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

ABSTRACT

This paper analyzes the performance and timing overhead trade-off for a novel data off-chip bus encoding for low-power based on input lines reordering. The Bus Switch (BS) mechanism introduces greater activity savings than previous approaches. The paper illustrates a circuit implementation with a single dedicated line for transmitting the reordering scheme, sacrificing little timing overhead. This solution 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 depends on a prior traffic analysis.

Keywords

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

1. INTRODUCTION AND BACKGROUND

The increased demand for low-energy electrical interconnections at high data-rate cannot always be satisfied with the actual strategies for encoding address and data buses. The activity reduction in address buses [10] [3] [4] considers most of two consecutive words have a reduced Hamming distance, due the instruction locality during program execution. In particular, the address space might identify different *working zones* used to describe the memory access with a reduced bus [5]. Adaptive techniques trace the address bus for a given program, choosing the best encoding scheme in address stream with low level of sequentiality (Beach Code) [2]. The low-power data bus-encoding introduces an adaptive approach in order to estimate the data correlation between two consecutive words. The main scientific contributions in this field try to encode the input data according to a coding function. Additionally, different encoder/ decoder models have been proposed for a general approach to the problem [1] [7] (Information-Theoretic code). Such approaches considered the input data statistics known in advance; this assumption does not held in many applications, where the general adaptive techniques have been preferred. The experimental result indicates how the performance degrades increasing the data-rate as number of lines employed at fixed bus frequency. Fig. 1 illustrates the performance of the main strategies for data buses: transition based bus-invert (BI) [9] and adaptive partial bus invert (APBI) [8]. The paperwork illustrated and extended a novel approach to bus encoding for low-power, clustering, reordering and encoding the input lines according to a known reordering pattern and a coding function [6]. The authors in [6] demonstrated the convenience of this technique with respect to other adaptive strategies for high-capacity off-chip buses. Additionally, our experimental results indicated how this approach is strongly insensitive for the bus width (Fig. 1), permitting low-energy high data-rate transmissions. Unfortunately, the encoder complexity limits the field of applications to off-chip buses. In this scenario, the dynamic energy savings dominate the encoder power consumption, from technologies at 180nm and below [6]. The VLSI implementation requires the reordering pattern transmission over extra lines, which implies a further switching activity, reducing the efficiency of coding. Moreover, the availability of pads in a low-cost package represents a key resource in a off-chip scenario. We propose a different method for reordering pattern transmission over a single, power-optimized, line. This issue increases the efficiency of coding, improving the BS' field of application in terms of minimal parasitic capacitance for a convenient utilization as our considered performance. The proposed solution requires a *variable timing overhead* for a complete reordering pattern transmission, which can be

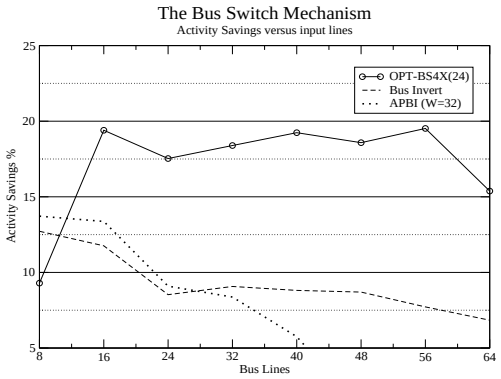

Figure 1: The BS activity savings varying the bus input lines, compared to the known algorithms

Circuit	Coding Function	Decoding Function
BS Type I	$B(t) = S_w(b(t), p(t))$	$b(t) = S_w(B_{opt}(t), p^{-1}(t))$
BS Type II	$B(t) = S_w(b(t), p(t)) \oplus S_w(b(t-1), p^{-1}(t))$	$b(t) = S_w[(S_w(b(t-1), p^{-1}(t)) \oplus B_{opt}(t)), p^{-1}(t)]$
BS Type III	$B(t) = S_w(b(t), p(t)) \oplus S_w(B_{opt}(t-1), p^{-1}(t))$	$b(t) = S_w[(S_w(B_{opt}(t-1), p^{-1}(t)) \oplus B_{opt}(t)), p^{-1}(t)]$

Table 1: Different coding and decoding functions

optimized by an off-line analysis. The paperwork addressed the design issues, analyzing the allowed information demand and energy saving trade-off, introducing a variable timing overhead in the reordering pattern transmission. Additionally, we employed a FIFO asynchronous memory (from ST Microelectronics and available in 180nm only) for increasing the allowed bandwidth. Results indicated that the off-line analysis is the major requirement for an increased information demand and an effective usefulness of FIFO memory. Nowadays, several silicon foundries do not consider the max timing performance (e.g. core speed) the basic guideline for modern and silicon-based systems on a chip. These companies consider the energy dissipation, costs and full connectivity the primary goals for the future generation of silicon systems. 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 method for transmitting the reordering pattern onto 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 is reserved for the conclusion.

