

A²BC: Adaptive Address Bus Coding for Low Power Deep Sub-Micron Designs

Jörg Henkel, Haris Lekatsas
C&C Research Laboratories, NEC USA
4 Independence Way, Princeton, NJ 08540
{henkel, lekatsas}@ccrl.nj.nec.com

Abstract

Due to larger buses (length, width) and deep sub-micron effects where coupling capacitances between bus lines are in the same order of magnitude as base capacitances, power consumption of interconnects starts to have a significant impact on a system's total power consumption. We present novel address bus encoding schemes that take coupling effects into consideration. The basis is a physical bus model that quantifies coupling capacitances. As a result, we report power/energy savings on the address buses of up to 56% compared to the best known ordinary power/energy efficient encoding schemes. Thereby, we exceed the only to-date approach that also takes coupling effects into consideration. Moreover, our encoding schemes do not assume any a priori knowledge that is particular to a specific application.

1 Introduction

One of the (many) effects in deep sub-micron designs are coupling capacitances between *close* bus lines. The spatial closeness of bus lines increases the *wire-to-wire* capacitances that much that it may even exceed the base capacitance of a wire i.e. the *wire-to-metal layer*¹ capacitance². Therefore, with these coupling effects in mind, the number of switching activities on a bus (i.e. all transitions on all bus lines) do not necessarily reflect the power/energy that is consumed by the bus. However, this was true (see [1]) for *non-deep sub-micron* designs³. Hence, encoding mechanisms for bus power/energy reduction that solely rely on minimizing the number of transitions are not efficient any more. In fact, any efficient encoding scheme for deep sub-micron buses should be based on a precise physical bus model.

Early work on minimizing the transition activities on buses has been conducted by Stan/Burleson [2]. They transmit the inverted word through the bus when the *Hamming Dis-*

tance (HD for simplicity) of the non-inverted word would result in $HD > N/2$ with N being the number of bus lines. Panda/Dutt [3] approached the problem of reducing switching activities of address busses by investigating various scenarios for memory mapping schemes. Benini et al. [5] presented an adaptive approach for encoding signals that are transmitted through wide buses. The exploitation of correlated access patterns (like in address buses) has been studied in [3] (see above) by using *Gray Code* encoding according to Metha et al. [6] and Su et al. [7]. Benini et al. [8] have improved upon Gray Code through a method that benefits from the fact that a high number of patterns in address buses are consecutive. *Working Zone Encoding* has been proposed by Musoll et al. [9]. They encode according to *where* on an address word switching activity actually takes place. A synthesis method for a spatially adaptive bus interface is presented by Acquaviva/Scarsi [10]. Zhang et al. [11] segment a bus and thus exploit the effect of having smaller effective bus capacitances that apply during bus transitions. Stan/Burleson [12] focus on low power encoding techniques under consideration of influences on possible area and performance impacts. A recent approach by Sotiriadis/Chandrakasan [4] that is close to our approach also takes into consideration the capacitances between wires rather than just the *wire-to-metal-layer capacitance*. Their static encoding technique (i.e. an encoding technique that is fixed) obtains results of an average of 40% power savings.

We address the low power bus encoding problem through the introduction of the *Extended Transition Activity Measure* ETAM that we use to control one of our encoding schemes. Furthermore, we present a physical bus model that quantifies the sizes of coupling and base capacitances. Our novel adaptive (it can be changed over time in order to adapt to different patterns on the address bus) bus encoding scheme eventually takes coupling effects into consideration. That is why we are able to achieve power/energy savings on the address buses of up to 56% compared to the best known ordinary power/energy efficient encoding schemes (*Gray Code* encoding).

This paper is structured as follows: the following Section 2 introduces our physical bus model and shows its characteristics. This bus model is the basis for our encoding schemes in Section 3. In Section 4 we show the effectiveness of our schemes by means of an extensive set of applications that we used to apply our technology. Finally, Section 5 gives a conclusion.

2 Physical Bus Model and its Characteristics

Fig. 1 shows a simplified sectional view through a couple of bus lines. C_B is what we call the base capacitance since it is the intrinsic capacitance between the bus line and the metal layer(s). C_C $_{i,i+1}$ is the coupling capacitance between bus line i and bus line $i + 1$ (not all coupling capacitances are shown in Fig. 1). A simple formula for these capaci-

¹We denote *metal layer* as the layer on a chip layout that carries 0V.

²Note that we implicitly assume that power consumption of CMOS circuits is due to switching activity only. Leakage currents might also become a larger source of power consumption in the future but switching activity in CMOS circuits will continue to be the *main* source of power consumption.

