Low-Energy-Transmission of Data on Submicron Interconnects

A. MAHDOUM[1], L. HAMIMED[1], M. LOUZRI[2], M. SAADAOUI[2]
[1]Division of Microelectronics & Nanotechnologies
Centre de Développement des Technologies Avancées
BP 17 Baba Hassan 16303 Algiers Algeria
amahdoum@cdta.dz http://www.cdta.dz
[2]Departement of Computer Science
University Saad Dahlab
BP 270 Blida, Algeria

Abstract: - We present in this paper a CAD tool that aims at designing low-energy buses. The Graphical User Interface (GUI) we developed manages many techniques dealing with the addressed problem: simple coding, coding subject to fixed / dynamic probabilities and an enhanced dynamic probabilities- based technique. Moreover, this environment allows tuning the parameters of data encoding / decoding and is able to generate different gains by varying the size of the bus transferring the encoded data. Finally, this tool can be easily configured to integrate new coding techniques and use one of them when favorably compared against the other techniques.

Key-Words: - Communications Submicron interconnects Energy dissipation Data transfer

1 Introduction

In recent years, the remarkable evolution of technology has allowed the development of many embedded systems (mobile phones, laptops, PDAs, etc ...) incorporating increasingly complex functionalities while operating with high frequencies. However, the design of such systems should pay particular attention to the aspect of energy dissipation due to both the limited autonomy of the batteries and the reliability of the system. Indeed, a considerable increase in energy dissipation would cause a sharp increase in temperature which in turn would affect the proper functioning of the considered system.

Thus, the design of systems meeting performance and energy constraints must take into account a proper design of all components of the system, at different design levels. These energy reduction techniques include those based on a: synthesis of logic gates [1], voltage scheduling in order to find the desired tradeoff between energy and performance [2], design of a real-time system with energy constraints [3], [4].

Techniques based on standby circuits are also used [5]. At low levels of design (transistor, layout), the desired compromise performance / energy can be produced by using different supply voltages feeding the different gates as well as different threshold voltages for the transistors of the circuit [6].

In a VLSI system, the energy dissipation due to data transmission through a bus can be very important, especially with new technologies where the energy due to parasitic capacitances is no longer negligible (note that a parallel capacitance is 8 times larger than the intrinsic one in the CMOS 0.13 µm technology !!). Most of the works reducing the energy dissipated on a bus are based on minimizing the number of transitions ($0 \rightarrow 1$; $1 \rightarrow 0$) over the bus during data transmission. Some of such works aim at reducing the energy dissipation by exploiting redundancy [7], the Beach solution [8], Gray coding [9] or a bus invert-based technique [10].

This paper presents a simulation environment for determining the behavior of the bus when transmitting / receiving data with different coding techniques. Paradoxically, the basic idea of these techniques is to use a bus larger than the original one, causing more possible transitions than the range of data to send. The energy gain achieved by this approach lies at the policy level coding, where one considers only the less costly transitions in terms of energy consumption. We will see after how the new bus transits from one state to another while transmitting / receiving data.

This paper is organized as follows: Different types of power dissipation as well as different ways to reduce this dissipation are presented in the next section. The overall structure and outline of the environment we developed are presented in section 3. The tests and the results will be presented in

section 4, and then we conclude the paper in section 5.

2. Power Dissipation and Delay Problems

2.1 Types of power dissipation

Technology scaling allows us to integrate millions of transistors in the same chip. Unfortunately, this strong integration has some drawbacks, the most important one concerns the power dissipation. Indeed,

- the operating time of batteries is still limited
- systems on board satellites should operate during the night
- a strong dissipation yields an increase of the temperature, which affect the reliability of the system

These problems lead us to take care while designing current systems. So, in order to design low-power systems, we have to analyze each type of power dissipation:

- short circuits
- static dissipation
- switching dissipation

Power dissipation due to short circuits was crucial in older technologies (e.g. NMOS). CMOS-based technologies no longer suffer from this kind of power since NMOS and PMOS transistors are complementary, namely when an NMOS (PMOS) transistor is conducting the corresponding PMOS (NMOS) one is cut-off. So, there exists no path from the supply rail to the ground (when such path exists –due to signal transitions on the transistor grids- it rapidly disappears since the new signals on the transistor grids rapidly settle to their new values).

On the other hand, the static dissipation was negligible in older technologies. Unfortunately, this is no longer true in new technologies. Indeed, the threshold voltage of the transistors becomes too weak (which is good for a rapid commutation). So, even if some part of the circuit is not operating at a given time, $V_{gs} > V_{th}$ could be true, yielding a leakage current, thus a leakage power dissipation (V_{gs} is the voltage between the grid and the source nodes of the transistor; V_{th} is the threshold voltage of this transistor). The leakage power dissipation is defined in BACPAC (Berkeley Advanced Chip Performance) by Equation (1). W_{avg}, L, N_{trans} and V_t are the average transistor width, the transistor length (in µm), the total number of transistors in the circuit and the threshold voltage, respectively. K=10 µA/µm , α_V=0.095 V.

$$P_{leak} = 0.2813 \times V_{dd} \times K \times N_{trans} \times W_{avg} \times L \times 10^{-V_t/\alpha_V} \quad (1)$$

A switching power is dissipated as long as a circuit (system) is operating. This is due to charging and discharging the load capacitances of the logic gates. This dissipation is given by Equation (2) where V_{dd} is the supply voltage, f is the frequency, C_{Gi} is the load capacitance of the i^{th} logic gate, N_{Gi} is the number of times C_{Gi} switches.

$$P_{sw} = 0.5\, V_{dd}^2\, f \sum_{i=1}^{Nb\_Gates} C_{Gi}\, N_{Gi} \quad (2)$$

Concerning the submicron interconnects, the power dissipation will be detailed in the next sections.

