ETAM++: Extended Transition Activity Measure for Low Power Address Bus Designs

Haris Lekatsas and Jörg Henkel
NEC USA
{lekatsas, henkel}@nec-lab.com

Abstract

Interconnection networks in Systems-On-Chip begin to have a non-negligible impact on the power consumption of a whole system. This is because of increasing inter-wire capacitances that are in the same order of magnitude as intrinsic capacitances as far as deep-submicron designs are concerned. This trend has been recognized in recent research work. In this work, we present a physical model that takes into account inter-wire capacitances. Subsequently we propose a novel encoding scheme based on this physical model and targeted for address buses. We demonstrate that our encoding method improves power consumption by up to 62.5% and thus is exceeding all current approaches including our own previous one. In addition, the hardware of the bus encoding/decoding interfaces is compact to implement. We have conducted extensive simulations using SOC applications like, for example, an MPEGII encoder to evaluate the advantages of our approach.

1. Introduction

With the advent of Systems-on-Chip (SOC) that will reach 1 billion transistors within the next couple of years, the complexity and the physical length of bus systems/hierarchies will lead to an increased contribution to a chip's total power consumption. And most importantly, the closer geometrical proximity of adjacent bus lines will lead to effects that are almost negligible in technologies not advanced as 0.1micron and beyond. This is because two or more close bus lines will form a parasitic capacitance between them. This effect not only leads to cross-talk and delay effects it also leads to an increased power consumption since the parasitic capacitance is charged and discharged when there is a voltage swing between two or more bus lines. This effect takes place *in addition* to the intrinsic capacitance of a bus line i.e. the parasitic capacitance between the bus line and various metal layers beneath. Hence, more energy is being consumed.

There are several ways to diminish or at least reduce the problem of inter-wire capacitances. One way is to widen the distance between bus lines. However, this is typically not preferred since the total area of the bus systems grows too large. Another option is to use P&R tools (*place & route*) that avoid side-by-side routing of bus lines. This is what is actually done in the newest generation of P&R tools. However, the interconnect complexity of a 1-billion transistor SOCs with multiple bus hierarchies and long buses with many cores connected to them will prevent a satisfying solution at a feasible routing time (complexity of the routing problem). A third option is to change the geometrical shape of bus lines: the bus lines themselves can be re-shaped. For example the cross-sectional shape can be made narrower such that the distance between two bus lines increases *without* sacrificing space for the whole bus. However, the main disadvantage of this approach is that the cross-sectional area of a bus line is fixed since the *current-per-area* ratio is fixed for any certain technology. That typically leads to solutions where the bus line is buried deeper into the substrate with the height being larger than the width of a bus line. However, even though the inter-wire capacitance decreases due to a decreasing distance between bus lines, it *does increase* due the increased flank area of two opposing bus lines. In conclusion: what is won through a wider distance has to be, at least partly, given up through the effect of larger flank area.

Finally, another technique to reduce power due to inter-wire capacitances is through the use of bus encoding techniques. In our research we focus on such a technique, namely on finding an energy-efficient bus encoding technique. The reason is that a bus encoding technique can be applied in addition to other techniques discussed above. We will furthermore show that our approach delivers higher energy savings than any other bus encoding technique proposed so far. Before presenting our encoding method, we will discuss in detail the physical model that forms the basis of our approach. Unlike most previous work for bus encoding, we take into consideration both the intrinsic and the inter-wire capacitances and present a model that is more accurate compared to our previous work [16]. Our work focuses on address buses and the proposed encoding scheme takes advantage of the special characteristics of address bus transactions. However, no a priori knowledge of the application is necessary. We have conducted extensive experiments and found high energy savings across different application domains.

This paper is structured as follows: Section 2 discusses related work in the area. Section 3 provides an introduction to our techniques and describes our previous method as discussed elsewhere [16]. Section 4 proposes the new physical model estimating the power consumption of bus lines and taking into accounts coupling (i.e. inter-wire) effects. Section 5 describes our ETAM++ scheme, which is used to selectively invert words transmitted on the address bus. Section 6 presents experimental results, while Section 7 concludes.

2. Related Work

In recent research the above-mentioned trend of the increasing importance of interconnect in terms of power consumption has been recognized. In the following we will discuss basically two groups of related work: first, work that tries to minimize the number of transitions on buses to reduce power assuming that inter-wire capacitances are negligible and secondly a group of work that assumes that inter-wire capacitances do matter. The latter group applies to the newest technologies and thus is most relevant to our work.

Early work on minimizing the transition activities on buses has been conducted by Stan/Burleson [11]. The idea is to transmit the inverted word through the bus when the *Hamming Distance* (HD for simplicity) of the non-inverted word would result in HD > N/2 with N being the number of bus lines. This requires minimal additional logic only, plus one control bus line that tells whether invert mode is being applied for a particular transition or not. Panda/Dutt [7] approached the problem of reducing switching activities of address busses by exploiting the characteristics of accesses to memory arrays. They investigated various scenarios for memory mapping schemes due to different memory organizations.