2. THE BUS SWITCH MECHANISM

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 $x(t)(0), x(t)(1), , x(t)(M-1)$.

DEFINITION 1. *A reordering pattern $p(t)$ is an ordered set of M indices $i_0 \dots i_{M-1}$, 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 :*
$y(t)(0) = x(t)(\ i_0)$; $y(t)(1) = x(t)(\ i_1)$;
$\dots$, $y(t)(M\text{-}1) = x(t)(\ i_{M-1})$.

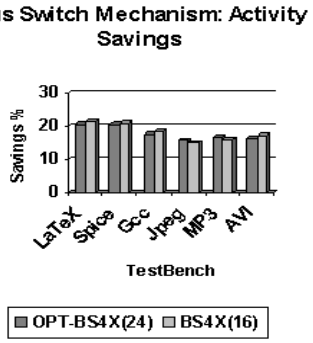

Figure 2: The Bus-Switch Activity Savings (Bus Lines = 32 , Cluster Depth = 4)

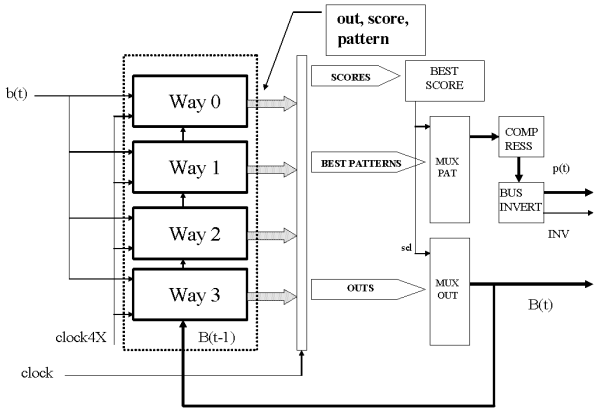

Figure 3: The 2 pipe stages BS encoder (Type III): 32-lines 4-Way and 16 optimized patterns

As an example, if $p(t)$ = "1,2,3,0" and $x(t)$ = "0100", then $S_w(x(t), p(t))$ = "1000". Note that each reordering pattern $p(t)$ has a unique inverse $p^{-1}(t)$, such that $x(t) = S_w(S_w(x(t), p(t)), p^{-1}(t))$. 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 $b(t)$ and employing any other words resulting from input or output observation.*

DEFINITION 3. *The optimal reordering pattern $p_{opt}(t)$ is the reordering pattern that minimizes the switching activity*

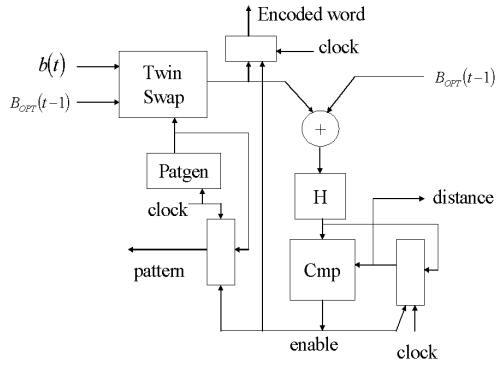

Figure 4: The single way architecture

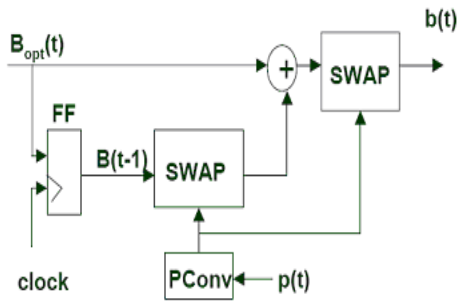

Figure 5: The BS-III Decoder architecture

$H[B_{opt}(t-1) \oplus B(t)]$, where H is the Hamming distance from previous transmission $B_{opt}(t-1)$ and the coding function result, varying the reordering pattern.