2.2 Interconnect Delays

While new technologies enable us to design fast logic gates, this is not true for the interconnects. Indeed, Fig. 1 clearly shows that the gate delays are shorter than the interconnect wires (the red curve versus the pink one). However, IBM enhanced the interconnect delay thanks to the use of a low dielectric and copper instead of aluminium (the pink and the yellow curves clearly show the difference). One can further reduce the interconnect delays using additional techniques, one of them is the buffer insertion-based technique [e.g. 15]. The aim is to transform the big capacitance of the wire into N ones that are much weaker. The value of N is determined according to the tradeoff concerning area, throughput and power dissipation. Equation (3) is a delay model of the wire portion between the nodes i and j ([16]). C_{Wij} and r_{Wij} are the capacitance and the resistance of the wire portion between the nodes i and j, respectively. l_{Wij} is the length of this wire portion.

$$d_{ij} = \frac{1}{2}\left(r_{Wij} C_{Wij}\, l_{Wij}^2\right) + r_{Wij}\, l_{Wij} C_{bj} \quad (3)$$

2.3 Reduction of the power dissipation

There are many ways to reduce the power dissipation. Unfortunately, this is not achieved

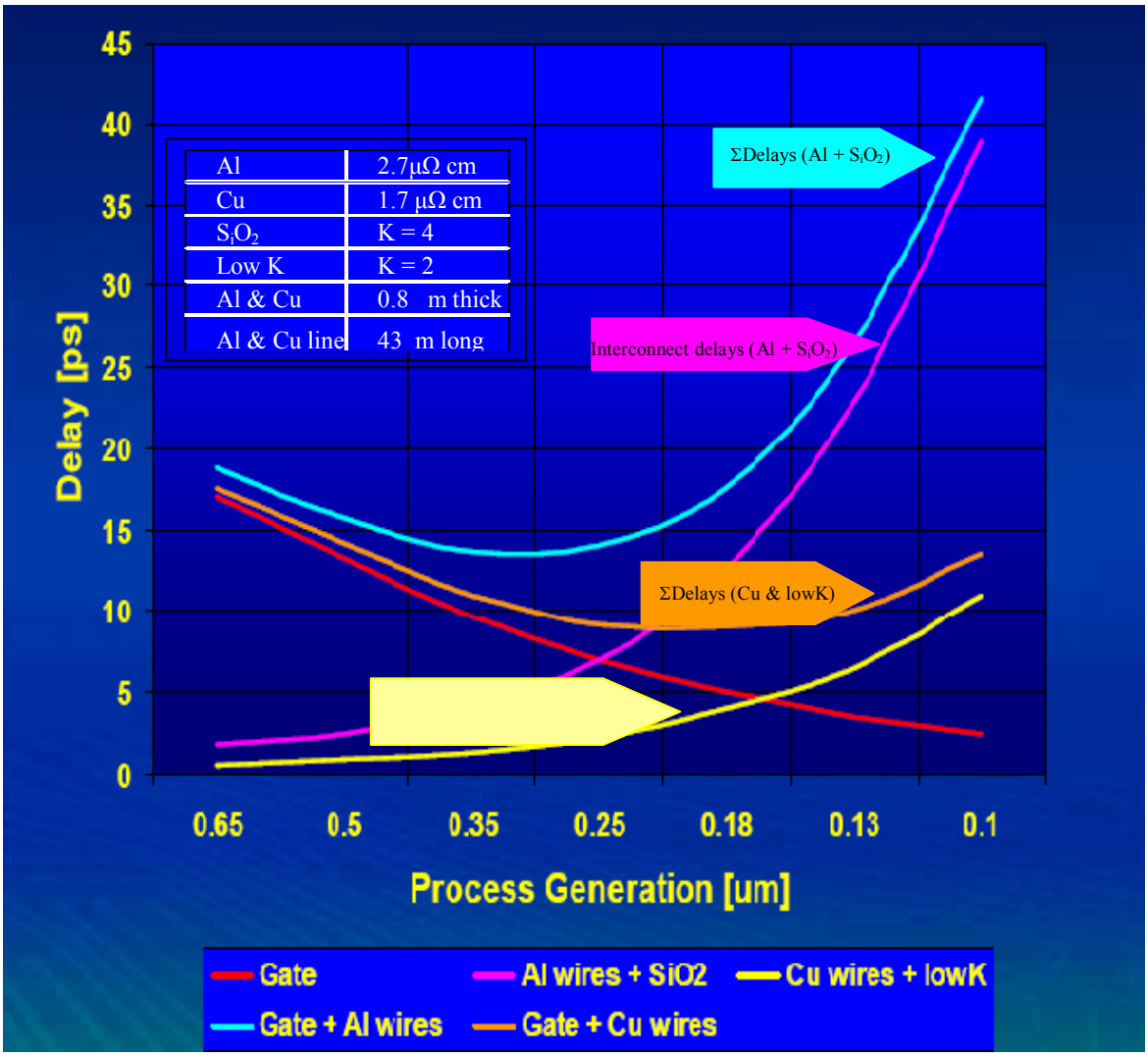

Fig. 1 Interconnect delays in submicron technologies (source: SIA NTRS Projection 2001)

without problems. Obviously, reducing V_{dd} and / or f is benefic (please see the previous equations). However, this is not good for performance. One then need to develop additional techniques so as to:

- find the desired tradeoff between power dissipation and throughput (the 2 constraints are of the same importance),
- perform all the tasks within their deadlines while minimizing the power dissipation (real time systems subject to the power constraint)
- perform the tasks with a given energy budget while minimizing as could as possible the execution time of the system

Such candidate techniques are the Dynamic Voltage and Frequency Scheduling (DVFS) or the Dynamic Voltage Scaling (DVS) (e.g. [4]). At the transistor level, one can use dual V_{dd}, dual V_{th} based techniques so as to find the desired tradeoff between power dissipation and throughput. Thus, Equations (1) and (2) become as shown by Equations (4) and (5), respectively.

$$P_{leak} = 0.2813 \times K \times W_{avg} \times L \times \sum_{i=1}^{Nb\_gates} \left[Nb_{N,i} \times 10^{-V_{tN,i}/\alpha_V} + Nb_{P,i} \times 10^{V_{tP,i}/\alpha_V} \right] V_{dd,i} \quad (4)$$

W_{avg} and L are the average transistor width and the transistor length (in μm), respectively.
K=10 μA/μm , α_V=0.095 V.
$Nb_{N,i}$ ($Nb_{P,i}$) is the number of NMOS (PMOS) transistors in the i^{th} logic gate.

$V_{tN,i}$ ($V_{tP,i}$) is the threshold voltage of the i^{th} NMOS (PMOS) transistor (we assume that the transistors of the same type –NMOS or PMOS- that are in the same logic gate have the same threshold voltage. This is due to avoid technical problems while fabricating the circuit).
$V_{dd,i}$ is the supply voltage feeding gate i.

In the same manner, Equation (2) is transformed so as it could be possible to use 2 different supply voltages for the same circuit:

$$P_{sw} = 0.5 \times f \left[V_{dd,L}^2 \sum_{i=1}^{|E_L|} C_{Gi} N_{Gi} + V_{dd,H}^2 \sum_{i=1}^{|E_H|} C_{Gi} N_{Gi} \right] \quad (5)$$

E_L (E_H) is the set of the gates that are fed by the lowest (highest) supply voltage.

Then using an appropriate CAD tool (e.g. [6] that targets the desired tradeoff between throughput and power dissipation), one achieve the assignment of the supply and the threshold voltages to the gates and the transistors, respectively.

Reducing the gate load capacitances is also benefic for power dissipation but this is not always simple as this parameter depends on many other ones (transistor widths, diffusion capacitances, the wire at the gate output, …). One then have to make use of CAD tools related to physical synthesis of the integrated circuits.

As equation (2) shows, the switching power dissipation strongly depends on the number of times each load capacitance switches. This number can be reduced by using an appropriate style of logic design (reducing also the glitches) but with taking care with the circuit throughput.

At the highest levels of design, developing appropriate algorithms and architectures may drastically reduce the power dissipation while taking into account the area and throughput constraints.

Loop transformations are one of many ways to reduce power dissipation at the algorithmic level.

Sometimes, it is possible to make use of pipeline or parallel architectures in order to reduce V_{dd}, thus the power dissipation. Indeed, for the same throughput T, small logic blocks performing in parallel require a shorter V_{dd} value ($V_{dd,new}$) than a big block performing the same functionality but needs a subsequent supply voltage V_{dd} so as to ensure the same throughput T. Because $V_{dd,new} < V_{dd}$ and because the switching power dissipation is function of V_{dd}^2, the gain could be very interesting.

3 Overall Architecture

In this section, we present the overall system architecture for encoding / decoding data in order to reduce the power dissipated on a bus. We will then show the various modules and the main techniques for data encoding / decoding.

3.1 System Architecture

The overall system architecture is shown in Fig.2. The energy dissipation of a bus strongly depends on the nature of the data passing through. To reduce such dissipation, it is very obvious to exploit those data that induce the lowest dissipation. This is done by increasing the initial size of the bus, then exploring the most interesting transitions of the bus. The original data (of size n) are then encoded with the most interesting ones (of size m; m > n) in energy point of view, then retrieved through a decoding module.

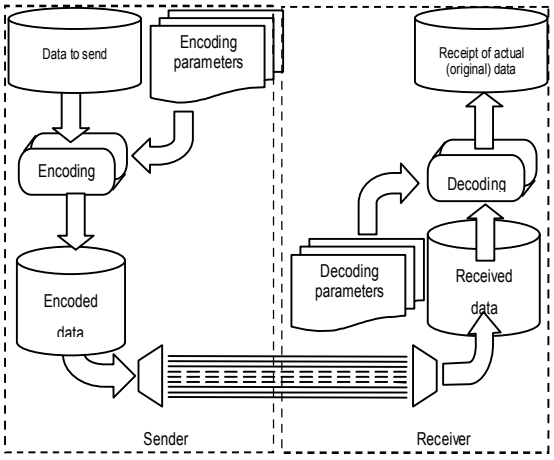

Fig. 2 A global system architecture of data coding / decoding

Table 1 Simple Encoding N = 2 and M = 3

	000		001		010		011		100		101		110		111	
	d	p	d	p	d	p	d	p	d	p	d	p	d	p	d	p
000	**00**	**0**	**10**	**4**	**-**	**7**	**-**	**5**	**11**	**4**	**-**	**8**	**-**	**5**	**01**	**3**
001	**01**	**0.5**	**00**	**0**	**-**	**10.5**	**11**	**4**	**-**	**4.5**	**-**	**4**	**-**	**8.5**	**10**	**2**
010	**10**	**0.5**	**-**	**7.5**	**00**	**0**	**10**	**1**	**-**	**7.5**	**-**	**14.5**	**11**	**1**	**-**	**2**
011	**-**	**1**	**11**	**3.5**	**01**	**3.5**	**00**	**0**	**-**	**8**	**-**	**10.5**	**-**	**4.5**	**10**	**1**
100	**11**	**0.5**	**-**	**4.5**	**-**	**10.5**	**-**	**8.5**	**00**	**0**	**11**	**4**	**-**	**4**	**10**	**2**
101	**11**	**1**	**01**	**0.5**	**-**	**14**	**-**	**7.5**	**10**	**0.5**	**00**	**0**	**-**	**7.5**	**-**	**1**
110	**10**	**1**	**-**	**8**	**11**	**3.5**	**-**	**4.5**	**-**	**3.5**	**-**	**10.5**	**00**	**0**	**01**	**1**
111	**01**	**1.5**	**-**	**4**	**-**	**7**	**10**	**3.5**	**-**	**4**	**-**	**6.5**	**11**	**3.5**	**00**	**0**

3.2 Energy due to a transition

With the old CMOS technologies, the power consumption of a bus is limited to that of the dynamic one (charging and discharging intrinsic capacitances). In current technologies, this is no longer true: dissipation due to parasitic parameters has become very important (in 0.18 m CMOS technology, the parallel capacitance between 2 neighbor wires of the bus is 8 times larger than that of the intrinsic capacitance of the wire!) [11], [12]. Thus, the energy consumption by a bus is estimated by considering the two types of energy dissipation [11], [14]:

$$E = \sum_{i=1}^{n} V_i^{new}\left(V_i^{new} - V_i^{old}\right) + \lambda \sum_{i=1}^{n-1}\left(V_{i+1}^{new} - V_i^{new}\right)\left[\begin{matrix}\left(V_{i+1}^{new} - V_i^{new}\right) - \\ \left(V_{i+1}^{old} - V_i^{old}\right)\end{matrix}\right] C_L . V_{dd}^2 \quad (1)$$

V_i^{new} is the current data on the ith wire.

V_i^{old} is the previous data on the ith wire.

λ is a parameter dependent on the used technology (it is equal to 8 in the $0.18\mu m$ CMOS technology)

C_L is the intrinsic capacity of a wire bus

3.3 Data Encoding

3.3.1 Introduction

Let N be the initial size of the bus and M the one of the encoded bus (M > N). The basic principle of data coding is to use the $2^M * 2^N$ transitions that are less costly in terms of energy, thus eliminating $2^M * (2^M - 2^N)$ additional transitions. According to Table 1, Fig.3 shows the transitions of a bus (2-3) where N = 2 and M = 3, after the elimination of the additional transitions. Indeed, Table 1 shows that for each state among the 8 ones ($2^M = 2^3$), there are 2^M= 8 possible transitions. But because the original size of the bus is N=2, we only need 2^N= 4 transitions for each of the 8 states. Thus, the total number of transitions is $8*2^N = 2^M * 2^N$. The transition of the bus from one state to another accordingly occurs with the used encoding. For example, we illustrate in Fig.4 these transitions from state 000 to another state by using the 4 (=2^N) cheapest ones from Table 1. Notice that data in this table are obtained from Equation (6)

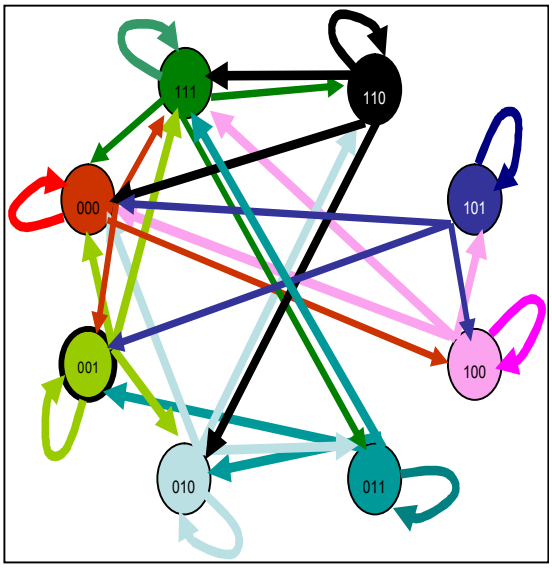

Fig. 3 The possible transitions in a bus of size 3

and that a line represents the current state while a column denotes the next state. The intersection of a row and a column, when it exists, indicates the data (d) enabling this transition and the "energy dissipation" (p) which is generated. From the current state 000, the 4 cheapest transitions lead to the next state 000 (d=00; p=0), 111 (d=01; p=3), 001 (d=10; p=4), 100 (d=11; p=4). Notice also that the actual energy dissipated when transiting from one state to another is $p * C_L * V_{dd}^2$. It is clear that the coding function G is defined as follows:

G: S x D $\rightarrow$ S'

(cs,d) $\rightarrow$ ns

S = {current states} ; |S|=2^M

S' $\subseteq$ S = {next states}; |S'|=2^N

D = {data d to send / receive; size(d)=N bits}

Thus, giving the current state *cs* and data *d* to send, we reach the next state *ns* dissipating some amount of energy according to Equation (6). Note that in this equation V_i^{old} and V_i^{new} are the previous and the current state, respectively.

Obviously, the coding function H is defined as follows:

$$H: S \times S' \rightarrow D$$

$$(cs,ns) \rightarrow d$$

Namely, the original data is retrieved thanks to the values of the previous and current states.

The coding techniques almost follow the encoding / decoding scheme we have just presented. However, they differ in how this encoding / decoding scheme is defined. We briefly review some of them presented in [13].

3.3.2 Simple Coding

In this approach, the assignment of the less costly transitions is done randomly.

Example: For N = 2 and M = 3, see Table 1. For each current state (from 000 to 111), the 4 cheapest transitions in the increasing order are associated with data 00, 01, 10, 11.

3.3.3 Coding Using Static Probabilities

The main disadvantage with the simple coding is that the cheapest transition is associated with data 00 (in case N=2) even if 00 is not frequently sent. The use of probabilities is to overcome such a drawback. Thus, the basic principle of this approach is to assign an expected probability for each data, then to schedule the transitions accordingly.

Example: N = 2 and M = 3

Table 2 shows the expected probabilities of data transmission. Table 1 and Table 2 provide, for the transition from state 000, the encoding shown in Fig. 4 by replacing 00, 01, 10 and 11 by 11, 00, 01 and 10, respectively.

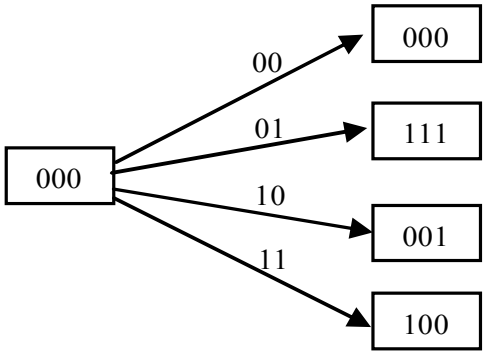

Fig. 4 Example of a bus transition from the state 000, according to the data encoding shown in Table 1

Table 2 Fixed probabilities assigned to data

Data	00	01	10	11
Probability	0.3	0.2	0.1	0.4

3.3.4 Coding Using Dynamic Probabilities

The principle of this approach is similar to the previous one except that the assignment of probabilities is not fixed but varies according to the appearance frequency of data to send. The probability assigned to the data D_i is defined as the ratio of the number of occurrences of D_i and the total number of data sent during a period. Once the probabilities are calculated, the principle of the encoding is exactly the one using the static probabilities. Thus, these probabilities are recalculated in each period.

Example:

Let us assume that the data as depicted in Fig. 5 will be sent. Let us also assume that given a period, the probabilities assigned to the different data are those shown in Table 3.

Note that for each current state, the transitions leading to the 4 least power dissipations are selected for the data encoding, considering:

- the increasing order of these 4 power dissipations,
- the probability of each data (data 11 which has the highest probability is combined with the least power dissipation among the 4 ones for the encoding, and so on …)

Assuming that 000 is the current state, Fig. 6 shows the encoding scheme and depicts the first line of Table 4. Reasoning in the same way for the remaining current states (from state 001 to 111), we obtain Table 4 which will be used for the data encoding.

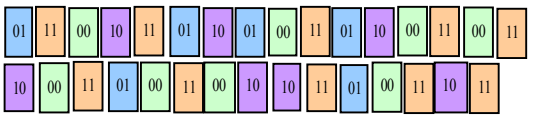

Fig. 5 Data randomly generated.

Table 3 Probability of data occurrence

Data	Frequency	Probability
00	8	0.26
01	6	0.19
10	7	0.23
11	10	0.32
Total	31	1.00

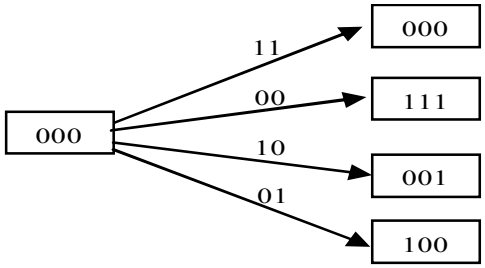

Fig. 6 Example of the bus transitions from the state 000, according to Table 1

Table 4 Encoding (N = 2 and M = 3) based on the probabilities shown in Table 2

	000		001		010		011		100		101		110		111	
	d	p	d	p	d	p	d	p	d	p	d	p	d	p	d	p
000	01	0	00	4	-	7	-	5	11	4	-	8	-	5	10	3
001	10	0.5	01	0	-	10.5	00	4	-	4.5	-	4	-	8.5	11	2
010	10	0.5	-	7.5	01	0	01	1	-	7.5	-	14.5	-	1	-	2
011	10	1	00	3.5	-	3.5	-	0	-	8	-	10.5	-	4.5	11	1
100	10	0.5	-	4.5	-	10.5	-	8.5	01	0	00	4	-	4	11	2
101	00	1	10	0.5	-	14	-	7.5	11	0.5	01	0	-	7.5	-	1
110	10	1	-	8	00	3.5	-	4.5	-	3.5	-	10.5	01	0	11	1
111	10	1.5	-	4	-	7	00	3.5	-	4	-	6.5	11	3.5	01	0