Benini et al. [2] present an adaptive approach for encoding signals that are transmitted through wide and heavily loaded buses. The exploitation of correlated access patterns (like in address buses) has been studied in (see above) by using Gray Code encoding according to Metha et al. [6] and Su et al. [14]. Benini et al. [3] have improved upon Gray Code through their method called T0 that benefits from the fact that a fairly high number of patterns in address buses are consecutive. Then, the receiving side of an address bus can calculate the address without the necessity to actually having the address code being transmitted via the address bus. Working Zone Encoding has been proposed by Musoll et al. [6]. They perform encoding adjusted 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 [1] that does not need any a priori knowledge of the data being transferred. Ramprasad et al. [8] present a framework to study various encoding schemes for address and data buses that can be applied to high capacitance buses. The approach of Zhang et al. [15], is to segment a bus and thus exploit the effect of having smaller effective bus capacitances applying during bus transitions. Another system-level oriented approach for communication architectures is presented by Stan/Burleson [12] as they focus on low power encoding techniques under specific consideration of influences on possible area and performance impacts.

The following work consists of the most recent approaches that take inter-wire capacitances into consideration. Sotiriadis/Chandrakasan [10] use a static encoding technique (i.e. an encoding technique that is fixed) and obtain results of an average of 40% power savings. Another approach is introduced by Kim et al. [4] where a coupling-sensitive invert scheme is introduced leading to around 30% power savings. The work of Shin/Sakurai [9] presents a coupling driven bus encoder that capitalizes on the fact that the data sent via the bus might be known a priori. They report for those cases up to 46% energy savings. Taylor et al. [13] present an approach to model the power consumption of interconnects.

We can summarize the related work with respect to our work as follows: there have been many approaches to bus transition reduction resulting in significant reduction in bus power consumption. However, reducing transition activity i.e. the number of low/high, high/low transitions is not necessarily leading to low power consumption in deep sub-micron designs as we will show throughout the course of this paper. Deep sub-micron characteristics of buses and the exploitation of these effects are just starting to be taken into consideration, since designs with those characteristics inter-wire capacitances are in the same order of magnitude as intrinsic capacitances are to be launched in the not so far future. As opposed to other approaches (see above) taking this effect into consideration, our approach has additional features: our coding scheme is adaptive and thus it can exploit characteristics on the (address) bus that are changing over time. As opposed to [10] we do not assume any a priori knowledge of the application running on the system. Furthermore, we are able to improve power consumption by up to 62.5%.

3. Assumptions and Previous Work

The work presented in this paper presents a major improvement over our previous work. In this paper we present a new technique that is used as the basis for a low-energy-consumption **Bus Encoding Interface (BEI)**. For a proper understanding of this paper the knowledge and principle of our previous work is crucial and therefore the main ideas are summarized here.

The following assumptions/observations hold:

A. In deep-submicron designs of 0.1u and beyond there are inter-wire capacitances i.e. capacitances between bus lines that will consume energy when transitioning from 0->1 or 1->0. These capacitances are in the same order of magnitude than intrinsic capacitances i.e. the capacitances between single bus lines and the various metal layers [16].

B. Since there are capacitances between wires (and not only between adjacent wires but also, at least to a smaller degree, between a wire 'A' and a wire 'C' with a wire 'B' in between, for example) every wire has a different effective capacitance to switch: those wires with no or less adjacent wires have a smaller capacitance to switch; those wires with two or more adjacent wires have a higher capacitance to switch. This leads to a different energy consumption depending on where (i.e. which bus line) the information is being transferred. This work concentrates on buses. One of the most important characteristics of address buses, as far as this work is concerned, is the fact that lower bits tend to switch more often than higher bits. Thus, the profile of an address bus has similarities with a counter. In our previous work we exploited this effect via a scheme we call **ACCS**. In the work presented here, we apply our bus encoding method on top of ACCS i.e. temporally after ACCS has been applied.

Figure 1 shows the whole bus encoding scheme. On the left side the addresses to be encoded enter the BEI. Address bits are re-shuffled via the ACCS scheme such that bus lines with a lesser amount of transitions (according to assumption/observation *B* from above) are transmitted through bus lines with a lower effective capacitance.

Then we apply an encoding of the incoming address words. The most crucial part is to find a measure that can, on-the-fly, predict what is the least energy consuming encoding. That is what we call **ETAM++ i.e. Extended Transition Activity Measure**. It is a measure that does not only count the number of transitions but that also accounts for the inter-wire effects. ETAM++ will then control the invert encoder that actually does the selective encoding.

The work presented here focuses on ETAM++ since it is the heart of our scheme that makes the high-energy savings of our bus encoding interface possible. The next section will explain and derive the ETAM++ scheme in detail.

