0	*1*	0 1
3	0 1 1	**1 0 0**	0	*1*	0 0
4	**1 0 0**	0 1 0	1	*1*	0 0
5	**1 0 1**	0 0 1	1	*1*	0 1
6	**1 1 0**	0 0 1	1	*1*	1 0

needed. The resulting simplified version of the BGI codec for $I = 2^K - 2$ is shown in Fig. 2(B). (Actually, BI is identical to the MSB of $w(D)$ for $I = 2^K - 1$ as well, so that the comparator that implements (4) is also not needed.)

IV. PARAMETER ESTIMATION AND COMPARISON

Here we shall show some experimental results to prove that the new BGI encoding results not only in less hardware but also leads in reduced error rate and less power consumption.

A. Error Rate Evaluation

The probabilistic model used is an I-bit channel on which the 2^I data patterns are equally probable, i.e. the probability of occurrence of any I-bit data pattern is $1/2^I$. The probabilistic distribution of bit patterns that occur on extra lines (BI and Berger code check bits) depends directly on the probabilities of data patterns to which they are attached. Since the channel considered is asymmetric, we assume that: (i) the probability of a $1 \rightarrow 0$ error on each data and check bit equals to e and it is independent for each bit, and (ii) the probability of a $0 \rightarrow 1$ error is 0. Consequently, the probability that a given bit has no error is $p = 1 - e$.

The error rates E and E_{BI} of simple and BI-coded I-bit data can be computed using the following equations taken from [20] (Eqns (1), (2), and (8)) with minor corrections and modifications):

$$E = \frac{1}{2^I} \sum_{i=0}^{I} C_I^i \cdot (1 - p^i) \approx \frac{Ie}{2} \quad (\text{if } Ie \ll 1), \tag{9}$$

$$E_{BI} = \frac{1}{2^I} \Big(\sum_{i=0}^{\lceil I/2 \rceil} C_I^i \cdot (1 - p^i) + \sum_{\lceil I/2 \rceil + 1}^{I} C_I^i \cdot (1 - p^{I-i+1}) \Big). \tag{10}$$

The following general formula can be used to determine the error rate E_B of any Berger-encoded I-bit data

$$E_{BC} = \frac{1}{2^I} \sum_{i=0}^{I} C_I^i \cdot (1 - p^{i + w(C(i))}), \tag{11}$$

where $w(C(i))$ is the weight of the check part $C(i)$ that corresponds to data of weight i, whatever encoding is used. In particular, for the BGI encoding proposed here, (11) becomes

$$E_{BGI_N} = \frac{1}{2^I} \Big(\sum_{i=0}^{\lfloor I/2 \rfloor} C_I^i \cdot (1 - p^{i + w(2^K - 1 - i \bmod 2^{K'})}) + \sum_{i=\lfloor I/2 \rfloor + 1}^{I} C_I^i \cdot (1 - p^{I-i+1+w(C(i+Const))}) \Big). \tag{12}$$