³In this context we say *non-deep sub-micron designs* when we mean designs where the spatial proximity of bus lines or devices etc. does not lead to coupling capacitances that are in the same order of magnitude as the intrinsic (i.e. base) capacitances.

Design Automation Conference ©

Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. DAC 2001, Las Vegas, Nevada

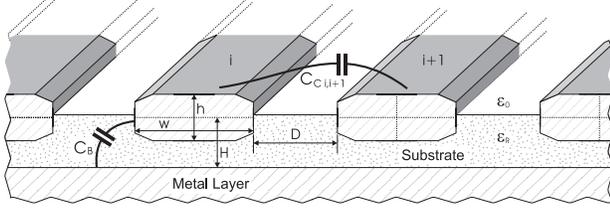


Figure 1: Physical bus model

tances cannot be given because the cross section shape of the bus lines is actually neither a rectangle nor a circle. But we can very closely approximate the solution by assuming that the cross section shape of a bus line is a number of circular cross sections for which solutions can be found directly through the solution of the corresponding differential equations. Consequently, we can represent the per-length capacitance C'_i ⁴ for line i as a superposition of the base capacitance $C'_{B i}$ (*wire_to_metal*) and a coupling capacitance $C'_{C i,i+1}$ (*wire_to_wire*) between i and the closest right neighbor $i + 1$.⁵

$$\begin{aligned} C'_i &= C'_{B i} + C'_{C i,i+1} \\ &= \underbrace{a_0 \cdot \frac{2\pi\epsilon_r\epsilon_0}{\operatorname{arcosh}\left(\frac{H}{h}\right)}}_{\text{wire_to_metal}} + \underbrace{b_0 \cdot \frac{\pi\epsilon_r\epsilon_0}{\ln\left(2\frac{D+w}{h}\right)}}_{\text{wire_to_wire}} \end{aligned} \quad (1)$$

The factors a_0 and b_0 represent the correction factors that allows us to deploy simple equations for the solution of the differential equation rather than using numerical methods⁶. From a geometrical/physical point of view, there exist various ways to minimize $C'_{B i}$ and $C'_{C i,i+1}$ (and thus minimize the implied energy consumption. The basic ways are (discussion applies to Eq. 1):

- **Distance between metal layer and bus line**
It influences the size of $C'_{B i}$. The larger the distance H is the smaller becomes $C'_{B i}$. H is determined by the technology process.
- **Cross section shape of bus lines**
The smaller h and w are (please note that w is implicitly contained in a_0 according to the above explanation) the smaller becomes $C'_{B i}$.
- **Distance between two adjacent bus lines**
A larger distance D reduces $C'_{C i,i+1}$.

If D grows beyond a certain size then the space (i.e. chip area) becomes unacceptably larger since this distance is between all adjacent bus lines and a bus can be very large (typically 32 or 64 lines). Though a reduction of the cross section area ($\approx h \times w$) is beneficial for reducing $C'_{B i}$, there is a technological limit to do so: since the voltage V_{DD} is given for a certain

⁴For brevity reasons we may from now on use the term *capacitance* when we actually mean *per-length capacitance*.

⁵Please note that our bus capacitance model does neglect capacitances resulting from wires in other layers of the chip layout.

⁶It can be proved that this approximation is actually very accurate representation. The reason why we provided the simplified formula within this paper is that it can conveniently be used to explain the characteristics like the relationship between $C'_{B i}$ and $C'_{C i,i+1}$. However, for brevity reasons the proof (obtained by simulation) is not given here.

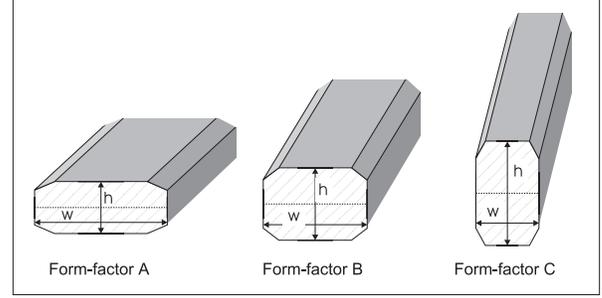


Figure 2: Shown are three possible cross section shapes for bus lines

technology and according to that a certain charge (number of electrons) have to be transferred during a switching phase, the current density $I/(h \times w)$ is a constant. Otherwise, the bus line might be destroyed through overheating.