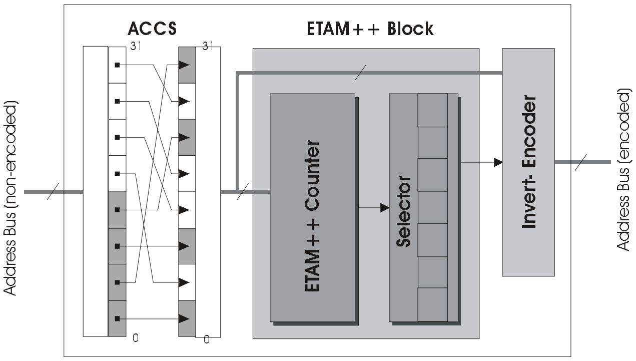

Figure 1: Bus encoding interface (BEI)

4. Inter-wire Capacitance Model

In this section we describe the physical model we use to calculate power consumption due to inter-wire (i.e. cross-coupling) capacitances. This leads us to what we call the ETAM++ definition. We will show how to use ETAM++ to minimize power consumption by selectively inverting the words transmitted on the address bus.

In Figure 2 two bus lines, i and j, are shown. C_B is the intrinsic capacitance (base capacitance) between each bus line and the ground (i.e. underlying metal layers) and C_{ij} is the coupling capacitance between these two bus lines i and j. Note that these bus lines are not necessarily spatially next to each other (though we have found that those who are spatially next to each other have the largest coupling capacitances between them). E_i is the voltage applied to the bitline and can either be equal to V_{DD} or 0, depending on the logical value that is transmitted via the bus. The R's are resistances on which energy is dissipated. We will now derive the power consumed during bit transitions based on this model. It is the first step towards deriving the ETAM++ scheme.

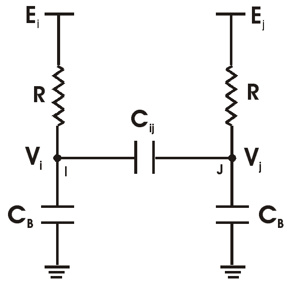

Figure 2: Model diagram

We will derive the scheme for what we call a **window**. A window in our definition is a contiguous series of bits (bus lines) on the bus. The reason is that within one window inter-wire capacitances do have an impact whereas inter-wire capacitances between bus lines of a certain window and a bus lines outside of that window is negligible. We have found (not shown here) that four bits for a window are an adequate size as a compromise between the effort for deriving the ETAM++ scheme on the one side and the amount of the inter-wire effects on the other side. Note that the whole bus is partitioned into such 4-bit windows and to each of these windows the ETAM++ scheme is applied.

Assume that the bit values in our 4-bit window are B_0, B_1, B_2, and B_3 respectively. In the previous cycle (temporally preceding) the corresponding values were B_0^{-1}, B_1^{-1}, B_2^{-1}, and B_3^{-1} respectively. The voltages E_i are assumed to remain constant during the transition period and take the value dictated by B_i. Assuming the voltages on the bus lines are $V_i(t)$ as shown in Fig. 1, the following equations hold:

$$E_i = V_{DD} * B_i \quad (1) \qquad V_i(0) = V_{DD} * B_i^{-1} \quad (2)$$

$$V_i(\infty) = V_{DD} * B_i \qquad (3)$$

Equation (1) shows that for any t>0 E_i is either zero or equal to $V_{DD,}$ depending on the new value of bit B_i. Equation (2) shows that the voltage V_i of each bus line i is initially, i.e. at time t=0, equal to the previous bit value B_0^{-1}. Finally Equation (3) shows that at time t=∞ we assume that voltage V_i settles down to the value of E_i.

By applying Kirchhoff's law, we find that all bus line voltages $V_i(t)$ satisfy the following differential equations:

$$\frac{E_i - V_i}{R} = C_B \frac{dV_i}{dt} + \sum_{j \neq i} C_{ij} \frac{d(V_i - V_j)}{dt} \qquad (4)$$

Equation (4) shows current inflow equilibrium in node i (Kirchhoff's law).

Let

$$C_i = \sum_{j \neq i} C_{ij} \quad (5), \quad C_{ii} = -(C_B + C_i) \quad (6), \quad Tij = - R * Cij \quad (7)$$

Note that the negative sign in Eq. (6) for C_{ii} has no physical meaning. We define as a mathematical tool to simplify our final results. Also note that matrix T_{ij} is symmetric i.e. $T_{ij} = T_{ji}$. Using Eq. (5), (6), (7) and (4) we derive:

$$\frac{E_i - V_i}{R} = -C_{ii}\frac{dV_i}{dt} - \sum_{j \neq i} C_{ij}\frac{dV_j}{dt} = -\sum_j C_{ij}\frac{dV_j}{dt} \quad \Rightarrow$$

$$E_i - V_i = \sum_j T_{ij}\frac{dV_j}{dt} \qquad (8)$$

Power is proportional to the square of voltage, thus Eq. (8) can be rewritten as:

$$\frac{(E_i - V_i)^2}{R} = \frac{1}{R}\sum_j T_{ij}(E_i - V_i)\frac{dV_j}{dt} \qquad (9)$$