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-off:

- The bus switch encoder 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 how 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: BI and APBI (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 performance better than BI and APBI. The efficiency of coding could be easily 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 process $B(t)$ increases its correlation in Type III much better than other analyzed functions. However, other encoding functions permit similar activity savings implying different VLSI design implementation. For instance:

$$B(t) = b(t) \oplus S_w\big(B_{opt}(t-1), p^{-1}(t)\big) \qquad (1)$$

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

2.1 The Bus Switch VLSI implementation

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

2.1.1 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 $M \cdot log_2(M)$ 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:

$$S_w\big(x1(t), p(t)\big) \oplus S_w\big(x2(t), p^{-1}(t)\big) \qquad (2)$$

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 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

Benchmark	BS-I	BS-II	BS-III	BI	APBI
LaTeX	7.22%	12.65%	20.38%	7.78%	6.70%
Spice	4.96%	13.99%	20.51%	10.65%	10.21%
Gcc	6.36%	13.11%	17.51%	9.06%	8.37%
JPeg	14.27%	17.29%	15.61%	10.02%	6.54%
MP3	14.46%	17.46%	16.70%	10.27%	6.39%
AVI	5.32%	2.26%	16.24%	0.04%	6.12%

Table 2: Convenience of BS approach, with respect to known algorithms

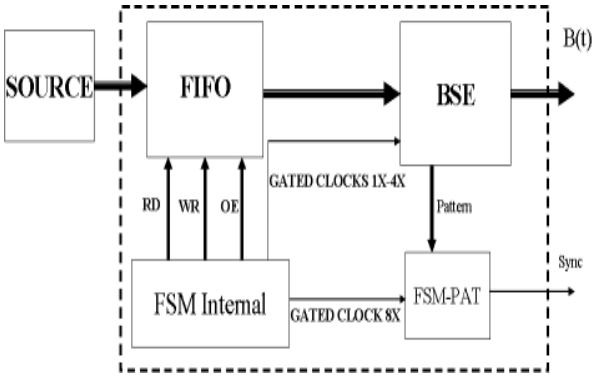

Figure 6: The Proposed BS-Transmission Side

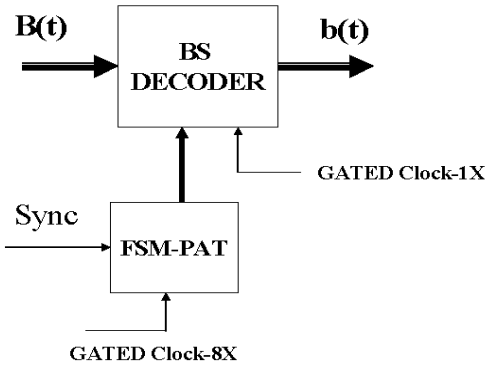

Figure 7: The Proposed BS Reception Side

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 attention 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.

2.1.2 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 III.

3. INCREASING PERFORMANCE WITH A DEDICATED LOW-POWER LINE FOR PATTERN TRANSMISSION

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 illustrates 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.

4. INFORMATION DEMAND AND TIMING OVERHEAD ANALYSIS

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:

$$\frac{f_{BUS} \cdot P}{f_{FSM}} - 1 = N \tag{3}$$

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

$$f_{BUS} \leq f_{FSM} \leq P \cdot f_{BUS} \tag{4}$$

The overhead introduced for each transmission is a random variable $C = \{0, 1, 2, 3, .., N\}$, which depends on the statistical properties of the source data *and* the pattern placement in the FSM. An off-line analysis could place the most used patterns in the first positions, decreasing the average latency overhead, which is expressed by:

$$\overline{Overhead_{cycle}} = T_{BUS} \cdot E\{C\} \tag{5}$$

The FIFO introduces an *Out of service probability*, which depends on the statistics of C. The F-size FIFO is a MM/1