3.3.5 Enhanced Dynamic Probabilities-Based Encoding

The principle of this approach is mainly inspired by the encoding technique we have just presented. However, contrary to the previous technique, the current one calculates the probabilities **after** receiving a number of data, **not before**. The reason is that the previous data encoding determines the probabilities of the data in period P_{i+1} with observing the occurrence of each data sent during period P_i. But it is not sure that each data will exactly appear as in the previous period. Thus, in order to overcome such a drawback, a stream of data is first stored in a buffer of size N, and then examined for determining the accurate probabilities.

Assuming a data stream of size 12 as shown in Fig. 7, the enhanced data encoding processes as follows:

1. Store the data in the buffer (please see Fig.7)
2. Determine the probability of each data type (the result is shown in Table 5)
3. Use Equation (6) to determine the energy due to a transition from one state to another (Table 6 depicts the result)
4. Encode the data such that the power dissipation of the bus is the least one and send them
5. If there are still data to transmit go to step1, else stop.

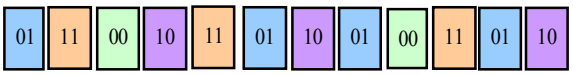

Fig. 7 Storing a data stream in a buffer

Based on the model described in [11,14] for estimating the energy dissipation on submicron interconnects, claiming that our method efficiently reduces the energy dissipation is straightforward:

- the probability of each data is not expected but rather precisely determined (because the data are first stored in a buffer before they are encoded: steps 1 and 2 of the algorithm)

- to the best of our knowledge, the best model for estimating the energy dissipation on submicron interconnects is that which is described in [11,14]. We used this model to determine the energy dissipation due to a state transition (step 3 of our algorithm)

- for each state transition, the data encoding is performed so that the 2^N state transitions (N is the initial size of data) that yield the least energy dissipation among the 2^M ones (M is the size of the encoded data; M > N) are selected (step 4)

- this process iterates for transmitting other data (step 5).

It is clear that the drawback of our algorithm is that the data are buffered and encoded before they are transmitted. So, the data are delayed, which makes our method suitable for low-energy applications rather than for real-time ones.

4 Tests and Results

Table 7 shows the results of the energy dissipated by buses whose size varies from 9 to 17 bits when transmitting 8-bit data. CWP, CFP, CDP and EC respectively stand to the encoding: - without using probabilities, - when using fixed probabilities, - when the encoding is subject to dynamic probabilities, - using the enhanced and latter described

method. Note that this table instead shows the number of charges and discharges of the capacitance of each wire of the bus, not the dissipation of energy.

Table 5 Probabilities of the transmission of the data shown in Fig. 7

Data	Occurence	Probability
00	2	0.17
01	4	0.33
10	3	0.25
11	3	0.25
NB	12	1.00

Table 6 New data encoding

	000		001		010		011		100		101		110		111	
	d	p	d	p	d	p	d	p	d	p	d	p	d	p	d	p
000	11	0	10	4	-	7	-	5	01	4	-	8	-	5	00	3
001	00	0.5	11	0	-	10.5	01	4	-	4.5	-	4	-	8.5	10	2
010	00	0.5	-	7.5	11	0	01	1	-	7.5	-	14.5	10	1	-	2
011	00	1	01	3.5	-	3.5	11	0	-	8	-	10.5	-	4.5	10	1
100	00	0.5	-	4.5	-	10.5	-	8.5	11	0	01	4	-	4	10	2
101	01	1	00	0.5	-	14	-	7.5	10	0.5	11	0	-	7.5	-	1
110	00	1	-	8	01	3.5	-	4.5	-	3.5	-	10.5	11	0	10	1
111	00	1.5	-	4	-	7	01	3.5	-	4	-	6.5	10	3.5	11	0

Table 7 Energy dissipation due to the variation of the bus size, using different Techniques

Bus size	CWP	CFP	CDP	EC
9	1527.5	1494.5	1039.5	494
10	1360.5	1362.5	1003.5	498.5
11	1292.5	1271.0	986	481
12	1288	1252.0	971.5	471.5
13	1248	1201	943.5	472.5
14	1201	1200.5	937	468
15	1222	1170.5	933.5	466
16	1117.5	1135.5	941	452.5
17	1163	1128.5	913	459.5

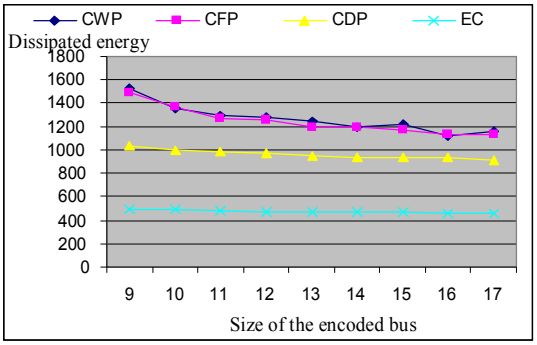

Fig. 8 Variation of the energy dissipation due to the use of different encoding techniques and different bus sizes

This latter is obtained by multiplying each of these numbers by the square of the used voltage and the value of the capacitance C_L. For example, when CWP is used, the energy dissipated by a 9-bit bus is: $1527.5 * C_L * V_{dd}^2$. The results of Table 7 are graphically depicted in Fig. 8.