Taking into account that all Ej's are constant and summing over all bus lines, we obtain the equation giving the power dissipated on all resistances R:

$$\sum_i \frac{(E_i - V_i)^2}{R} = -\frac{1}{R}\sum_i \sum_j T_{ij}(E_i - V_i)\frac{d(E_j - V_j)}{dt} \qquad (10)$$

We integrate equation (10) to obtain the total energy dissipated:

$$P = \sum_i \int_0^\infty \frac{(E_i - V_i)^2}{R}dt = -\frac{1}{R}\sum_i \sum_j T_{ij}\int_0^\infty (E_i - V_i)\frac{d(E_j - V_j)}{dt}dt \qquad (11)$$

Integrating by parts the integral in Eq. (11)

$$\int_0^\infty (E_i - V_i)\frac{d(E_j - V_j)}{dt}dt = (E_i - V_i)*(E_j - V_j)\Big]_0^\infty -$$
$$\int_0^\infty (E_j - V_j)\frac{d(E_i - V_i)}{dt}dt \quad (12)$$

and taking into account conditions (1) and (3), we obtain the following result:

$$\int_0^\infty (E_i - V_i)\frac{d(E_j - V_j)}{dt}dt = -(E_i - V_i(0))*(E_j - V_j(0)) -$$
$$-\int_0^\infty (E_j - V_j)\frac{d(E_i - V_i)}{dt}dt \quad (13)$$

Using Eq. (13) and the fact that matrix T_{ij} is symmetric it is very simple to prove that:

$$P = \frac{1}{2*R}\sum_i \sum_j T_{ij}\{(E_i - V_i(0))*(E_j - V_j(0))\} \qquad (14)$$

Using equations (1) and (2) as well as (9) we obtain the following final result:

$$P = -\frac{V_{DD}^2}{2}\sum_i \sum_j C_{ij}\{(B_i - B_i^{-1})*(B_j - B_j^{-1})\} \qquad (15)$$

Equation (15) calculates total power P dissipated during the transition period. As expected P is a function of B_i and B_i^{-1}. Note that this equation does not depend on the window size. In the following we will analyze the 4-bit window case, for which our experiments exhibit the best results in terms of power reduction.

5. ETAM++ Scheme

In this section we will specialize the general physical power model presented above for a 4-bit window (i.e four adjacent bus lines). As mentioned above our simulations have shown that partitioning a 32-bit bus into 8 sub-groups where each sub-group is a 4-bit window is advantageous. A discussion of this is beyond the scope of this paper, and the reader should refer to [16] for more information.

In the 4-bit window (we will just use 'window' in the following for brevity) lets look at the capacitance from the point of view of one of the four bit lines. Assume C_B is the intrinsic capacitance while C_A,C_C, and C_D are the capacitances between this bus line and its three neighbors. Using the definitions of equations (5) and (6) from above we can define the capacitance matrix C_{ij} as follows:

$$[C_{ij}] = \begin{bmatrix} -(C_B+C_A+C_C+C_D) & C_A & C_C & C_D \\ C_A & -(C_B+2*C_A+C_C) & C_A & C_C \\ C_C & C_A & -(C_B+2*C_A+C_C) & C_A \\ C_D & C_C & C_A & -(C_B+C_A+C_C+C_D) \end{bmatrix} \qquad (16)$$

In order to simplify our power scheme we make the assumption that all coupling capacitances are of the same order of magnitude (this is justified by physical sizes of the actual capacitances we derived through simulation; see [16]). We thus assume that $C_A = C_B = C_C = C_D = C$. Using equations (15) and (16) we obtain:

$$[C_{ij}] = \begin{bmatrix} -4*C & C & C & C \\ C & -4*C & C & C \\ C & C & -4*C & C \\ C & C & C & -4*C \end{bmatrix} \qquad (18)$$

and

$$P = \frac{C*V_{DD}^2}{2}\Big\{4*(B_0 - B_0^{-1})^2 + 4*(B_1 - B_1^{-1})^2 + 4*(B_2 - B_2^{-1})^2 +$$
$$4*(B_3 - B_3^{-1})^2 - 2*(B_0 - B_0^{-1})*(B_1 - B_1^{-1}) - 2*(B_0 - B_0^{-1})*$$
$$(B_2 - B_2^{-1}) - 2*(B_0 - B_0^{-1})*(B_3 - B_3^{-1}) - 2*(B_1 - B_1^{-1})*(B_2 - B_2^{-1})$$
$$-2*(B_1 - B_1^{-1})*(B_3 - B_3^{-1}) - 2*(B_2 - B_2^{-1})*(B_3 - B_3^{-1})\Big\} \quad (19)$$