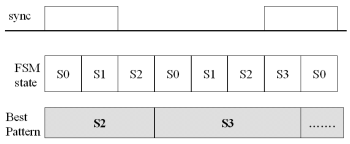

Figure 8: The principle of reduced-power pattern transmission

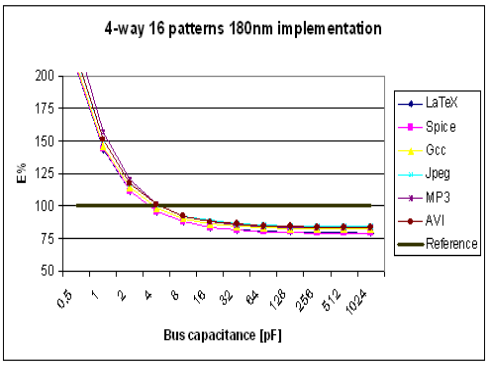

Figure 9: Minimum bus capacitance for convenient BS at 180nm (32-line bus)

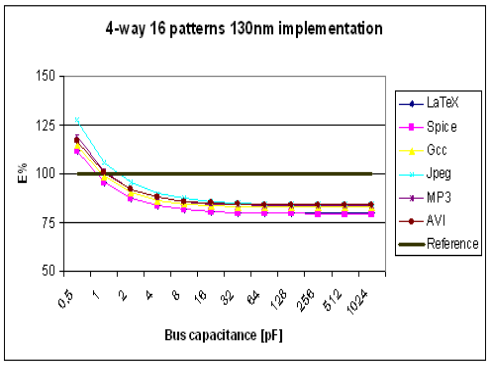

Figure 10: Minimum bus capacitance for convenient BS at 130nm (32-line bus)

queue with constant birth and death ratio:

$$\lambda = \frac{1}{T_{BUS}} \cdot \delta \tag{6}$$

Delta represents the information rate ($\delta < 1$).

$$\mu = \frac{1}{T_{BUS} \cdot \left(E\{C\} + 1\right)} \tag{7}$$

$$P_{outofservice} = \left(\frac{\lambda}{\mu}\right)^F = \left(E\{C\} + 1\right)^F \cdot \delta^F < 1 \tag{8}$$

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

$$\delta < \frac{1}{E\{C\} + 1} \tag{9}$$

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}.

5. SIMULATION RESULTS

The effectiveness of the BS approach has been demonstrated calculating the activity saving performance both in the original and the newly proposed implementation, which differs for the architecture used for the reordering pattern transmission (Table 2 and 3). We considered some representative benchmarks both data 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 the original scenario and, taking account of the sync line activity in the proposed approach. Additionally, we explored the minimum capacitance for the convenient use of the BS approach in these two operative conditions. The total balance of average energy saving per bus cycle is therefore:

$$E_{saved} = 0.5 \cdot \alpha \cdot C_{Bus} \cdot V_{dd}^2 - E_{overhead} \tag{10}$$

Where α represents the switching activity reduction and $E_{overhead}$ the encoder's energy dissipation; the total energy saving percentage is expressed by the ratio:

$$E_{\%} = \frac{\left(0.5 \cdot C_{Bus} \cdot V_{dd}^2 - E_{saved}\right)}{\left(0.5 \cdot C_{Bus} \cdot V_{dd}^2\right)} \cdot 100.0 \tag{11}$$

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 the recent sub-micrometric technologies 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 employment is 4pF and 2pF in 180nm and 130nm respectively. The proposed approach for reordering pattern transmission decreases this capacitance to 2 pF in 180nm (32-line bus). The power and timing overhead reduction has been accomplished analyzing the considered benchmarks, and measuring the frequency of the used reordering patterns. This off-line analysis placed the most sixteen 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 showed how the FIFO employment increased 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 25Mhz respectively, translating the RTL with a 180nm low-leakage technology library. FSM operated at 200 Mhz (8X). The trade-off analysis concluded 56 words for FIFO permitted a sufficient information rate.

BenchMark	BS classic	Proposed BS circuit
LaTeX	20.38%	26.94%
Spice	20.51%	26.80%
Gcc	17.51%	23.67%
JPeg	15.61%	21.71%
MP3	16.70%	22.93%
AVI	16.24%	20.07%

Table 3: The classic BS activity saving compared to the proposed circuit

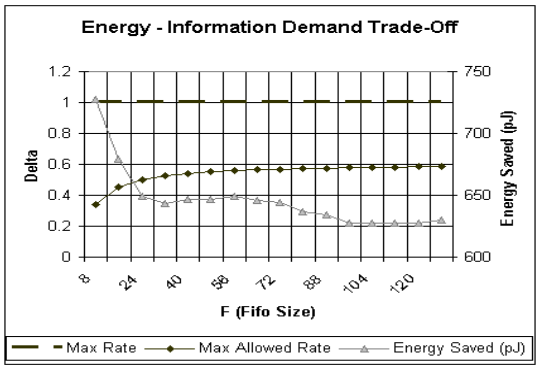

Figure 11: Energy - Information demand Trade off (optimized pattern placement, 180nm tech lib)