TABLE V
ERROR RATE ESTIMATIONS FOR $I = \{2^K - 3, 2^K - 2\}$

K I	e	E	E_{BI}	E_{BC}	E_{BGI_N}	$Red.$
3 5	E-10	2.50E-10	2.06E-10	3.81E-10	2.78E-10	27.0
	E-06	2.50E-06	2.06E-06	3.81E-06	2.78E-06	27.0
	E-03	2.50E-03	2.06E-03	3.81E-03	2.78E-03	27.0
4 13	E-10	6.50E-10	5.53E-10	8.46E-10	6.46E-10	23.7
	E-06	6.50E-06	5.53E-06	8.46E-06	6.46E-06	23.7
	E-03	6.48E-03	5.52E-03	8.43E-03	6.44E-03	23.6
5 29	E-10	1.45E-09	1.28E-09	1.70E-09	1.40E-09	18.1
	E-06	1.45E-05	1.28E-05	1.70E-05	1.40E-05	18.1
	E-03	1.44E-02	1.28E-02	1.69E-02	1.39E-02	18.0
3 6	E-10	3.00E-10	2.41E-10	4.41E-10	2.88E-10	34.7
	E-06	3.00E-06	2.41E-06	4.41E-06	2.88E-06	34.7
	E-03	3.00E-03	2.40E-03	4.40E-03	2.87E-03	34.8
4 14	E-10	7.00E-10	5.93E-10	8.98E-10	6.63E-10	26.2
	E-06	7.00E-06	5.93E-06	8.98E-06	6.63E-06	26.2
	E-03	6.98E-03	5.91E-03	8.94E-03	6.61E-03	26.1
5 30	E-10	1.50E-09	1.33E-09	1.75E-09	1.42E-09	18.9
	E-06	1.50E-05	1.33E-05	1.75E-05	1.42E-05	18.9
	E-03	1.49E-02	1.32E-02	1.74E-02	1.41E-02	19.0

The error rate estimations provided in Table V were computed according to (9)–(12). They show that in the asymmetric channel the BI coding alone leads to from 11% (for $I = 29, 30$) to 20% (for $I = 5, 6$) error rate reduction compared to simple data transmission, although it does not allow to detect any transmission errors. All $1 \rightarrow 0$ transmission errors are detected by the Berger coding alone, although at the increase of the error rate E_{BC} from 17% (for $I = 30$) to 52.5% (for $I = 5$), due to extra check bits which are also susceptible to errors. Finally, it is seen that the BGI encoding is the best solution: it not only lowers the error rate E_{BGI_N} even below the level E of an unprotected channel (except for $I = 5$), but it also allows to detect all $1 \rightarrow 0$ transmission errors. The last column shows that the error rate of the BGI encoding compared to Berger coding alone is still reduced by $Red = 100 \cdot (E_{BI} - E_{BGI_N})/E_{BI}$, i.e. it ranges from 19% (for $I = 30$) to 34.7% (for $I = 6$), at no cost of extra bus lines or increase of codec's hardware (although for $I = 2^K - 2$ only). We have not compared error rates of the new BGI encoding against those from [20], since (8) and (9) used to compute figures for BGI given in Table II of [20] are incorrect.

B. Implementation Results

The codecs by Huang [20] and the new ones were synthesised for selected bus widths using the Synopsys® Design Compiler® suite of tools for a STMicroelectronics 90nm 1.0V technology. Table VI presents the area estimations and compares the two versions according to $Red = 100 \cdot (BGI_H - BGI_N)/BGI_H$%. Significant area reduction of the new encoders is observed (up to over 43%). The areas of decoders are similar, as expected, since the two versions are almost identical. The area reduction for codecs (encoder + decoder) ranges from over 12% for small I to about 40% for large I.

To evaluate accurately the power consumption, some simulation experiments were performed on BGI-protected circuits (codec+bus) for various I, assuming the Metal-4 layer and 2-mm wire length of the bus. For each circuit, 10000 uniformly distributed random data patterns were generated using MATLAB®. As timing constraints for synthesis are set for

TABLE VI
AREA ESTIMATION (IN μm^2) FOR VARIOUS BUS' WIDTHS I

K	I	BGI_H		BGI_N		Red		
		Enc	Dec	Enc	Dec	Enc	Dec	Codec
3	6	317	285	283	244	10.72	14.31	12.51
4	7	406	334	347	285	14.59	14.47	14.53
	10	790	549	561	497	29.02	9.40	19.21
	14	1172	798	809	718	30.99	10.03	20.51
5	16	2613	1482	1503	938	42.50	36.66	39.58
	30	5132	2897	2879	1907	43.90	34.17	39.04
6	32	5958	3652	3651	3073	38.70	15.85	30.01

TABLE VII
POWER CONSUMPTION ESTIMATION (IN μW) FOR VARIOUS BUS' WIDTHS

K	I	Simple	BGI_H	BGI_N	$\uparrow BGI_H$	$\uparrow BGI_N$	Red
3	6	72.88	153.20	144.08	2.10	1.97	6.2
4	7	90.57	196.78	191.51	2.17	2.11	2.8
	10	129.87	307.27	275.27	2.36	2.12	10.2
	14	175.76	487.22	415.36	2.77	2.36	14.8
5	16	203.93	1093.33	758.60	5.36	3.72	30.6
	30	380.16	3925.19	2059.03	10.32	5.41	47.6
6	32	403.70	4419.37	2296.49	10.96	5.69	48.1

100 MHz frequency, we used the same constraints which were easily met for all circuits considered. The activity of the circuits was generated by Mentor Graphics® ModelSim SE Plus 6.4B simulation tools after synthesis for the cells in the nominal library (the most accurate approach available without actually generating the layout). The estimations of the power consumption of codecs and wires were obtained respectively using PrimeTime® B-2008.O6 version from Synopsys® and the high-level estimation CAD tool *Interconnect Explorer* [23].

The statistical results shown in Table VII take the bus wires and the codec into account, obviously except for a simple bus which has no codec. It is seen that the power consumption of the wires is negligible compared to codecs, for which it grows quickly with I. The increase of power consumption of two versions of BGI-encoded buses compared to simple bus ($\uparrow BGI_H$ and $\uparrow BGI_N$) is from about twice for either design for small I to over 10 and 5 times for $I \geq 30$ for Huang's and our designs, respectively. Clearly, our designs allow to reduce power consumption even by about a half for $I \geq 30$, due to simpler encoder. The enclosed data shows that availability of codecs as simple as possible is of utmost importance for any low-power design.

V. CONCLUSION

The Berger-invert (BGI) code is a coding scheme proposed recently to protect communication channels against all asymmetric errors. It not only ensures that all transmission errors are detectable at the receiver side but also enjoys the advantages of decreased error rate and reduced power consumption. In this paper, a slightly modified BGI encoding scheme has been proposed which offers the same error protection as the previous one, but which reduces the error rate from 18 to about 35%. Owing to new encoding, one bus line is saved for all data widths I but for $I = 2^K - 1$ and the codecs (encoders and decoders/checkers) for these codes are significantly simpler. Implementation results of the sample bus systems confirm the reduction of both the area (from 12.5 to about 40%) and the power consumption (up to 48%). They also reveal that the cost of protecting the communication bus against errors, measured by the power consumption, is generally very high (at least two-fold). Interestingly, the new codec for the BGI code with $I = 2^K - 2$ data bits is as simple as for the Berger code.

REFERENCES

[1] R. J. Fletcher, "Integrated circuit having outputs configured for reduced state changes," U.S. Patent no. 4 667 337, May 1987.

[2] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 49–58, Mar. 1995.

[3] J. F. Tabor, "Noise reduction using low weight and constant weight coding techniques," M.Sc. Thesis, Tech. Rep. 1232, Artif. Intell. Lab., MIT, Cambridge, MA, 1990.

[4] K. Nakamura, and M. A. Horowitz, "A 50% noise reduction interface using low-weight coding," in *Digest of Technical Papers of Symp. on VLSI Circuits*, 1996, pp. 144–145.

[5] M. R. Stan and W. P. Burleson, "Coding a terminated bus for low power," in *Proc. Fifth Great Lakes Symp. on VLSI (GLSVLSI'95)*, 1995, pp. 70–73.

[6] H. Zhang, V. George, and J. M. Rabaey, "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Trans. VLSI Systems*, vol. 8, pp. 264–272, June 2000.

[7] C. R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device and Materials Reliab.*, vol. 5, pp. 305–316, Sept. 2005.

[8] Y. Cassuto *et al.*, "Codes for limited magnitude asymmetric errors in flash memories," US Patent 20080168320, July 2008.

[9] P. G. Emma, R. K. Montoye, and W. R. Reohr, "Dynamic memory architecture employing passive expiration of data," US Patent 7 290 203, Oct. 30, 2007.

[10] H. T. Weaver, W. T. Corbett, and J. M. Pimbley, "Soft error protection using asymmetric response latches," *IEEE Trans. Electron. Devices*, vol. 38, pp. 1555–1557, June 1991.

[11] N. Azizi, A. Moshovos, and F. N. Najm, "Low-leakage asymmetric-cell SRAM," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, 2002, pp. 48–51.

[12] H. Deogun *et al.*, "Leakage and crosstalk-aware bus encoding for total power reduction," in *Proc. 41th ACM/IEEE Design Automation Conf. (DAC)*, 2004, pp. 779–782.

[13] F. Nemati, M. R. Kasnavi, and R. H. Igehy, "Memory cell error recovery," US Patent 6785169, Aug. 2004.

[14] A. Abdollahi, F. Fallah, and M. Pedram, "An effective power mode transition technique in MTCMOS circuits," in *Proc. of Design Autom. Conf. (DAC)*, 2005, pp. 37–42.

[15] J. M. Berger, "A note on error detection codes for asymmetric binary channels," *Inform. Contr.*, vol. 4, pp. 68–73, Mar. 1961.

[16] C. V. Freiman, "Optimal error detection codes for completely asymmetric binary channels," *Inform. Contr.*, vol. 5, pp. 64–71, Mar. 1962.

[17] J. E. Smith, "The design of totally self-checking check circuits for a class of unordered codes," *J. Des. Autom. Fault-Tolerant Comput.*, vol. 2, pp. 321–342, Oct. 1977.

[18] J. E. Smith, "On separable unordered codes," *IEEE Trans. Comput.*, vol. C-33, pp. 741–743, Aug. 1984.

[19] J. H. Weber, "Asymptotic results on codes for symmetric, unidirectional, and asymmetric error control," *IEEE Trans. Inform. Theory*, vol. 40, pp. 2073–2075, Nov. 1994.

[20] T.-C. Huang, "A low-power dependable Berger code for fully asymmetric communication," *IEEE Commun. Lett.*, vol. 12, pp. 773–775, Oct. 2008.

[21] S. J. Piestrak, *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*, Scientific Papers of Inst. of Techn. Cybern. of Techn. Univ. of Wrocław, no. 92, Ser.: Monographs no. 24, Wrocław 1995.

[22] S. J. Piestrak, "Efficient Hamming weight comparators of binary vectors," *Electron. Lett.*, vol. 43, pp. 611–612, 24 May 2007.

[23] A. Courtay, O. Sentieys, J. Laurent, and N. Julien, "High-level interconnect delay and power estimation," *J. of Low Power Electron.*, vol. 4, pp. 21–33, Apr. 2008.

TCAS-II 6788
Stanislaw J. Piestrak, Sebastien Pillement, and Olivier Sentieys
"On Designing Efficient Codecs for Bus-Invert Berger Code for Fully Asymmetric Communication"

The list of the major changes, suggested by reviewers

Modifications requested by at least two reviewers

(M1) The same issue of many undefined symbols was raised by all three reviewers. Therefore, we explain here how we have addressed this issue.

1. Below Eqn. (12), i.e., at the very first occurrence of the symbol, we have included the following explanation: "*Note:* henceforth all parameters of the new and Huang's designs will be denoted with indices N and H, respectively."
2. One symbol 'Red' that appears thrice in Tables V–VII was replaced with three separate symbols E_{Red}, A_{Red}, and P_{Red}, accompanied by their explicit definitions in the text.
3. The figures that occur in the last three columns of Table VI are explained in the text.

(M2) We claim that both the equations (8) and (9) by Huang [21] and corresponding error rates for BGI included in his Table II are incorrect (the error rates in BGI encoding by Huang seem to be slightly higher than reported). We attempt to clarify this issue (apart of a few other) in our separate paper "Comments on 'A Low-Power Dependable Berger Code for Fully Asymmetric Communication'" submitted to *IEEE Communications Letters*. We feel that we could reach the definitive conclusion only once we would have an answer to our Comments from the author. Therefore, at the moment, we decided to accept his results as they are, which is reflected by the following statement concluding Section IVB: "The error rate reduction reported in Table II of [21] ranges from 12.3% (for $I = 30$) to 21.3% (for $I = 6$) which is less than here."

(M3)
1. We have done our best to eliminate several errors.
2. The word "since", which was used incorrectly throughout the paper in reference to causality, was replaced with "because".
3. Many cases of "which" was replaced with more appropriate "that".

Reviewer: 1

- The meanings of "red" in TABLE V and VI are different. I believe it presents the efficiency ratio of the old/new approaches. But it is really confuse to use different equations for the same parameter.
  See M1 above.

- In Table V BC and BGI are compared while in Table VI and VII, BGI_H with BGI_N are compared. Why are two different sets of approaches compared for performance and power consumptions?
  All symbols that appear in Tables V–VII are better explained (see M1 above) which, hopefully, would allow to avoid any confusion now.

- Actually, it is quite confused about these notations:
  1. BI, BGI: These two are okay.
  2. BGIN: Does this represent your proposed approach?
  3. BGIH: No definition in the paper

  See M1 above.

- In this paper, 10000 random data patterns are generated for simulations. Why this number is chosen? Is it large enough for accurate/good simulation results?
  New reference was added (it appears as [24]): R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte Carlo approach for power estimation," *IEEE Trans. VLSI Systems*, vol. 1, no. 1, pp. 63–71, Mar. 1993.

Reviewer: 2

While I think the work in this paper is promising, I think the organization and the writing style could be significantly improved. In particular, I found the notation to be confusing. I went through the paper several times,

and could not find the definitions of BGIH and BGIN (did I somehow miss them?). Without a proper definition, it is impossible to figure out what is actually going on in Table VII. Also, perhaps an expanded table caption guiding the reader would improve the flow.
See M1 above.

Detailed comments:

- The authors should have a native English speaker read over their document before resubmission. There are several (picky) errors, especially on the first page.
  See M3 above.
- p.2 Section II-B, first paragraph: I do not understand the comment in the parentheses (to note that...); it appears to be missing some words.
  We intended to say: "(however, notice that an asymmetric error correcting code does not necessarily correct all unidirectional errors [19])." We removed this notice as irrelevant, since here we do not consider error correcting codes at all.
- p. 4 Section IV-A: is the comment regarding the mistake in Table II of [20] your on observation, or has this been recognized elsewhere? Please add a brief explanation.
  See M2 above.

Reviewer: 3

In the contact information, use actual names in the email address. Ex. bob,joe@irisa.fr. Otherwise it is unclear if the first or last name should be used.
Complete email addresses are given now.

The abstract should include some actual numbers for savings in power, area and error rate. Pick a common value of I.
Included.

Section I

- In the first paragraph it is stated that interconnect and associated circuitry can be responsible for up to 50% of energy consumption in an integrated circuit. There should be at least one reference here to back up that claim.
  A new reference was added (it appears as [1]): R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," *Proc. IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.
- The first four sentences sound quite awkward. The term communication systems is very broad. You need to start broad, maybe with something about how designers try to improve the efficiency of ICs while keeping power consumption low, bandwidth high, etc., and then narrow your focus, explaining how intend to accomplish this. You are focusing on data bus wires and I/O circuitry within an integrated circuit. A main goal for this is high bandwidth, and also low power consumption. Then talk about what people have done in the past, and how you improve on it (you've done this fairly well). Fixing up the first few sentences will make it much more clear to the reader what the focus and purpose of your work is and frame it in the correct context.
  Modified.
- Other than that first part I like most of the content in the introduction. The organization is fairly good, and it provides a logical review of past literature and how you arrive at the focus of the paper.
- Unfortunately, the grammar is poor. There are several places throughout the paper where poor grammar makes for an awkward read. I recommend having someone read through the paper to correct the grammar.
  See M3 above.