Equation (19) can be transformed to:

$$P = \frac{C * V_{DD}^2}{2} \Big\{ (B_0 - B_0^{-1})^2 + (B_1 - B_1^{-1})^2 + (B_2 - B_2^{-1})^2 + (B_3 - B_3^{-1})^2 + [(B_0 - B_0^{-1}) - (B_1 - B_1^{-1})]^2 + [(B_0 - B_0^{-1}) - (B_2 - B_2^{-1})]^2 + [(B_0 - B_0^{-1}) - (B_3 - B_3^{-1})]^2 + [(B_1 - B_1^{-1}) - (B_2 - B_2^{-1})]^2 + [(B_1 - B_1^{-1}) - (B_3 - B_3^{-1})]^2 + [(B_2 - B_2^{-1}) - (B_3 - B_3^{-1})]^2 \Big\} \quad \textit{(20)}$$

We shall use the following definitions and identities:

$$r_i = B_i \oplus B_i^{-1} = (B_i - B_i^{-1})^2 \qquad \textit{(21)}$$

$$\textit{and} \qquad [(B_i - B_i^{-1}) - (B_j - B_j^{-1})]^2 = r_i \oplus r_j + 4 * d_{ij} \qquad \textit{(22)}$$

where

$$d_{ij} = \overline{B}_i B_i^{-1} B_j \overline{B}_j^{-1} \bigcup B_i \overline{B}_i^{-1} \overline{B}_j B_j^{-1} \qquad \textit{(23)}$$

The meaning of r_i and d_{ij} is:

- r_i is 1 iff there is a change of bit i. It does not contain any information concerning the direction of the change (0 to 1 or 1 to 0).
- d_{ij} is 1 iff both bits, i and j, change but in the **opposite** direction. In such a case the voltage difference across the coupling capacitance is double and when squared it results in power 4 times as high compared with the other case described in next bullet. This explains factor 4 appearing in eq. 22.
- $r_i \oplus r_j$, appearing in eq.(22) is 1 iff only one of the two bits is changing. Note that if both bits are changing in the **same** direction then the voltage difference across the coupling capacitance is zero.

Using the above definitions we obtain the following:

$$P = \frac{C*V_{DD}^2}{2} \Big\{ r_0 + r_1 + r_2 + r_3 + r_0 \oplus r_1 + r_0 \oplus r_2 + r_0 \oplus r_3 + r_1 \oplus r_2 + r_1 \oplus r_3 + r_2 \oplus r_3 + 4*(d_{01} + d_{02} + d_{03} + d_{12} + d_{13} + d_{23}) \Big\} \qquad \textit{(24)}$$

Apart from V_{DD} and C, all symbols appearing in the above equation are functions of the four input bits B_i and their previous values B_i^{-1}. We can therefore introduce a measure of the power dissipated as follows:

$$ETAM{+}{+} = \Big\{ r_0 + r_1 + r_2 + r_3 + r_0 \oplus r_1 + r_0 \oplus r_2 + r_0 \oplus r_3 + r_1 \oplus r_2 + r_1 \oplus r_3 + r_2 \oplus r_3 + 4 * (d_{01} + d_{02} + d_{03} + d_{12} + d_{13} + d_{23}) \Big\} \qquad \textit{(25)}$$

Note that ETAM++ is used as a scheme to measure power in a way that is easy to implement within a small logic (otherwise the power savings we are looking for on the bus would be eaten up by this additional logic). This logic is implemented for each window and it decides whether inverting the address data on the bus in beneficial in terms of power consumption or not. Note that our definition of ETAM++ in this paper is quite different than the scheme presented in our previous work [16] as it is based on a different (more accurate and easier to implement) physical bus model. As will become evident in the experimental section this new model yields significantly better results.

The ETAM++ takes the following values:

1. If no bit changes, i.e. r_i=0 ∀i, then ETAM++ =0
2. If only one bit changes, e.g. r_0 is 1 and r_1=r_2=r_3=0, then ETAM++ =1+ 3=4.
3. If two bits change, e.g. r_0 and r_2 are 1 and r_1=r_3=0, then ETAM++=6+4*d_{02}. There are two sub-cases:
   A. The 2 bits change in the **same** direction i.e. d_{02}=0 and ETAM++=6.
   B. The 2 bits change in the **opposite** direction i.e. d_{02}=1 and ETAM++=10
4. If 3 bits change, e.g. $r_{0,}$ r_2 and r_3 are 1 and r_1=0, then ETAM++=6+4*(d_{02}+d_{03}+d_{23})
   There are two sub-cases:
   A. All 3 bits change in the **same** direction i.e. d_{02}+d_{03}+d_{23}=0, then ETAM++=6.
   B. One bit changes in the **opposite** direction e.g. d_{03}=1, d_{23}=1 and d_{02}=0, then ETAM++=10.
5. If all 4 bits change, i.e. r_i=0 ∀i, then ETAM++=4+4*(d_{01}+d_{02}+d_{03}+d_{12}+d_{13}+d_{23})
   There are three sub-cases:
   A. All 4 bits change in the **same** direction i.e. all d_{ij}=0, then ETAM++=4
   B. One bit changes in the **opposite** direction compared with the other 3, then ETAM++=16.
   C. Two bits change in the opposite direction with respect to the other two, then ETAM++=20.