In deep sub-micron designs this problem is solved by considering different *form-factors* (cross section shapes) as shown in Fig. 2. Though the cross section area is nearly constant, form-factor c) is more compact in terms of chip area than a) and b) but it also leads to a higher coupling capacitance $C'_{C i,i+1}$ due to increased h (see Eq. 1). Form-factor a) is not used for deep sub-micron designs because of the discussed chip area disadvantage. This is a major difference between "non deep sub-micron designs" and deep sub-micron design. Form-factor b) represents a good compromise and is being used by us. Later on we will see what this means in terms of the size of $C'_{C i,i+1}$ in relation to $C'_{B i}$.

So far, we only considered one coupling capacitance between one bus line and only one other adjacent bus line. In fact, coupling capacitances are present between *any* two bus lines (though they differ in size, of course). For example, let us assume bus lines are enumerated from 0 to $N - 1$ where N represents the total number of bus lines. The total capacitance for bus line 0 is given by the following formula:

$$C'_0 = C'_{B 0} + C'_{C 0,1} + C'_{C 0,2} + \dots + C'_{C 0,N-1}$$

In general, we can formulate:

$$C'_i = C'_{B i} + \sum_{j=0, j \neq i}^{N-1} (C'_{C i,j} \cdot s_fct(|i,j|) \cdot x_{i,j}) \quad (2)$$

The components are:

- $C'_{C i,j}$: The coupling capacitance between line i and line j .
- $s_fct(|i,j|)$: Represents the shield-factor since bus lines between any two other bus lines represent a shield thus diminishing the otherwise higher coupling capacitance.
- $x_{i,j}$:
This factor reflects the fact that there can, or cannot, be a coupling capacitance between any two lines. For explanation, let us assume that line i is switching from "low" to "high". If, at the same time, line j is also switching from "low" to "high" than there is at no point in time a

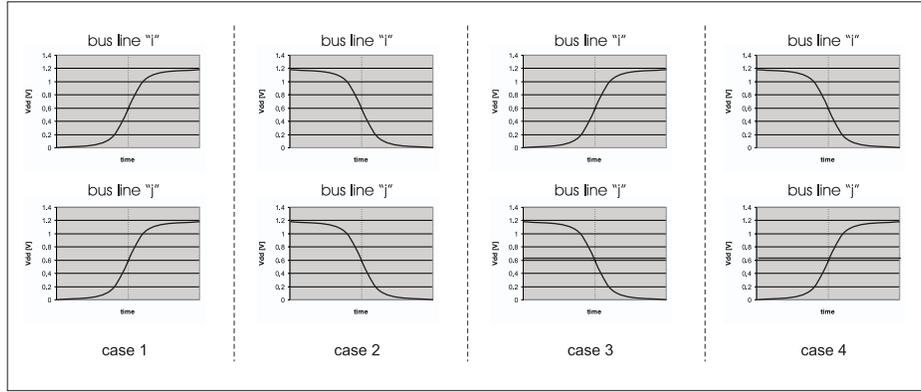


Figure 3: All relevant switching cases between bus line i and bus line j are shown (Not shown: the trivial case where i not switching)

difference in the voltage level between these two lines. Thus, line i does not "see" any coupling capacitance to line j . Neither does line j i.e. $C'_{C_{i,j}} = C'_{C_{j,i}} = 0$. Same holds if both lines have a transition from "high" to "low". However, if one line has a transition from "low" to "high" and the other has a transition from "high" to "low" (or vice versa) than there is a coupling capacitance since there is a difference in the voltage level of V_{DD} before the switching takes place. This assumes that switching takes place at the same point in time⁷. Hence, there is a coupling capacitance. All four relevant cases are shown in Fig. 3. Accordingly we obtain the following for $x_{i,j}$:

$$x_{i,j} = \begin{cases} 0 & : \text{ case 1, case 2} \\ 1 & : \text{ case 3, case 4} \end{cases} \quad (\text{see Fig. 3 for cases})$$

This bus model has various important characteristics that will be key for our encoding schemes. We will discuss the characteristics in the following.

As we can see from the factor $x_{i,j}$ of the bus capacitance model, the actual switching capacitance C'_i of a bus line i depends on:

- the behavior of all other bus lines during the switching of bus line i
- whether bus line i switches or not. In the trivial case (not shown in Fig. 3) it is not. Energy, however, is only consumed by a bus line if there is a high/low or low/high transition. In that case it is obviously important whether the temporal preceding value of the same bus line is different to the present or not.

Point b) is obvious and the basis for bus encoding techniques that have been proposed so far. However a) is new: since we do have a coupling capacitance that is in the same order of magnitude (details are given later) as the base capacitance, the existence of the portions of the coupling capacitance depends on the behavior of the other bus lines and thus it is *time dependent*.

In summary, a very important characteristic of our bus model is that the capacitance and, consequently, the energy con-

⁷We can make this assumption though physically the two signals may have a skew due to layout issues, slightly different signal speed through manufacturing tolerances etc.

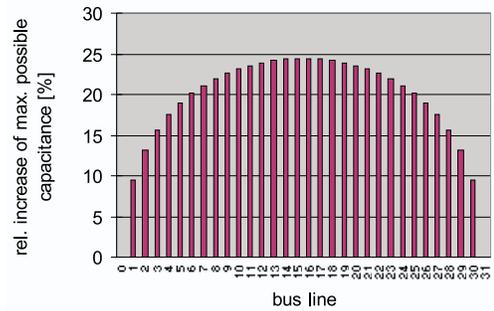


Figure 4: Capacitance profile of a 32 bit bus according to the technology shown in Table 1

sumption for transmitting information via a the bus varies depending on the activity of the other bus lines.

Table 1 gives an overview of the characteristics in terms of the actual capacitances. The values are based on a 0.1 μ m, 1.2V CMOS process and have been obtained by simulations using our bus model.

Technology Parameters and Bus Characteristics	
Technology	0.10 μ m
V_{DD}	1.2V
Base capacitance $C'_B = \min(\sum C'_i)$	42.22pF/m
$\max(\sum C'_i)$	631.3pF/m
Coupl. capac. of two adj. lines $C'_{C_{i,i+1}}$	35.89pF/m
$C'_{i \min}$	0.67pF/m
$C'_{N/2 \max}$	40.11pF/m
$C'_{(i=N/2) \max} / C'_{(i=0) \min}$ see Eq. 2 for C'_i	59.87x

Table 1: Technology parameters and bus characteristics (all capacitances are per-length)

It is interesting to see that $\max(\sum C'_i) / \min(\sum C'_i) \approx 15$ i.e. a spread 15x in energy consumption for submitting one 32-bit word via the bus. In the worst case there is:

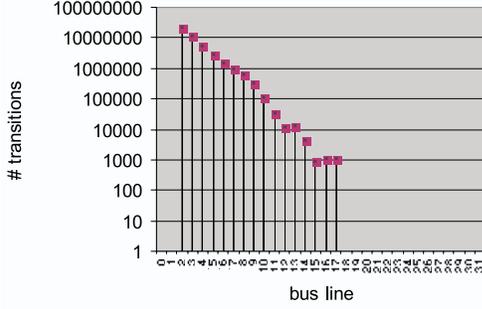


Figure 5: Shown are the bus transitions per bus line for an application (MPEGII encoder)

$$C'_{i \max} = C'_B + \sum_{\forall j \in \{0, \dots, N-1\}, j \neq i} C_{C_i}(x_{i,j}, s_fct(|i, j|))$$

with $\forall j \in \{0, \dots, N-1\}, j \neq i \quad x_{i,j} = 1$

whereas in the best case $C'_{i \min} = C'_B$ holds (when all $x_{i,j}$ equal 0). Fig. 4 shows the normalized maximum capacitance increases for all bit lines of a 32-bit bus. That is the max. capacitance of each bit line (according to Eq. 3 it is set in relationship to the smallest maximum capacitance

$$\min (\forall_{i \in \{0, \dots, N-1\}} C'_{i \max}) \quad (4)$$

of all bit lines. In Fig. 4 we actually show the increase in [%]. We can see that the max. capacitance of bit line 16 is around 25% bigger than the max. capacitance of bit line 0 or bit line 31. We exploit these and other characteristics for a low power encoding.

3 Encoding Schemes

Our goal is to minimize the power/energy consumption that is related to transmitting information via a bus. Therefore, we study the characteristics of address bus transactions as shown in Fig. 5. Obviously, the address space of the application is 2^{17} byte wide (please note that from the transitions we cannot tell *where* in the address space the program is located because those bits that never change have zero transitions). The chart (Fig. 5) shows a decreasing number of transition with increasing bus lines which looks like a counter profile. In addition, many bit lines are not used. Other applications (MPEG encoder profile) show similar characteristics with the only difference that the address space might vary (size of an application).

3.1 Adaptive Cross Connection Scheme ACCS

These observations can be used for power/energy efficient encoding since we can assign the most active bit lines to those bus lines that are expected to have the smallest capacitance (see Fig.4). Therefore, we define a *window* as

$$w_{l,h}(ww) = \{l, h \mid h - l = ww - 1, h > l, h, l \geq 0, h, l \leq bw - 1\} \quad (5)$$

with l, h being the lower and upper border bit positions of the window, respectively, ww the window size in bits and bw the bus size in bits. Thus we can define a cross connection scheme as

$$w_{target_{c,b}}(ww_1) := w_{source_{a,b}}(ww_1)$$

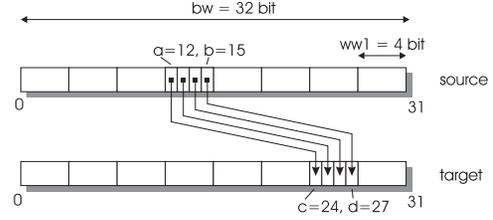


Figure 6: An example for a window definition

and an example is shown in Fig. 6. The size of a window ww is a compromise between hardware effort (more smaller windows cost more hardware) and obtaining a high shield effect (windows with high transition activities should be separated by windows with no transitions). Since different ap-

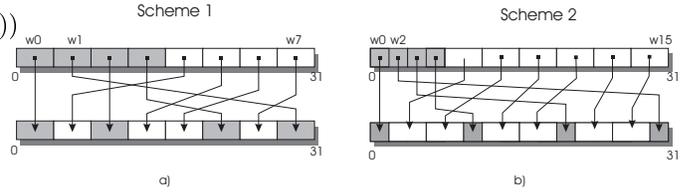


Figure 7: ACCS schemes 1 and 2 for a 32 bit bus

plications have different address spaces, we implemented two schemes, both maximizing the non-transition areas (white areas in Fig. 7) between the expected high transition areas (grey areas in Fig. 7). Which scheme is active at a certain time is decided by the operating system that knows the address space of an application/process. *Scheme1* is for large address spaces whereas *Scheme2* is for smaller ones.

3.2 Locally Spatial Invert Scheme LSIS

Through the ACCS scheme we obtained, among others, a scenario where windows with high transition activities are shielded against each other resulting in minimizing effects of coupling capacitances. However, we did not minimize the transition activities and coupling capacitance effects *within* a particular window (though we minimized the potential power/energy consumption to assign high activity windows to low capacitance areas of the bus). This is the goal of the scheme LSIS (Locally Spatial Invert Scheme) introduced in the following. Please note that LSIS is applied upon the output of ACCS.

Let us first define what we call the Extended Transition Activity Measure ETAM for a window $w_{l,h}(ww)$ (as defined in Eq. 5) In order to make the formula easier readable we simply use w to denote the window. Furthermore, let us assume that b_x is the x .th bit within a window with B_x being the value of that bit (i.e. $B_x \in \{0, 1\}$). Thus, we can define the ETAM measure as follows

$$ETAM(w) = \sum_{\forall b_i \in w} ((B_i \oplus B_i^{-1}) + (B_i \oplus B_i^{-1})) \cdot \sum_{\forall b_j \in w, b_j \neq b_i} (B_i \oplus B_j) \quad (6)$$

Thereby B_i^{-1} gives the value of bit b_i at time $t - 1$ i.e. the temporal predeccessing value. Thus, $B_i \oplus B_i^{-1}$ determines whether bit b_i has a high/low or low/high transition (=1) or not (=0). Accordingly this specific bit will contribute to the

ETAM measure or not. Fig.8 gives an idea on how ETAM is measured using an example of two stages. In the first stage the portion of the ETAM measure contributed by $i = a + 1$ is demonstrated. The dotted line shows the scope that is important for the calculation of the respective ETAM portion. It equals to 2. In the case of $i = a + 2$ (right part of Fig.8) the respective ETAM portion is 0 since the bit being viewed does not perform a transition.

It is very important to note that ETAM as shown does NOT violate the causality principle as it might seem from the Fig. 8. Therefore, please note that the bus word referring to time $t - 1$ is stored in a register. But even the bus word for time t is stored in a register since the word is not yet put on the bus (it is just in the I/O register of a device, for example) and thus ETAM does work as intended by Eq. 6.

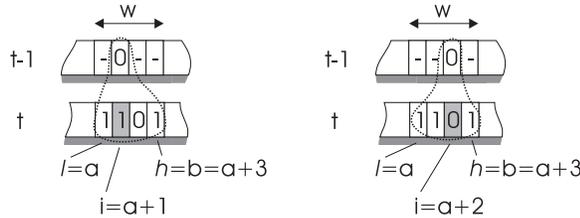


Figure 8: ETAM measure explained by means of an example.

According to Eq. 6 every value of a bit different to the bit under review is contributing 1 or 0 to the value of ETAM depending on whether it is different in value or not. That each contribution is equally sized (1 or 0 with no other values allowed) is justified by our capacitance measure that gives us values of base capacitance compared to coupling capacitances of the closest neighbors (a maximum of three left or right neighbors in a 4-bit window) that are approximately the same and thus contribute the same to the power/energy consumption. Furthermore, the shield effect makes more distant coupling capacitances negligible. This is the justification to use a window size of 4 bits. Also, this window size results in a reasonable hardware amount to implement.

In the next step we use ETAM as a measure whether we should invert the information in the window or not. Please note that our ETAM scheme is able to measure the impact of coupling capacitances. A Hamming Distance measure, as used for regular invert schemes would not lead to a reasonable improvement in power/energy consumption. It would only reduce the number of transitions. But the number of transitions do not necessarily reflect the amount of power/energy that is consumed. Our whole LSIS scheme works according to the following procedure: For all windows the ETAM measure is calculated (lines 1-2). If the ETAM measure exceeds half of the maximum value (dependent on the window size ww) then it is counted (lines 3-5). After all ETAM measures are calculated, it is determined whether more than half of the windows have a high ETAM value. If that is the case the information in the windows is transmitted inverted. This is the LSIS encoding scheme. Please note that decoding can be done inversely. Only 1 extra bit line is used for that since *all* windows will be inverted or not (majority vote).

Please note that this code explains only the strategy. It does not in any way reflect the implementation that, of course, is in hardware.

The bus interface (not shown) integrates both encoding schemes ACCS and LSIS. First, it is decided whether ACCS

Strategy of LSIS Scheme

- 1) **For All** windows $w_i \in W$
- 2) determine $ETAM(w_i)$
- 3) **If** $ETAM(w_i) > etam_{max}(ww)/2$
- 4) **Then**
- 5) $hi_etam+ = 1$
- 6)
- 7) **If** $hi_etam > (\#windows)/2$
- 8) **Then**
- 9) **For All** windows $w_i \in W$
- 10) $invert(w_i)$
- 11) done.

Figure 9: The strategy of our LSIS Scheme

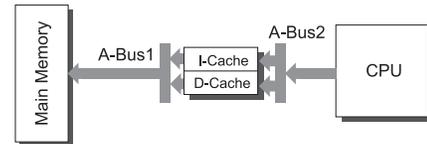


Figure 10: Generic part of application system

Scheme1 or ACCS Scheme2 is to be used. This depends on the expected size of the address space of the currently running program/process. Actually, in our case the Operating System will write the program/process size to an I/O register from where it is read by a comparator that activates either ACCS Scheme1 or ACCS Scheme2. Then, the LSIS scheme is following by counting the ETAM values for the individual windows and, by majority vote, all single windows are inverted or not. Thus, besides the encoded bus lines we have one more output that is used for the decoding side of the bus to properly decode. Please note that encoding/decoding is done on-the-fly i.e. it does not cost an additional cycle. Our current design for the bus encoding interface uses approximately 400 gates.

Within our whole encoding schemes, first ACCS is applied then LSIS. Please let us summarize. The whole scheme ACCS \circ LSIS makes use of:

- the profile of a deep sub-micron bus where it is more energy efficient to transmit information via the outer bus lines
- the general characteristic of an address bus that transmits addresses in a counter-like manner when executing a program/process
- the minimization of coupling capacitances by dividing bus lines into windows and separating those windows in order to make use of the shield effect.

In the next section we show the results obtained by applying our ACCS \circ LSIS scheme.

4 Results

We applied our encoding schemes to several SOC designs (set-top box, digital camera etc.). Fig. 10 shows the generic part of the system with a split address bus (A-Bus1, A-Bus2). The encoding scheme (Gray Code Encoding) we compare our encoding schemes to what was applied to the same systems under exactly the same conditions. Tab. 2 shows the results with the application name shown in the first column. The next column shows the instruction cache sizes. The third column

