ON-CHIP BUS LINE ORDERING FOR MINIMIZATION ON CROSSTALK EFFECTS

Carolina M. Metzler, Luigi Vaz Ferreira, Gustavo Wilke,
Ricardo Reis

PPGC/PGMicro - Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil

ABSTRACT

With the continuous technology scaling, interconnects are more and more significant in power consumption evaluation. In on-chip buses this effect is more evident because due to the long wire length there is more interconnects, thus, it is more susceptible for crosstalk effects.

In the present paper we present an approach to minimize the crosstalk effects on power consumption in on-chip buses. Experimental results show that is possible improves the correlation between adjacent bus lines up to 23,5% and consequently the power consumption in a switching bus.

1. INTRODUCTION

Significant amount of power dissipated in a chip derives from on-chip buses. As the number of intra-chip modules increases the number of on-chip buses connecting these modules will increase as well as the need for significant data transfer and power dissipation. Due to the long wire length, in on-chip buses there are more sensitive to the crosstalk effect. Crosstalk Effects do not affect only performance, they affect power consumption as well [1]. Because on-chip buses are more sensitive to the crosstalk effects, they cannot be considered as a regular signal net in evaluating power dissipation.

Bus delay is negligible as in comparison to combinational blocks delay on clock period [4], for this reason in this paper the delay caused by crosstalk is not taken into consideration.

Crosstalk happens when at least one wire switches, in a pair of coupled wires. The aggressor wire induces a voltage variation in the victims. Therefore, when adjacent bus lines are switching in opposite directions the coupling capacitance between them leads to crosstalk, causing noise in adjacent lines impacting in timing and power of a signal on a chip.

Most of related works concern with crosstalk effects mainly in timing [3, 4 and 5]. To deal with crosstalk effects there are many techniques proposed recently that focus on energy minimization, Many of them utilize coding schemes to minimizing the number of simultaneous transitions on adjacent bus lines, as seen in [6, 7]. Other different techniques that involves physical design explore add some extra space between the lines of the on-chip bus, but this techniques have impacts on area [2].

In this paper is presented an optimization method for on-chip bus positioning. This paper purposes a method to optimize the ordering of each bus line so that crosstalk power is minimized.

In section 2, the Correlation Computation Algorithm is presented, the algorithm is utilized to obtain the correlation values between adjacent bus lines from different buses. Section 3 describes the bus placement optimization implemented to arrange bus lines according their switching correlation. Finally, section 4 correlate results between switching transition lines are evaluated statistically utilizing known benchmarks.

2. CORRELATION COMPUTATION ALGORITHM

Correlation between lines shows how many times one line switches equally then other line adjacent, for this reason is important note that a bigger correlation between lines, results in less crosstalk effects in the victim line.

To calculate correlation between the module outputs, the values that will bus is defined as the proportion between the number of times that one output pin has the same value from another output pin and all input possibilities in output results. For every input value, a comparison are made for each output set and a variable are incremented always that equal values are founded.

To save all data the structure utilized are a squared matrix called Co-relation Matrix, the output values are in rows and columns. Following the Co-relation Matrix evaluation, the variable with the number of coincidences is divided by input statistic arrangement.

Figure 1 shows a pseudocode of the Correlation Algorithm. The Co-relation Matrix (CM), “oresult” is the output data vector for a given input vector, “vectorO” is the vector of outputs and the input vector is “vectorI. The function GenerateInput is a function that provide a set of inputs that never repeats. “Fresult” is a function that calculates output due to input vector. “Matrix” is the matrix with desired correlated results. number_of_inputs and number_of_outputs are the number of elements in output and input vectors respectively.

```
Correlation(CM){
    Int I=0;
while(I<statistic_arrangement(number_of_inputs, 2)){
    GenerateInput(vectorI);
    Int J=0;
    while (J!=number_of_output){
        oresult[J]=Fresult(vectorI, vectorO[J]);
        J++;
    }
    J=0;
    while(J!=number_of_output){
     K=J+1;
      while(K!=number_of_output-1){
        if (OutputsResult[J]==OutputsResult[K])
              Matrix[J][K]++;
         K++;
      }
    J++;
    }
 I++;
}
 I=0;
 while(I!= number_of_outputs){
    J=0;
    while(J!= number_of_outputs){
      temp=statistic_arrangement(number_of_inputs, 2);
      Matrix[J][K]=Matrix[J][K]/ temp;
      J++;
    }
   I++;
 }
```

Figure 1: Pseudo code to generate the Co-relation Matrix.

According to pseudocode in figure 1, the algorithm fills the CM passed as function Correlation parameter with correlation value calculated for each input. First while compares if I is minor then the statistic arrangement of number of inputs, then, it generates the input vector and it calculates the function oresult for one output according the input vector generated before. After this calculus, another while, could be a for, generates the matrix. Finally the last two while will create the correlation matrix with the correlation values calculated as a statistic arrangement of every matrix entry. The complexity is of order $O(n^2)$, it depends on number of outputs.

3. BUS PLACEMENT OPTIMIZATION

The Co-relation Matrix generated by Correlation Function can be represented by a fully connected undirected graph (clique). The objective is to solve the outputs in the minimum cost order. This cost is the sum of output Correlation with their neighbors.

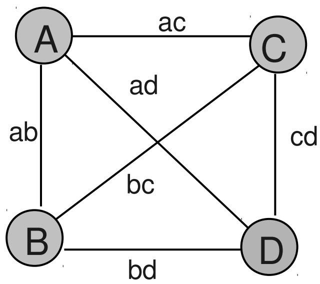


Figure 2: Correlated graph where every node is an output and the edges are the distances between nodes.

To solve the minimum path algorithm is necessary decrease Y from each element as seem in equation (1). “Correlation_Value” is acquired from the algorithm described in section 2. Y is a constant and its value is 100 in order to express the total correlation. By this way, the maximum correlation between bus lines is 100% and the minimum is 0%.

$$\text{Distance} = Y - \text{Correlation\_Value} \qquad (1)$$

A search algorithm is derived from Travelling Salesman Problem (TSP) and applied to find the minimum path in the correlation graph. The algorithm will depart from one node and will search through all the graph nodes only one time. It will repeat until all graph nodes were search from all pattern possibilities. First it is necessary choose one graph node to initiate the search, then all distance from this node is compared with all other nodes and then save all possible paths. The actual path will be chosen by the minimum cost from one node to another, from the actual node will be taken all directions to other nodes and saved in a queue. As seem in equation (2), the ActualCost” is the sum of all distance between the nodes visited and ”Distance” is calculated in equation (1).

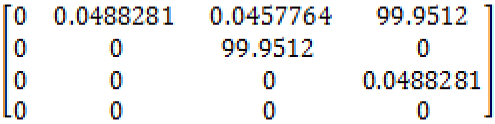
$$\text{Cost} = \text{ActualCost} - \text{Distance} \qquad (2)$$

Observe Figure 2. Let A, B, C and D be bus lines from an on-chip bus represented as graph nodes and the edges ab, ac, ad, bc, cd and bd are distances between the nodes whose values acquired with equation (1). Every node has an edge to each node in graph assuming all possibilities to find the adjacent bus line.

For each best path reached from one node to another, the Bus Positioning Optimization Algorithm save the data in a queue with possible paths to trace. This queue is organized by priority, smaller is the cost higher will be the priority.

4. CASE STUDY

The matrix bellow shows the on-chip bus lines correlation values calculated in section 2.



$$\begin{bmatrix} 0 & 0.0488281 & 0.0457764 & 99.9512 \\ 0 & 0 & 99.9512 & 0 \\ 0 & 0 & 0 & 0.0488281 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

In the generated matrix in (3), rows and columns are obtained by the output values correlations. First, a matrix is generated with the original disposal as the example in Figure 2. As the example above, originally, the given order by the *blif* file is ABCD. Running the optimization algorithm described in the last section, first it will start from a initial node, let it be node A. Node A has connections with all other nodes with different correlation values, then the algorithm will run and decide the best cost value to the next output node. According the matrix, the best cost will be to node D, where the edge ab has the minimum cost. After deciding which node will be adjacent from node A, the next node will be D and the connections to the remaining node will be evaluated. In this example the correlated node-to-node D will be C and then consequently till no more nodes left.

After the bus positioning optimization algorithm the best order due to the switching is ADCB according the correlation information seem in the matrix (3).