Knowing that the "energy" dissipated on a 8-bit bus (i.e. without using any encoding technique) is 2173.00, Table 8 shows the obtained relative errors due to this naïve data transmission technique against each of the reviewed ones. This table clearly shows the benefits of the data encoding techniques since the relative error lies in the range 42.26% - 380.22 % !! For these data streams, the highest relative

Table 8 Relative errors (%) of the naïve technique against each of the reviewed data encoding technique

Bus size	CWP	CFP	CDP	EC
9	42.26	54.4	109.04	339.88
10	59.78	59.48	116.54	335.91
11	68.12	70.97	120.38	351.77
12	68.71	73.56	123.67	360.87
13	74.12	80.93	130.31	359.89
14	80.93	81.01	131.91	364.32
15	77.82	85.65	132.78	366.31
16	84.54	91.37	130.92	380.22
17	86.84	92.59	137.87	372.9

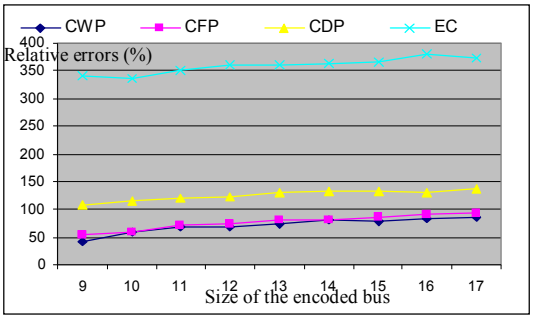

Fig. 9 Graphical representation of the errors shown in table 8

error is obtained when using the enhanced technique EC and a 16-bit bus. So, as we see, the energy is paradoxally reduced with enlarging the initial bus. However, this is no longer true with certain sizes since the energy dissipated on a 17-bit bus is higher than on a 16-bit one. One can explain this effect that with a 17-bit bus, the dynamic energy dissipation increased importantly while the one which is due to the parallel capacitances slightly reduced when comparing with a 16-bit bus. Again, the results in table 8 are graphically depicted in Fig.9 which clearly shows the benefits of the enhanced method (EC) against the others.

Figs. 10, 11, 12 and 13 clearly show that our Graphical User Interface (GUI) that is developed with *qt* under *Linux* operating system is a user-friendly interface as it allows to easily select the encoding technique, the original and the new bus sizes and to visualize the different results. Fig. 10 and Fig. 11 depict the transitions from one state to another while indicating the "energy" due to each transition.

5 Conclusion

We presented in this paper a number of data coding techniques (among them our own enhanced method) that aim at designing buses with low energy consumption. The results clearly show the benefit of the data coding techniques against the classical one (i.e. transmitting data over the bus without any encoding). These results also show that our enhanced technique favourably competes with older ones. Finally, Figs. 10-13 show that the GUI we developed is a user-friendly interface and allows to integrate a new data coding technique, comparing it against others, and then select the best one which reduces the power dissipation, which is crucial for current systems.