Figure 3 illustrates all possible cases as described above. It should be clear from the above discussion that when bits change in opposite directions there is higher ETAM++ value generated. This has a physical explanation: the voltage difference between such lines is double the voltage of one line changing and the other remaining constant, therefore the power consumed on the resistances due to this voltage difference is quadrupled.

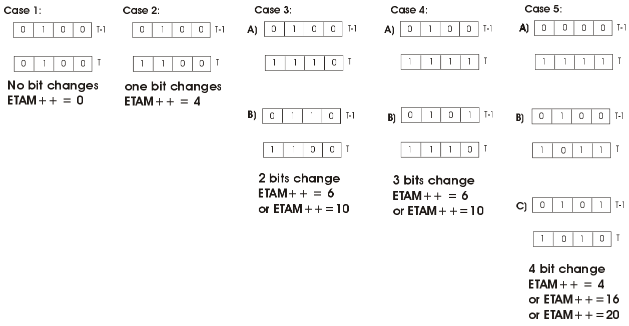

Figure 3: ETAM cases depending on window values at time =T-1 and T

The following pseudo code summarizes our whole bus encoding strategy:

```
BUS ENCODING SIDE:

Invert_flag = 0
foreach window w:
        e = ETAM++(w)     // defined by equation (25)
        if (e>ETAM++_MAX/2):
                invert_flag = invert_flag + 1
if (invert > NUM_OF_WINDOWS/2):
        invert_bus_word()
        invert_line = 1
else:
        invert_line = 0
transmit_bus_word()


BUS DECODING SIDE:

if (invert_line = 1):
        invert_bus_word()
```

We use this ETAM measure as an indication to invert a word (32 bits) or not. The maximum attainable value for the ETAM value is MAX = 20. Our algorithm works as follows: An ETAM value is measured for every 4-bit window. If the majority of these ETAM values are more than 10, i.e. more than its MAX/2 value, the word is inverted; otherwise it is transmitted as is. The decoder part of the design will know if the word has been inverted by using an extra line.

In addition, the whole BEI (Bus Ending Interface) is shown in Figure 1. As implemented in hardware it costs approximately 2,500 gates.

6. Experimental Results

For our experimental results we have deployed the architecture such as shown in Figure 4. It contains a CPU, a unified data & instruction cache and main memory and a bus system in between. We assume that the address bus is split into two parts, Bus 1 and Bus 2, where Bus 1 connects the cache with the main memory while Bus1 connects the cache to the CPU. Typically, Bus 1 will have a smaller length as the cache tends to be placed closer to the CPU than main memory. This may be beneficial in terms of power as well as performance, as we expect more traffic on Bus 2 than on Bus 1[1]. Note that although we used this particular architecture for our experiments, our method does not depend on this architecture and we expect similar improvements for a single non-split bus. In our experiments we report power on both Bus1 and Bus2 for various bus encoding schemes. The experiments have been conducted with the SOC power estimation framework.

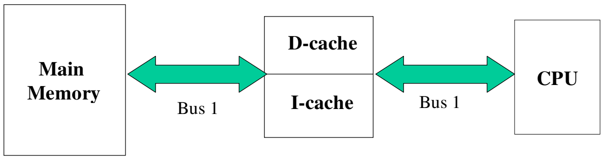

Figure 4: Basic Architecture

Unlike previous work in the area, our schemes do not only minimize the number of bus-line transitions since this is not an accurate measure of power consumption in deep sub-micron design where inter-wire capacitances matter.
In our case a considerable amount of power/energy is consumed through inter-wire (coupling) capacitances. For comparison purposes we present experiments that compare our schemes to Gray Encoding, which is the best known encoding method for address bus power minimization. We also show results for the ETAM scheme (this is our previous work and is based on a different physical model and results in a completely different logic implementation [16]). For each of the three schemes Gray Code, ETAM, ETAM++ the power consumption on Bus1 and Bus2 are presented, as well as the total bus power consumption on both buses (Bus1+2). The energy of one single bus transaction is:

$$Energy = \frac{1}{2} \cdot \sum_{i=1}^{N-1} (C_i \cdot Length\ (bus)) \cdot V_{DD}^2 \qquad (26)$$

where C_i is the per-length capacitance of a bus line. The length of the bus lines are taken from our recent settop box chip design.

