Interconnect Characteristics of 2.5-D System Integration Scheme

Yangdong Deng
Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
412-268-5234
yangdon@andrew.cmu.edu

Wojciech P. Maly
Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
412-268-6637
maly@ece.cmu.edu

ABSTRACT

Growing number of excessively long on-chip wires in modern monolithic ICs is a byproduct of growing chip size. To address this problem instead of placing all system's components in one layer (i.e. in 2-D space) one can use a stack of single layer monolithic ICs (called here a 2.5-D integrated IC). To assess the potential benefits of such a 2.5-D integration schema this paper compares wire length distributions, obtained for 2-D and 2.5-D implementations of benchmark circuits. In the assessment two newly developed floorplanning and placement tools were used. Significant reductions in both total wirelength and worst-case wirelength was observed for the systems implemented as 2.5-D ICs.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: Integrated circuits – *placement and routing.* J. 6 [**Computer Applications**]: Computer-aided Engineering-CAD.

General Terms

Algorithms, Performance, Design.

Keywords

2.5-D System Integration, VLSI, