Performance - Timing Overhead Trade-off Analysis for a low-power data bus encoding based on input lines reordering

Mauro Olivieri
Department of Electronic Engineering
La Sapienza University of Rome
Via Eudossiana 18-00184-Rome,Italy
Email: Olivieri@dei.uniroma1.it

Francesco Pappalardo and Giuseppe Visalli
ST Microelectronics M6
Contrada Blocco Torrazze -95121 Catania, Italy
Email: Francesco.Pappalardo@st.com
Email: Giuseppe-ast.Visalli@st.com

Abstract—This paper analyzes the performance and timing overhead trade-off for a recently proposed data bus encoding scheme for low-power based on data lines reordering. The Bus Switch (BS) mechanism introduces greater activity savings than previous approaches; the hardware complexity of the encoder suggests to apply BS in off-chip buses, where the parasitic capacitance makes dynamic power dissipation in the bus lines the dominant contribution to power consumption. In the basic BS implementation, the encoding circuits included extra bus lines which degrade the energy saving. This paper illustrates and analyzes a circuit implementation with only one extra line, at the cost of a small time overhead. This solution strongly enhances the advantage in off-chip communications, where the available number of pads represents a key resource in low-cost packages. Our results indicate that the effectiveness of the approach strongly depend on an a-priori traffic analysis.

KEYWORDS
Bus transfer, Encoding, MOS memory integrated circuit, Power demand, Very-large-scale integration

I. INTRODUCTION AND BACKGROUND

Nowadays, several silicon foundries do not consider the maximum timing performance (e.g. core speed) as the basic guideline for modern and Systems-on-chip (SoC). In many cases, energy dissipation, cost and full connectivity are the primary goals for the future generation of SoC designs. The increased demand for low-energy-consuming and high-data-rate interconnections in digital embedded systems have been encouraging the development of encoding schemes for address and data buses. The activity reduction in address buses [10] [3] [4] exploits the fact that consecutive fetch addresses mostly show a small Hamming distance, due the instruction locality during program execution. Moreover, the address space can be partitioned in different working zones which are used to make the memory access with a reduced bus [5]. Adaptive techniques trace the address bus for a given program, choosing the best encoding scheme in those applications showing low level of address sequentiality (Beach Code) [2]. Additionally, different encoder/decoder models have been proposed for a general approach to the problem [1] [7] (Information-Theoretical code). As a general result, while statistical approaches assume data statistics known in advance, such assumption does not hold in many applications, where the more general adaptive techniques are preferable.

In [6], a novel encoding scheme devoted to high capacitance off-chip buses was proposed, based on clustering, reordering and encoding the bus lines according to a run-time-defined re-ordering pattern and coding function. The performance results demonstrated the effectiveness of the technique in reducing the switching activity, with respect to other adaptive strategies. Additionally, the approach results to be almost insensitive to the bus width (Fig. 1), permitting low-energy high-data-rate transmissions. The encoder complexity limits the field of applications to off-chip buses, where the dynamic energy saving dominates the encoder power consumption (for CMOS technologies below 180 nm [6]). The basic VLSI implementation of the BS requires transmitting the reordering pattern over extra bus lines, which implies a further switching activity, reducing the efficiency of the technique. Even more importantly, the availability of pads in a low-cost package represents an important constraint for an off-chip bus. In this work we propose an extension of the technique by introducing a different method for transmitting the reordering pattern over a single line. This issue improves the BS range of application in terms of minimal parasitic capacitance on the bus lines for an effective utilization. The proposed design solution implies a variable timing overhead for a complete reordering pattern transmission. The design includes an asynchronous FIFO memory to increase the average bandwidth. An off-line analysis allows us to properly size the design so to achieve definitely better performance than the basic BS approach.

This work illustrates the design issues, analyzing the allowed information band and energy saving trade-off, introducing a variable timing overhead in the reordering pattern transmission. The design includes an asynchronous FIFO memory to increase the average bandwidth. An off-line analysis allows us to properly size the design so to achieve definitely better performance than the basic BS approach.

This work illustrates the design issues, analyzing the allowed information band and energy saving trade-off, introducing a variable timing overhead in the reordering pattern transmission. The paper follows this organization: section 2 illustrates the bus switch mechanism, the design issues and the related VLSI circuits. Section 3 shows the proposed technique for transmitting the reordering pattern on a single line, minimizing...
the related switching activity. Section 4 analyzes the allowed information demand and timing overhead depending on the maximum latency for reordering-pattern transmission. Section 5 illustrates the simulation results that indicates how the prior analysis implies the best activity savings at minimal average timing overhead. Section 6 summarizes our conclusions.

II. THE BUS SWITCH MECHANISM

A. Theory

In principle, the Bus Switch technique can be logically expressed as a four-step process:

1) A large bus is divided into several identical clusters of M (cluster depth) lines each.
2) Each M-line bus is coded by swapping the input lines using a particular reordering pattern
3) A tentative data encoding is obtained by applying to the swapped M lines a fixed coding function
4) The process is repeated M! times from step 2 until the optimal reordering pattern is found, that minimizes the output switching activity in the encoded data of the whole bus.

In the following a formal definition of the process is given. Let \( b(t) \) be input data word to the bus encoder and \( B_{opt}(t) \) the encoded data word on the bus, at clock cycle \( t \). The single bits of any M-bit data word \( x(t) \) will be indicated as \( x(t)(0), x(t)(1), \ldots, x(t)(M - 1) \).

**Definition 1:** A reordering pattern \( p(t) \) is an ordered set of M indices \( i_0 \ldots i_{M-1} \), associated with clock cycle \( t \). Given a data word \( x(t) \), the swap operator \( S_w \) with reordering pattern \( p \) is a combinational logic function producing a swapped data word \( y(t) = S_w(x(t), p(t)) \), such that:

\[
\begin{align*}
y(t)(0) &= x(0)(i_0) \\
y(t)(1) &= x(0)(i_1) \\
&\quad \vdots \\
y(t)(M-1) &= x(0)(i_{M-1}).
\end{align*}
\]

As an example, if \( p(t) = "1,2,3,0" \) and \( x(t) = "0100" \), then \( S_w(x(t), p(t)) = "1001" \). Note that each reordering pattern \( p(t) \) has a unique inverse \( p^{-1}(t) \), such that:

\[
x(t) = S_w(x(t), p(t)), p^{-1}(t)
\]

For instance if \( p(t) = "1,2,3,0" \) than \( p^{-1}(t) = "3,0,1,2" \)

**Definition 2:** A coding function is a combinational logic function producing a data word \( B(t) \), applying swapping to

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Coding Function</th>
<th>Decoding Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS Type I</td>
<td>( B(t) = S_n(b(t), p(t)) )</td>
<td>( b(t) = S_n(B_{opt}(t), p^{-1}(t)) )</td>
</tr>
<tr>
<td>BS Type II</td>
<td>( B(t) = S_n(b(t), p(t)) \oplus S_n(b(t-1), p^{-1}(t)) )</td>
<td>( b(t) = S_n(S_n(b(t-1), p^{-1}(t)) \oplus B_{opt}(t)), p^{-1}(t) )</td>
</tr>
<tr>
<td>BS Type III</td>
<td>( B(t) = S_n(b(t), p(t)) \oplus S_n(B_{opt}(t-1), p^{-1}(t)) )</td>
<td>( b(t) = S_n(S_n(b(t-1), p^{-1}(t)) \oplus B_{opt}(t)), p^{-1}(t) )</td>
</tr>
</tbody>
</table>
Table 1 illustrates different coding/decoding functions, for the proposed low-power bus encoding approach. Fig. 2 shows the activity savings, stimulating the bus with binary and multimedia benchmarks and employing the Type III coding and decoding functions (Table 1). The bus switch mechanism implicates several design performance trade-offs:

- The bus switch encoder (BSE) grows in complexity increasing the cluster depth M. In particular, the optimal BS allows M! different patterns. This issue suggests the employment of a reduced and optimized sub-set of reordering patterns, decreasing the hardware complexity. Fig. 2 shows that a reduced and optimized pattern set (M=4 and 16 patterns BS4X (16)) does not relevantly affect performance, compared to the optimal scheme (OPT-BS4X (24)).
- The process of patterns selection could be on-line or off-line (Fig. 2) driven by the activity savings.
- The activity savings do not significantly change, varying the bus input lines (Fig. 1).

Table 2 illustrates the activity savings employing the three different encoding functions, compared with the known data-bus encoding for low-power: bus invert (BI) [9] and adaptive partial bus invert (APBI) [8] (32-line bus, cluster depth 4 and 24 patterns). Results indicated how the simple bus reordering (Type I) does not guarantee a clear convenience with traditional approaches. The combined actions of reordering and coding permits better performance than BI and APBI. The efficiency of coding can be theoretically justified since the bus reordering de-correlates the input stream b(t). Encoding functions, whose XOR operator receives strongly un-correlated processes, gives the best activity savings. The stochastic process B(t) increases its correlation in Type III much better than other analyzed functions. However, other encoding functions permit similar activity savings while implying more efficient VLSI implementation. For instance:

\[ B(t) = b(t) \oplus S_w(B_{opt}(t-1), p^{-1}(t)) \]  

(1)

This decreases area and power at similar switching activity reduction compared to BS-III (Average 15.85%).

B. The Bus Switch VLSI implementation

This section illustrates the Bus Switch VLSI circuit implementation, starting from the bottom level architecture to the top.

1) Encoder: The reordering patterns can be sequentially generated by a finite state machine (FSM) very similar to a binary counter. The direct binary representation of a swapping pattern is a vector of M binary numbers each ranging from 0 to M-1, therefore requiring \( M - \log_2(M) \) bits. The swap operation is performed by a set of multiplexers. Coding function I is directly the swapped word. Coding functions II and III (Table 1) are implemented by a "twin swap unit", which logic function is:

\[ S_w(x1(t), p(t)) \oplus S_w(x2(t), p^{-1}(t)) \]  

(2)

Where \( x1(t) \) and \( x2(t) \) are generic buses at clock cycle \( t \). The conversion from a swapping pattern to its inverse is directly implemented by a dedicated two-level combinational logic unit PConv. A fully sequential implementation of a

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>BS-I</th>
<th>BS-II</th>
<th>BS-III</th>
<th>BI</th>
<th>APBI</th>
</tr>
</thead>
<tbody>
<tr>
<td>LaTeX</td>
<td>7.22%</td>
<td>12.65%</td>
<td>20.38%</td>
<td>7.78%</td>
<td>6.70%</td>
</tr>
<tr>
<td>Space</td>
<td>4.96%</td>
<td>13.99%</td>
<td>20.51%</td>
<td>10.65%</td>
<td>10.21%</td>
</tr>
<tr>
<td>Gcc</td>
<td>6.36%</td>
<td>13.11%</td>
<td>17.51%</td>
<td>9.06%</td>
<td>8.37%</td>
</tr>
<tr>
<td>Jpeg</td>
<td>14.27%</td>
<td>17.29%</td>
<td>15.61%</td>
<td>10.02%</td>
<td>6.54%</td>
</tr>
<tr>
<td>MP3</td>
<td>14.46%</td>
<td>17.46%</td>
<td>16.70%</td>
<td>10.27%</td>
<td>6.39%</td>
</tr>
<tr>
<td>AVI</td>
<td>5.82%</td>
<td>2.26%</td>
<td>16.24%</td>
<td>0.04%</td>
<td>6.12%</td>
</tr>
</tbody>
</table>

**TABLE II**

**CONVENIENCE OF BS APPROACH, WITH RESPECT TO KNOWN ALGORITHMS**
BS decoder would require the unit to perform $M!$ sequential attempts before selecting the best pattern and corresponding encoded word. This would imply an operating clock frequency at least $M!$ times faster than the bus operating frequency. More conveniently, a partially or fully parallel implementation can be pursued, employing $L$ units, each performing $M!/L$ attempts. In the following we will refer to such solution as an $L$-way parallel architecture (Fig. 3). The corresponding architecture for the single way is shown in Fig. 4. PatGen is the FSM that generates the set of allowed pattern to be tried; $H$ produces the Hamming distance between two words by performing a population count after XOR-ing. The $Cmp$ unit compares the actual Hamming distance measured with the temporary minimum. When all the patterns have been tried and the minimum distance found, the threshold unit stores the pattern, the encoded word and the distance value on output registers (pattern, out and score respectively in Fig. 3). A special care is deserved by the pattern transmission over the bus. Though a direct representation requires $M \cdot \log_2(M)$ bits, the actual number of valid patterns is at most $M!$ in a full pattern set, and even less in a reduced pattern set. Therefore, by introducing a dedicated combinational compressor transforming the direct representation into a symbolic binary representation, the extra-lines to transmit the patterns must be at most $\log_2(M!)$. In addition, in order to minimize the switching activity in those extra lines, a conventional Bus Invert coding is used on them.

2) Decoder: When using coding function $I$, the decoder architecture is directly an inverse swap operator. The PConv combinational block performs the pattern conversion to obtain the inverse pattern. In addition, a BI decoding unit and a combinational de-compressor elaborate the extra-lines dedicated to pattern transmission, to reconstruct the direct representation of the transmitted pattern. Fig. 5 shows the architecture of the decoder for coding function $I$.

### III. Extension of the BS Technique

In order to increase the BS performance, a special attention is deserved by pattern transmission over the bus. A low-power implementation of it involves two identical FSMs, which have a number of internal states equal to the total reordering patterns. Each state ideally represents a particular pattern, which could be used in a coded stream $B(t)$ at time $t$. The two FSM have the same internal states at every time $t$. The FSM in TX-Side generates a synchronism signal (Fig. 8), inverting the sync line when its internal state is equal to the optimal pattern. When the FSM in RX-Side receives the sync signal, it read its internal state and produces the transmitted pattern. After that, both TX and RX FSM reset their internal state; this important issue makes variable the total latency. The proposed system has to be synchronized to avoid data losses. In particular, the two FSM work at frequency multiple of bus cycle $f_0$ and, in general, sub-multiple of encoder frequency (unit patgen), so the whole system is feasible introducing a variable timing overhead. It might happen that a transmission requires more than a bus cycle, so the FIFO is mandatory to avoid data losses. Fig. 6 and 7 illustrate the proposed architecture both for transmitting and receiving data and reordering pattern with minimal electrical activity. FSM internal controls the FIFO operations, gating the used clocks when bus in idle state. Table 3 shows how the proposed approach increases the performance.

### IV. Information Demand and Timing Overhead Analysis

In the following we analyse the performance of the proposed extension, using concepts and terms typical of queueing network theory. The proposed pattern transmission mechanism introduces an overhead in the total latency. In particular, for each transmission the circuit introduces a maximum overhead $N$ such that:

$$f_{BUS} \cdot P_{FSM} - 1 = N$$

![Fig. 8. The principle of reduced-power pattern transmission](image-url)
Fig. 9. Minimum bus capacitance for convenient BS at 180nm (32-line bus)

Where \( P = M! \) in a complete BS. We considered a reduced \( P = 16 \) from an off-line analysis, so:

\[
\begin{align*}
\text{fBUS} &\leq f_{FSM} \leq P \cdot \text{fBUS} \\
\text{Overhead cycle} &\leq T_{BUS} \cdot E\{C\}
\end{align*}
\]

The FIFO introduces an Out of service probability, which depends on the statistics of \( C \). The F-size FIFO is a MM/1 queue with constant birth and death ratio:

\[
\lambda = \frac{1}{T_{BUS}} \cdot \delta
\]

\[
\mu = \frac{1}{T_{BUS} \cdot (E\{C\} + 1)}
\]

\[
P_{\text{out of service}} = \left( \frac{\lambda}{\mu} \right)^{F} \cdot (E\{C\} + 1)^{F} \cdot F < 1
\]

The necessary condition for an out of service probability less than 1 in the steady state condition is:

\[
\delta < \frac{1}{E\{C\} + 1}
\]

The allowed information rate depends on the used patterns (\( P \)), the FSM design and frequency, and the traffic’ statistical distribution. Improvement in information rate come at price in terms of activity savings decreasing \( P \) or hardware feasibility increasing \( f_{FSM} \).

V. SIMULATION RESULTS

We demonstrated the effectiveness of the BS approach by calculating the activity saving performance both in the original and the newly proposed implementation (Table 2 and 3). We considered some representative benchmarks composed of both binary files and multimedia streams as typical traffic in a data bus: LaTeX distribution, Berkeley Spice, Gnu Gcc, and Jpeg, Mp3 and Avi samples. The ANSI C behavioral BS model counts the activity savings, employing the BI model in its basic implementation and in the new one, taking account of the sync line activity in the proposed approach. Additionally, we explored the minimum capacitance for the effective use of the BS approach in these two operative conditions. The total balance of average energy saving per bus cycle is therefore:

\[
E_{\text{saved}} = 0.5 \cdot \alpha \cdot C_{Bus} \cdot V_{dd}^2 - E_{\text{overhead}}
\]

Where \( \alpha \) represents the switching activity reduction and \( E_{\text{overhead}} \) the encoder’s energy dissipation; the total energy saving percentage is expressed by the ratio:

\[
E_{\%} = \frac{(0.5 \cdot C_{Bus} \cdot V_{dd}^2 - E_{\text{saved}})}{(0.5 \cdot C_{Bus} \cdot V_{dd}^2)} \cdot 100
\]

A value of \( E\% \) lower than 100% means that the BS is effective in reducing the total energy consumed per bus cycle, while \( E\% \) greater than 100% means that the bus capacitance is so small that the energy overhead of the encoder dominates and the BS technique is inappropriate. Referring to recent sub-micron CMOS implementations of BS-III (32-bit bus, 4-bit cluster size, 16 patterns) we can show the dependency of the \( E\% \) from the bus line capacitance, and compare it with the considered applications. Fig. 9 and 10 show the results for the 4-way implementation, in 180nm and 130nm respectively with parallel extra bus for pattern transmission. The minimum capacitance for the BS convenient use is 4pF.

<table>
<thead>
<tr>
<th>BenchMark</th>
<th>BS classic</th>
<th>Proposed BS circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>LaTeX</td>
<td>20.38%</td>
<td>26.94%</td>
</tr>
<tr>
<td>Spice</td>
<td>20.51%</td>
<td>26.80%</td>
</tr>
<tr>
<td>Gcc</td>
<td>17.51%</td>
<td>23.67%</td>
</tr>
<tr>
<td>Jpeg</td>
<td>15.61%</td>
<td>21.71%</td>
</tr>
<tr>
<td>MP3</td>
<td>16.70%</td>
<td>22.93%</td>
</tr>
<tr>
<td>AVI</td>
<td>16.24%</td>
<td>20.07%</td>
</tr>
</tbody>
</table>

TABLE III

THE CLASSIC BS ACTIVITY SAVING COMPARED TO THE PROPOSED CIRCUIT
and 2pF in 180nm and 130nm respectively. The proposed approach decreases this capacitance to 2pF in 180nm (32-line bus). The power and timing overhead reduction has been accomplished by analyzing the considered benchmarks, and measuring the frequency of the used reordering patterns. This off-line analysis placed the sixteen most used patterns in the first positions. Table 4 and 5 show the timing overhead at different FSM clock frequency (4X, 2X, 1X), with and without (random used patterns) power and timing overhead optimization respectively. Finally, figure 11 shows how the FIFO employment increases the allowed information rate (delta) reducing the energy saving, at fixed out-of-service probability (0.01). We operated with bus load and frequency at 8pF and 25 MHz respectively, translating the RTL with a 180nm low-leakage technology library. FSM operated at 200 MHz (8X). The trade-off analysis concluded that a 56-word FIFO permitted a sufficient information rate.

VI. CONCLUSION AND DISCUSSIONS

This paper presented the basic principle for a very low-power data bus encoding implementation, by means of a reordering-based technique, whose reordering scheme is transmitted over a single line with reduced switching activity. The proposed method uses two identical finite state machines whose states are synchronized by a single bus line. The bus and FSM clock difference implies the use of a FIFO asynchronous memory, which reduces the source out-of-service probability.

Such probability and the information demand can be strongly improved by an appropriate strategy for the FSM design, placing the most used reordering schemes in the first positions. The related trade-off analysis, referring to a 180nm technology, indicated a 56-word FIFO size with information rate almost 60% the best operating condition for this architecture. Although the maximum latency overhead represents an issue, in particular for some DRAM memory communications, the extra lines reduction has an important role in the energy and cost budgets of whole system. The obtained results are compliant with the view of energy cost reduction as the major target for most Systems-on-Chip in the semiconductor market. The proposed Bus Switch system could be used in low-cost packages with limited pin count, for reducing dynamic power consumption in off-chip buses.

REFERENCES