Section II

- Section A is alright.
- Section B is good.
- In this section you have described bus invert codes, and Berger codes, but not Berger invert codes. While it is somewhat obvious how the two codes are combined, I think there should be a brief formal description (Section C) before jumping right in to the codecs in section III.
  Section II.C was introduced, as suggested.

Section III

- The figures (mainly the font size) could be a little bigger and easier to read.
  Done.
- After equation (2) you should explain why +1 is added to $wo(D)$.
  Done.
- I like Table II, but I think it should be a part of section II.C (suggested above) along with your last paragraph of III.A.
  Now it is a part of new Section II.C, as suggested.
- It might be nice to see another diagram/table showing an example set of D, BI and C, before the channel (with no errors), after the channel (with errors), and the result of decoding and the checker, say for $I = 5$ to match the table. A similar figure could be made for your modified codec. (This isn't that important if space is limited)
  It cannot be included in the paper, due to space limitations. Included here on the last page, for inspection.
- The first line in the right column of page 3 should be moved back to the first time 2^{k-1} is used.
  Below equation (3) the sentence "... with the carry out bit of weight 2^K discarded." was replaced with "... with the carry out bit of weight 2^K discarded, the latter formally denoted as mod 2^K operation." We feel that this resolves the problem indicated by the reviewer and therefore we have decided to remove the sentence concerned "Recall that the operation mod 2^{K-1} relies on discarding the MSBs of weight 2^{K-1} and higher."

Section IV

- I wouldn't call results experimental unless they were found from a fabricated chip. Rather they are simulated results.
  We have replaced 'experimental' with 'simulation' throughout the paper.
- At the end of section A you state that the equations in [20] are incorrect. Give an explanation for this. What in particular is incorrect? Is it straightforward to correct the equations in [20] and reproduce those results for comparison?
  See M2 above.
- You say that the power consumption in the bus wires is negligible compared to the codecs. I wouldn't call it negligible, especially for low values of I where it is about 1/3 of the total.
  This statement was removed and replaced with more accurate comments. Two columns $\uparrow BGI_H$ $\uparrow BGI_N$ were removed from Table VI. The paragraph commenting the data included in Table VI was completely modified.
- In the text you should make it more clear what each column in the table is. Ex. what do subscripts H and N stand for.
  See M1 above.

Conclusion

- You say that the modified BGI encoding scheme reduces the error rate compared to the 'previous one'. What is the previous one? Berger coding alone, or the BGI scheme from [20]? The wording seems to imply that you are comparing to [20], but in Section IV you said you didn't compare error rates to [20].
  Now the 'previous one' is called explicitly 'Huang's BGI encoding'.
- In the very last line you are comparing your BGI codec to a Berger code codec. What is the Berger codec? You haven't shown it to us or referenced one.
  Due to the lack of space to justify this claim, the sentence "Interestingly, the new codec for the BGI code with $I = 2^K - 2$ data bits is as simple as for the Berger code." was removed.

Overall, I like what you've done with this work. You've obtained significant savings in power, and shown that your codec is an improvement on the one presented in [20]. By fixing up the grammar the paper will be much more readable.

Other modifications:

To save space, we removed error rates for $e = E - 10$ from Table V. Instead, we have included the following notice: "(to note that all error rates for $e = E$-10 have decimal parts identical to those for $e = E$-06)."