5. EXPERIMENTAL RESULTS

This section presents the simulation and algorithm results evaluation. In this paper the benchmarks are *blif* files acquired from Conferences. The results of this algorithm are commented in section 4.

Table 1 shows the benchmarks. These benchmarks represent intra chip modules where the input and output data are switching in intra-chip buses. These provide the data that will traffic in bus and power consumed will depend of switching pattern. First the correlation value is calculated for the original benchmark circuit, and then is possible to obtain the correlation value optimized after running the optimization algorithm.

As seem in table 1, the results allow to evaluate the correlation between bus lines. The algorithm improves the correlation between bus lines and in any case it was minimized, the improvement in correlation values was up to 26,5 % and proves the algorithm efficiency. A big correlation value means that adjacent wires will have a similar pattern switching and a small value indicates that crosstalk effects will be more expressive because most of switching will be in opposite directions. Correlation Cost values equal 50 means that a correlation improvement will not happen. As seem in table 1, a correlation improvement results in minor power consumption. Some Correlation Cost Improvement are zero, it means that the bus ordering result will be the same as the original.

Benchmarks	Number of Outputs	Correlation Cost Improvement With Optimization Algorithm (%)
misex1	7	13.50
rd84	4	2,47
sao2	4	10,05
sqrt8	4	0 (same order as original)
struct1	4	0 (same order as original)
x2	7	26,50
z4ml	4	13,05

Table1: Correlation results and power consumption simulation results for the benchmarks.

5. CONCLUSIONS

Simulation results show significant difference up to 26,5%. Here in this paper was presented a Correlation Algorithm that calculates the correlation values between bus lines and a bus line placement optimization algorithm. Choosing bus line adjacent lines by correlation values show that is possible minimize the power consumption in on-chip bus. Simulation results in [9] for a 65nm technology indicate a significant difference up to 40,95% in power consumption for a 1mm bus according the switching. This is a preliminary work and according simulation results, the minimization on power consumption expected are about 15%, it shows that crosstalk effects cannot be underestimated and ordering bus lines conform their switching can minimize power consumption.

5. REFERENCES

[1] M. Ghoneima, Y. Ismail, M. Khellah, V. De, "Variation-Tolerant and Low-Power Source-Synchronous Multi-Cycle On-Chip Interconnection Scheme", in *VLSI Design*, vol.2007, Bangalore, India 2007.

[2] L. Macchiarulo, E. Macii, M. Poncino, “Wire. Placement for Crosstalk Energy Minimization in. Address Buses,” DATE'02, pp 158–162, Paris 2002.

[3] Kei Hirose , Hiroto Yasuura, “*A bus delay reduction technique considering crosstalk*”, Proceedings of the conference on Design, automation and test in Europe, p.441-445, March 27-30, Paris, France, 2000.

[4] Sotiriadis, P. P. and Chandrakasan, A. 2001. *Reducing bus delay in submicron technology using coding.* In Proceedings of the 2001 Asia and South Pacific Design Automation Conference. ASP-DAC '01. ACM, pp 109-114. Yokohama, Japan 2001.

[5] M. Ghoneima, Yehea Ismail, "Accurate Decoupling of Capacitively Coupled Buses" International Symposium on Circuit and Systems(ISCAS), 2006.

[6] Sotiriadis P, Chandrakasan A. Bus energy reduction by transition pattern coding using a detailed deep submicrometer bus model. IEEE Trans. Circuits and Systems-I, pp.50(10): 1280–1295, 2003.

[7] Sotiriadis, P., A. P. Chandrakasan, "Bus Energy Reduction by Transition Pattern Coding Using a Detailed Deep Submicrometer Bus Model," IEEE Transactions on Circuits and Systems, pp. 1280-1295, October 2003.

[8] Sotiriadis P P, Chandrakasan A P. A bus energy model for deep submicron technology. IEEE Trans. VLSI Syst., pp.10(3): 341–350, 2002.

[9] Metzler C.M., Ferreira, L., Wilke G. ,Reis, R, "*Evaluating Power Consumption on Buses Under the Effect of Crosstalk" SIM. 25th South Symposium on Microelectronics pp 87-90, 2010.*