We used the SOC power estimation framework to obtain the traces on Bus1 and Bus2. Table 1 shows experimental

[1] Assuming the application is well suited for the chosen cache size.

results for 7 different applications that range in size from about 10kB to 250kB of C code executable specification. The leftmost column shows the applications:. I3D is an image processing application, CMP is the UNIX compress program, DIS is a diesel engine control controller, KEY is an HDTV chroma-key algorithm, MPG is a whole MPEGII encoder, SMO a smoothing algorithm for digital images, and finally TRK is a trick animation algorithm for digital video sequences.

Appl.	Num. of Transact.	Energy [Joule]									ETAM++ energy sav.[%]
		Gray Code Encoding			ETAM			ETAM++			
		Bus-1	Bus-2	Bus1+2	Bus-1	Bus-2	Bus1+2	Bus-1	Bus-2	Bus1+2	
I3D	19,911	2.08e-08	4.38e-09	2.51e-08	1.11e-08	2.33e-09	1.34e-08	9.70e-09	2.05e-09	1.18e-08	-52.99
CMP	23,976,781	1.95e-05	6.78e-06	2.63e-05	1.42e-05	4.95e-06	1.92e-05	1.25e-05	4.21e-06	1.67e-05	-36.50
DIS	34,368	1.45e-08	3.35e-08	4.81e-08	8.29e-09	1.90e-08	2.73e-08	7.21e-09	1.63e-08	2.35e-08	-51.14
KEY	9,49,864	1.35e-05	7.00e-09	1.35e-05	8.20e-06	4.25e-09	8.20e-06	7.22e-06	3.66e-09	7.22e-06	-46.51
MPG	22,408,513	3.52e-05	1.44e-07	3.53e-05	2.08e-05	8.58e-08	2.09e-05	1.83e-05	7.46e-08	1.83e-05	-48.16
SMO	1,716,150	6.51e-07	3.39-e06	4.05e-06	2.81e-07	1.47e-06	1.75e-06	2.44e-07	1.28e-06	1.52e-06	-62.47
TRK	520,860	7.94e-07	9.00e-10	7.94e-07	3.76e-07	4.27e-10	3.77e-07	3.27e-07	3.71e-10	3.27e-07	-58.81

Table 1: Power savings results of ETAM++ compared to Gray Code Encoding and our previous work

.

The second column gives the total number of address bus transactions that have been applied. Since some of the applications are periodic tasks we examined a run through one period. When that resulted in overly long traces (like it was the case for the MPEGII encoder) we applied a representative part of it (e.g. for the MPEGII encoder we used the traces referring to 6 frames which resulted already in a 300MB trace that took about a day to generate). The following columns give results for Bus-1, Bus-2 and Bus1+2 for GC (Gray Coding), ETAM and ETAM++.

As can be seen we yield energy savings (same holds for power savings) of up to 62.5% with an average of 51%, compared to Gray Code Encoding that is the benchmark for address bus encoding schemes.

The reason is that ordinary schemes do not take into considerations power/energy consumption through coupling capacitances and thus cannot optimize for it. Our model gives us detailed information about the sizes of coupling capacitances and base capacitances.

Figure 5 illustrates the improvement achieved using our method as well as the improvement achieved by our previous work [16]. As before, results are also compared with Gray Coding. For ETAM we obtain a maximum reduction of 56.71% and an average reduction of 44% over Gray Coding. ETAM++ gives a maximum reduction of 62.5% averaging a 51%. This is also more than what is reported in recent work on bus encoding schemes that also take inter-wire capacitances into consideration (see Section ‘Introduction’). As explained in Section 4 it is possible to further improve the accuracy of the model by taking into account the distance between bus lines when measuring capacitances. However, solving the corresponding equations is considerably more complex and would result in a hardware implementation too expensive for a bus encoding interface.

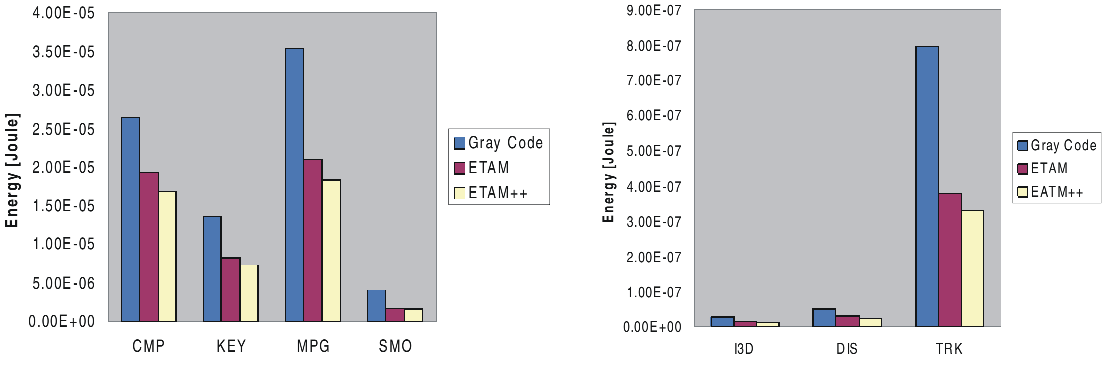

Figure 5: Energy reduction using ETAM++ compared to Gray Code Encoding and ETAM

Note that the limitation of our current work is that it targets address busses only.