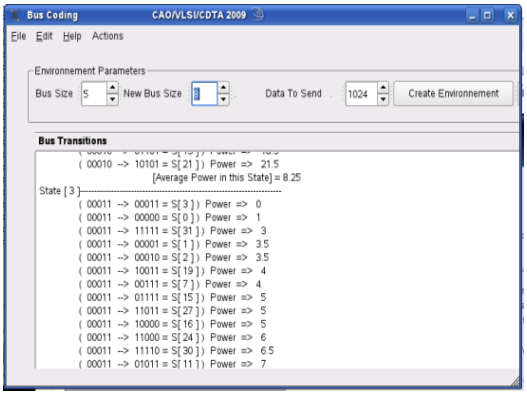

Fig. 10 General overview of our developed GUI

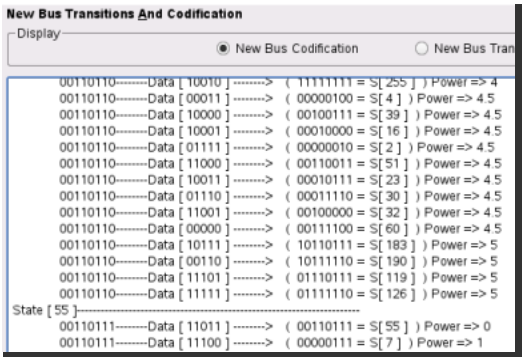

Fig. 11 Transitions on a 8-bit bus

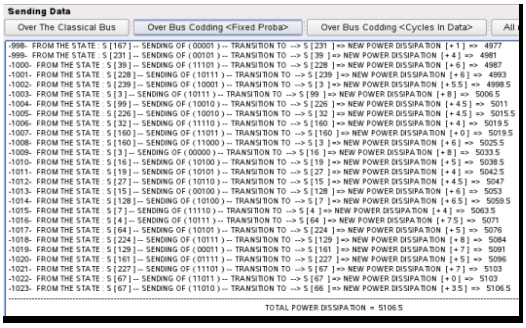

Fig. 12 Transitions on a 5-bit bus

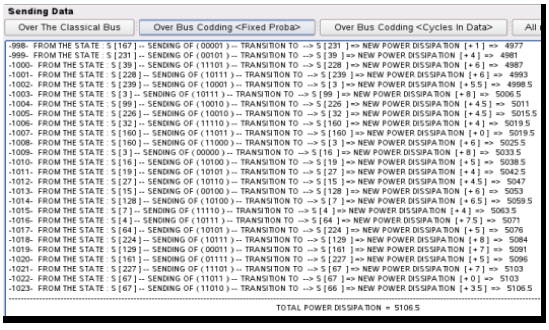

Fig. 13 A window showing power dissipations due to state transitions

References

[1] L. Benini, P. Siegel, and G. De Micheli "Saving Power by Synthesizing Gated Clocks for Sequential Circuits" IEEE Design Test Comput., 1994. vol.11, pp. 32-41

[2] S. Raje and M. Sarrafzadeh "Scheduling with Multiple Voltages" Integration, VLSI Journal, Oct. 1997, pp. 37-60

[3] L. A. Cortes, P. Eels and Z. Peng, "Quasi-static Assignment of Voltages and Optional Cycles for Maximizing Rewards in Real-Time Systems with Energy Constraints" Proceedings of the Design Automation Conference, 2005, pp. 889-894

[4] A. Mahdoum, N. Badache, H. Bessalah "An Efficient Assignment of Voltages and Optional Cycles for Maximizing Rewards in Real-Time Systems with Energy Constraints" Journal Of Low Power Electronics (JOLPE), Vol.2, No. 2, August 2006, American Scientific Publishers, pp. 189-200

[5] L. Benini, A. Bogliolo, G. De Micheli "A Survey of Design Techniques for System-Level Dynamic Power Management" Transactions on Very Large Scale Integration, June 2000, Vol. 8, Issue 3, pp. 299-316

[6] A. Mahdoum, M. L. Berrandjia "FREEZER2: Un Outil à Base d'un Algorithme Génétique pour une Aide à la Conception de Circuits Digitaux à Faible Consommation de Puissance" IEEE/FTFC'07, 21-23 May 2007, Paris, France, pp. 143-148

[7] J. Henkel and H. Lekatsas "A2BC: Adaptive Address Bus Coding for Low Power Deep Submicron Designs" Proceedings of CAR, 2001, pp. 744 -749

[8] L. Benini and al "System-Level Power Optimization of Special Purpose Applications: The Beach Solution" Proceedings of International Symposium on Low Power Electronics and Design, Aug. 1997, pp. 24-29

[9] M. Madhu et al "Dynamic Coding Technique for Low-Power Data Buses" Proceedings of IEEE Computer Society Annual Symposium on VLSI, 2003, pp. 252-253

[10] M. Stan and W. Burelson "Bus Invert Coding for Low Power I/O" IEEE Transactions on VLSI Systems 1995, pp. 49-58

[11] P. Sotiriadis and A. Chandrakasan, "Low Power Coding Techniques Considering Inter-wire capacitances" In Proc. of IEEE Custom Integrated Circuits Conferences, 2000, pp. 507-512

[12] Z. Khan et al "Novel Bus Encoding Scheme from Energy and Crosstalk Efficiency Perspective for AMBA Based Generic SoC Systems" Proceedings of VLSID, 2005, pp. 751-756

[13] A. R. Brahmbhatt, J. Zhang, Q. Wu, Q. Qiu "Low-Power Bus Encoding Using Hybrid Adaptive Algorithm" DAC'06, 24-28 July, 2006, San Francisco, California, USA.

[14] P. P. Sotiriadis A. P. Chandrakasan "Bus Energy Reduction by Transition Pattern Coding Using a Detailed Deep Submicrometer Bus Model" IEEE Transactions on Circuits and Systems, Vol.50, No. 10, October 2003

[15] A. Mahdoum, R. Benmadache, A. Chenouf, M. L. Berrandjia "A Low-Power Synthesis of Submicron Interconnects with Time and Area Constraints" International Journal of Circuits, Systems and Signal Processing, Issue 3, Vol.4, 2010, pp. 112-119

[16] R. R. Rao, D. Blaauw, D. Sylvester, C. J. Alpert, and S. Nassif, An efficient surface-based low-power buffer insertion algorithm, *ISPD*, pp. 86–93.

Dr. A. Mahdoum is a researcher at the Centre de Développement des Technologies Avancées in Algiers, Algeria. His research interests include CAD of integrated circuits and systems at different design levels.

L. Hamimed is a teacher at the High school of statistics and applied economics in Algiers, Algeria. He is conducting works relevant to many fields of computer science.

M. Louzri is teaching and researching at the Saad DAHLAB University of Blida, Algeria.

M. Saadaoui is involved in computer science and mathematical works at a National bank in Algiers, Algeria.