App.	IS sz.	Num. Transact.	Energy [Joule]									[%]Imp L S I S	[%]Imp (A C C S o L S I S)
			G C			L S I S			A C C S o L S I S				
			A-Bus1	A-Bus2	A-Bus1+2	A-Bus1	A-Bus2	A-Bus1+2	A-Bus1	A-Bus2	A-Bus1+2		
ISD	128	19,911	1.28e-08	1.76e-08	3.05e-08	1.25e-08	1.70e-08	2.93e-08	6.84e-09	9.44e-09	1.62e-08	-3.61	-46.60
	512		2.08e-08	4.38e-09	2.51e-08	2.00e-08	4.22e-09	2.43e-08	1.11e-08	2.33e-09	1.34e-08		
	2K		2.27e-08	1.08e-09	2.38e-08	2.19e-08	1.04e-09	2.29e-08	1.21e-08	5.78e-10	1.27e-08		
CMP	1K	23,976,781	1.95e-05	6.78e-06	2.63e-05	1.90e-05	6.60e-06	2.56e-05	1.42e-05	4.95e-06	1.92e-05	-2.70	-26.97
	2K		2.29e-05	1.15e-06	2.40e-05	2.23e-05	1.12e-06	2.34e-05	1.67e-05	8.42e-07	1.75e-05		
	8K		2.36e-05	7.44e-09	2.36e-05	2.29e-05	7.24e-09	2.29e-05	1.72e-05	5.43e-09	1.72e-05		
DIS	128	34,368	1.34e-08	3.52e-08	4.86e-08	1.29e-08	3.39e-08	4.69e-08	7.53e-09	1.98e-08	2.73e-08	-3.67	-43.16
	512		1.45e-08	3.35e-08	4.81e-08	1.40e-08	3.23e-08	4.64e-08	8.29e-09	1.90e-08	2.73e-08		
	4K		3.39e-08	1.01e-09	3.49e-08	3.27e-08	9.80e-10	3.37e-08	1.90e-08	5.71e-10	1.96e-08		
KEY	256	9,849,864	1.12e-05	3.70e-06	1.49e-05	1.12e-05	3.68e-06	1.49e-05	6.85e-06	2.25e-06	9.10e-06	-0.52	-39.28
	512		1.26e-05	1.51e-06	1.41e-05	1.25e-05	1.50e-06	1.40e-05	7.65e-06	9.18e-07	8.57e-06		
	8K		1.35e-05	7.00e-09	1.35e-05	1.34e-05	6.96e-09	1.34e-05	8.20e-06	4.25e-09	8.20e-06		
MPG	2K	22,408,513	3.42e-05	1.76e-06	3.60e-05	3.40e-05	1.75e-06	3.57e-05	2.02e-05	1.04e-06	2.13e-05	-0.67	-40.75
	4K		3.50e-05	4.40e-07	3.54e-05	3.48e-05	4.37e-07	3.52e-05	2.07e-05	2.60e-07	2.10e-05		
	16K		3.52e-05	1.44e-07	3.53e-05	3.49e-05	1.44e-07	3.51e-05	2.08e-05	8.58e-08	2.09e-05		
SMO	128	1,716,150	6.51e-07	3.39e-06	4.05e-06	6.45e-07	3.37e-06	4.01e-06	2.81e-07	1.47e-06	1.75e-06	-0.84	-56.71
	512		2.09e-06	7.55e-10	2.09e-06	2.06e-06	7.48e-10	2.06e-06	1.10e-06	3.20e-10	1.10e-06		
	4K		2.09e-06	8.82e-10	2.09e-06	2.06e-06	7.48e-10	2.06e-06	1.10e-06	3.20e-10	1.10e-06		
TRK	256	520,860	6.36e-08	1.21e-06	1.28e-06	6.25e-08	1.19e-06	1.26e-06	3.01e-08	3.78e-07	6.08e-07	-1.73	-52.55
	512		4.39e-07	6.09e-07	4.05e-06	4.21e-07	3.98e-07	4.02e-06	2.03e-07	2.89e-07	4.32e-07		
	2K		7.94e-07	9.00e-10	7.94e-07	7.80e-07	8.84e-10	7.81e-07	3.76e-07	4.27e-10	3.77e-07		

Table 2: Results of our encoding schemes (window size = 4 bit)

gives the total number of address bus transactions (this should not be confused with transitions) that have been executed. The quality measure of our encoding scheme is the energy that is consumed on the addresses buses compared to energy efficient address coding schemes like Gray Coding (GC). It is important to notice that unlike other work we do not measure the quality of our encoding schemes in terms of the number of total transitions since they do not reflect power/energy consumption through coupling effects.