6. CONCLUSION AND DISCUSSIONS

The 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 with states synchronized themselves by a single line. The bus and FSM clocks ratio different from unity, implicated the employment of a FIFO asynchronous memory, which reduces the source's out of service probability. This probability and the information demand could 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 indicated as 56 words FIFO size with information rate almost 60% in 180nm the best operating condition for this architecture. Although the max 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-Chips in the semiconductor market. This work implicates the proposed bus switch system could be used in low-cost packages, for reducing dynamic power consumption in off-chip buses.

7. REFERENCES

[1] L. Benini, A. Macii, E. Macii, M. Poncino, and R. Scarsi. Architectures and synthesis algorithms for power-efficient bus interfaces. *IEEE Transaction on Computer-Aided Design*, 19:969–980, Sept. 2000.

[2] L. Benini, G. D. Micheli, E. Macii, D. Sciuto, and S. Quer. Power optimization of core-based systems by address bus encoding. *IEEE Transaction on VLSI Systems*, 6:554–562, Dec. 1998.

[3] L. Benini, G. D. Micheli, E. Macii, D. Sciuto, and C. Silvano. Asymptotic zero-transition activity encoding for address busses in low power microprocessor-based systems. In *ACM/IEEE Great Lake Symposium on VLSI*, pages 77–82, Urbana, IL, Mar. 1997.

[4] L. Benini, G. D. Micheli, E. Macii, D. Sciuto, and C. Silvano. Address bus encoding techniques for system-level power optimization. In *IEEE Design Automation and Test in Europe*, pages 861–866, Paris, France, Feb. 1998.

[5] E. Musoll, T. Lang, and J. Cortadella. Working-zone encoding for reducing the energy in microprocessor address buses. *IEEE Transaction on VLSI Systems*, 6:568–572, Dec. 1998.

[6] M. Olivieri, F. Pappalardo, and G. Visalli. Bus-switch coding for reducing power dissipation in off-chip buses. *IEEE Transaction Very Large Scale Integration Systems*, 12:1374–1377, Dec. 2004.

[7] S. Ramprasad, N. Shanbhag, and I. Hajj. Signal coding for low power: Fundamental limits and practical realizations. *IEEE Transaction on Circuit and Systems II: Analog and Digital Signal Processing*, 46:923–929, 1999.

[8] R. Siegmund, C. Kretzchmar, and D. Muller. Adaptive bus encoding technique for switching activity reduced data transfer over wide system buses. In *Proc. of International Workshop on Power And Timing Modeling Optimization and Simulation*, Gottingen, Germany, 2000.

[9] M. Stan and W. Burleson. Bus-invert coding for low power I/O. *IEEE Transaction on VLSI Systems*, 3:49–58, Mar. 1995.

[10] C. Su, C. Tsui, and A. Despain. Saving power in the control path of embedded processors. *IEEE Design and Test of Computers*, 11:24–30, Apr. 1994.

BenchMark	Activity Savings	$E\{C\}$ N=3	$E\{C\}$ N=7	$E\{C\}$ N=15
LaTeX	22.23%	1.175	2.835	6.237
Spice	21.04%	1.331	3.176	6.875
Gcc	17.68%	1.222	2.980	6.453
JPeg	18.35%	1.042	2.543	5.522
MP3	20.28%	1.019	2.484	5.411
AVI	-2.37%	1.532	3.625	7.812

Table 4: Proposed approach without optimization: activity saving and timing overhead at different FSM clock (N=3,7,15)

BenchMark	Activity Savings	$E\{C\}$ N=3	$E\{C\}$ N=7	$E\{C\}$ N=15
LaTeX	25.45%	0.647	1.561	3.443
Spice	24.77%	0.794	1.929	4.287
Gcc	21.70%	0.801	1.922	4.221
JPeg	18.82%	1.014	2.507	5.536
MP3	20.04%	1.021	2.521	5.562
AVI	16.43%	0.176	0.403	0.882

Table 5: Proposed approach with optimization: activity saving and timing overhead at different FSM clock (N=3,7,15)