6. Conclusions

Inter-wire capacitances in interconnection networks are playing an increasingly important role as we move into deep submicron and are approaching the size of intrinsic capacitances. This has motivated plenty of research in the last couple of years.

In this work we have presented ETAM++, a novel scheme that controls bus-invert encoding for address buses in designs of 01.u and beyond where inter-wire capacitances matter. Applying our scheme to a variety of SOC applications we have achieved power savings on the address buses of up to 62.5% compared to the best known standard for address bus encoding i.e. Gray Code Encoding (that cannot take inter-wire capacitances into consideration). We also exceed all results achieved by other research groups (see 'Introduction') who recently proposed schemes taking inter-wire capacitances into consideration. In addition, our scheme is easy to implement into an SOC design as it optimized in size (about 2,500 gates for the bus encoding interface).

Our future work will concentrate on encoding schemes for general data buses.

References

[1] A. Acquaviva and R. Scarsi, "A Spatially-Adaptive Bus-Interface for Low Switching Communication", Proceedings of the IEEE Int'l Symposium on Low Power Electronics and Design (ISPLED00), pp.238-240, 2000.

[2] L. Benini, A. Macii, E. Macii, M. Poncino, R. Scarsi, "Synthesis of Low-Overhead Interfaces for Power-Efficient Communication over Wide Buses", Proceedings of IEEE 36th Design Automation Conference (DAC'99), pp.128-133, 1999.

[3] L. Benini, G. De Micheli, E. Macii, D. Sciuto, C. Silvano, " Asymptotic Zero-Transition Activity Encoding for Address Bues in Low Power Microprocessor-Based Systems", Proceedings of IEEE Conference on 7th Lakes Symposium on VLSI, pp. 77-82, 1997.

[4] K.-W. Kim, K.-H. Baek, N. Shanbhag, C.L. Liu, S.-M. Kang, "Coupling--Driven Signal Encoding Scheme for Low-Power Interface Design", Proceedings of IEEE 37th Design Automation Conference (DAC'00), pp.318-321, 2000.

[5] H. Mehta, R.M. Owens, M.J. Irwin, "Some issues in gray code addressing", Proceedings of IEEE Conf.erence on 6th. Great Lakes Symposium on VLSI, pp.178-181, 1996.

[6] E. Mussoll, T. Lang, J. Cortadella, "Working-Zone Encoding for Reducing the Energy in Microprocessor Address Buses", IEEE Transactions on VLSI Systems, Vol. 6(4), pp. 568-572, December 1998.

[7] P.R. Panda, N.D. Dutt, "Low-Power Memory Mapping Through Reducing Address Bus Activity", IEEE Transactions on VLSI Systems, Vol 7(3), pp. 309-320, September 1999.

[8] S. Ramprasad, N. Shanbhag, I.N. Hajj, "A Coding Framework for Low Power Address and Data Buses", IEEE Transactions on VLSI Systems, Vol. 7(2), pp. 212-221, June 1999.

[9] Y. Shin, T.Sakurai, "Coupling-driven Bus Design for Low-Power Application-specific Systems", Proc. of IEEE 38th Design Automation Conference, pp.750-753, 2001.

[10] P.P. Sotiriadis, A. Chandrakasan, "Low Power Bus Coding Techniques Considering Inter-wire Capacitances", Proceedings .of IEEE Conference on Custom Integrated Circuits (CICC'00), pp.507--510, 2000.

[11] M.R. Stan and W.P. Burleson, "*Bus-Invert Coding for Low-Power I/O*", IEEE Transactions on VLSI Systems, Vol 3(1), pp. 49-58, March 1995.

[12] M.R. Stan, W.P. Burleson, "Low Power Encodings for Global Communication in CMOS VLSI", IEEE Transactions on VLSI Systems, Vol. 5(4), pp. 444-455, December 1997.

[13] C. Taylor, S. Dey, Y. Zhao, "Modeling and Minimization of Interconnect Energy Dissipation in Nanometer Technologies", Proc. of IEEE 38th. Design Automation Conference, pp.754-757, 2001.

[14] C.L. Su, C.Y. Tsui, "Saving Power in the Control Path of Embedded Processors", IEEE Design & Test Magazine, Vol.11(4), pp.24-31, Winter 1994.

[15] 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.

[16] J.Henkel and H.Lekatsas. A2BC: Adaptive Address Bus Coding for Low Power Deep Sub-Micron Designs. *Accepted for publication at the 38th ACM/IEEE Design Automation Conference,* June 2001.

[17] Y.Li, J.Henkel, A Framework for Estimating and Minimizing Energy Dissipation of Embedded HW/SW Systems, *IEEE Proc. of 35th. Design Automation Conference (DAC98)*, pp.188-193, 1998.