We have examined three encoding schemes: the first one is a Gray Encoding scheme that we use as the standard to compare our results to. The second is our ACCS and the third one is ACCS o LSIS (LSIS applied on top of ACCS). For each encoding scheme the energy consumption is shown, split into the parts consumed by A-Bus1, A-Bus2 and the total (A-Bus1+2). The energy of one single bus transaction is calculated through:

$$E = 1/2 \cdot \sum_{i=0}^{N-1} (C'_i \cdot L_{Bus}) \cdot V_{DD}^2$$

with L_{Bus} being the length of the respective bus (please remember that C'_i represents the per-length capacitance) with C'_i calculated according to Eq. 2). Finally, the last column gives the percentage in energy savings of ACCS o LSIS compared to Gray Code encoding. We yield energy savings (same holds for power savings) of up to 56% with an average of 44%. The column left to that shows intermediate results i.e. only ACCS applied. Not surprisingly, ACCS does not directly contribute much to the final results. The main purpose of ACCS is to provide a higher optimization potential for LSIS that is applied afterwards. In this sense, ACCS contributes implicitly more than the numbers can express.

We observe that the energy savings are quite high compared to ordinary schemes (like the Gray Code encoding that is the benchmark encoding scheme for address buses). The reason is that ordinary schemes do not take into considerations power/energy consumption through coupling capacitances and thus cannot optimize for it.

There is a limitation to our approach: our approach is effective only on high capacitance buses since the encoding schemes implemented as hardware also consume power/energy that could otherwise exceed the savings. Note that this limitation only excludes the application of our schemes to small microprocessors that have small internal buses. However, for typical SOCs that feature long buses connecting various cores, our method is very efficient.

5 Conclusions

We have presented a novel adaptive address bus encoding scheme for low power deep sub-micron designs. Unlike ordinary schemes, the scheme is based on our physical bus model that takes coupling capacitance effects into account. The scheme is applied in the two stages ACCS and LSIS. Together they eventually lead to power/energy savings of up to 56% compared to the Gray Encoding scheme that is considered the best low transition encoding scheme for address buses.

References

- [1] F.N. Najm, "Transition Density: A New Measure of Activity in Digital Circuits", IEEE Tr. on CAD, Vol 12, No. 2, pp. 310–323, Feb. 1993.
- [2] M.R. Stan, W.P. Burleson, "Bus-Invert Coding for Low-Power I/O", IEEE Tr. on VLSI Systems, Vol 3, No. 1, pp. 49–58, March 1995.
- [3] P.R. Panda, N.D. Dutt, "Low-Power Memory Mapping Through Reducing Address Bus Activity", IEEE Tr. on VLSI Systems, Vol 7, No. 3, pp. 309–320, Sept. 1999.
- [4] P.P. Sotiriadis, A. Chandrakasan, "Low Power Bus Coding Techniques Considering Inter-wire Capacitances", Proc. of IEEE Conf. on Custom Integrated Circuits (CICC'00), pp.507–510, 2000.
- [5] L. Benini, A. Macii, E. Macii, M. Poncino, R. Scarsi, "Synthesis of Low-Overhead Interfaces for Power-Efficient Communication over Wide Buses", Proc. of IEEE 36th. Design Automation Conf. (DAC'99), pp.128–133, 1999.
- [6] H. Mehta, R.M. Owens, M.J. Irwin, "Some issues in gray code addressing", Proc. of IEEE Conf. on 6th. Great Lakes Symp. on VLSI, pp.178–181, 1996.
- [7] C.L. Su, C.Y. Tsui, "Saving Power in the Control Path of Embedded Processors", IEEE Design & Test Magazine, Vol. 11, No. 4, pp.24–31, Winter 1994.
- [8] L. Benini, G. De Micheli, E. Macii, D. Sciuto, C. Silvano, "Asymptotic Zero-Transition Activity Encoding for Address Buses in Low-Power Microprocessor-Based Systems", Proc. of IEEE Conf. on 7th. Great Lakes Symp. on VLSI, pp.77–82, 1997.
- [9] E. Musoll, T. Lang, J. Cortadella, "Working-Zone Encoding for Reducing the Energy in Microprocessor Address Buses", IEEE Tr. on VLSI Systems, Vol 6, No. 4, pp.568–572, Dec. 1998.
- [10] A. Acquaviva, R. Scarsi, "A Spatially-Adaptive Bus-Interface for Low-Switching Communication", Proc. of IEEE Int'l Symposium on Low Power Electronics and Design (ISLPED00), pp.238–240, 2000.
- [11] Y. Zhang, W. Ye, M.J. Irwin, "An alternative architecture for on-chip global interconnect: segmented bus power modeling", Conf. Record (Signals, Systems & Computers) of 32nd. Asilomar Conf. pp.1062–1065, 1998.
- [12] M.R. Stan, W.P. Burleson, "Low-Power Encodings for Global Communication in CMOS VLSI", IEEE Tr. on VLSI Systems, Vol 5, No. 4, pp.444–455, Dec. 1997.