Example of the codec with errors, requested by Reviewer 3

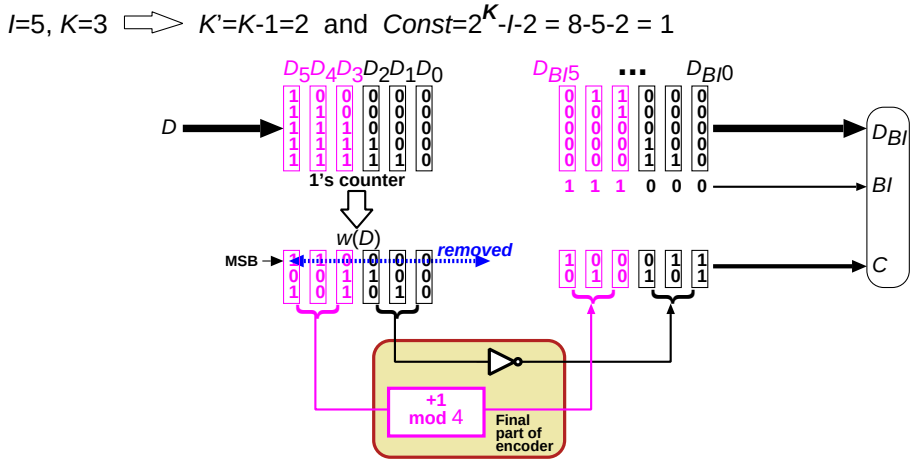

Figure 1: Encoder with six sample input vectors of different weights.

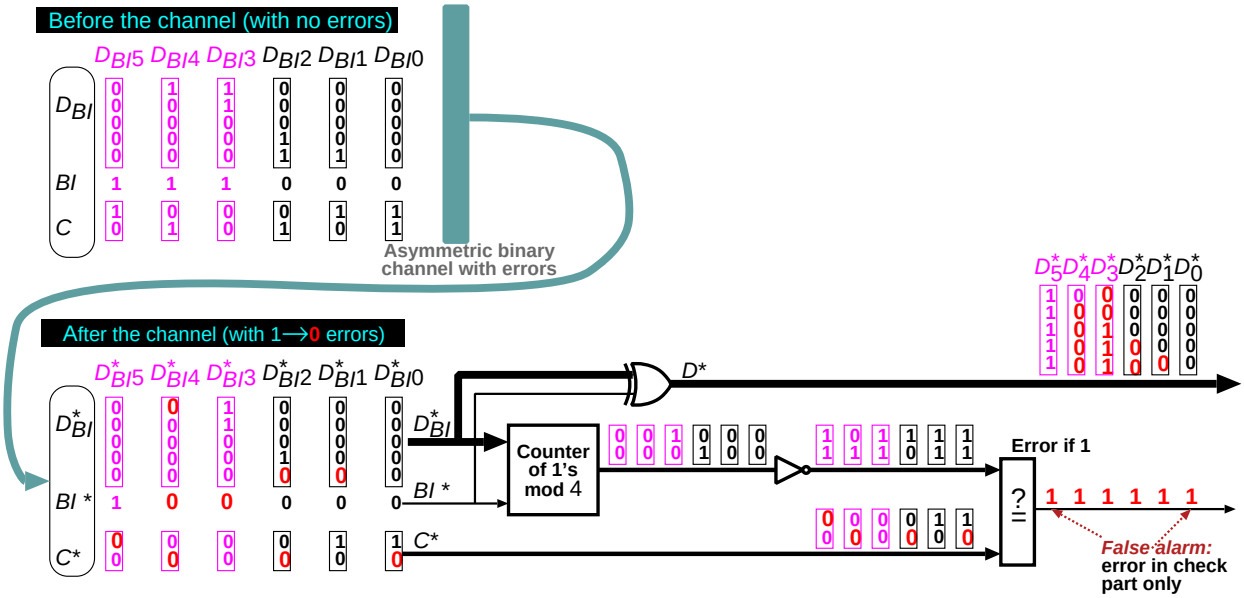

Figure 2: Decoder with the same six sample input vectors and various $1 \rightarrow 0$ errors.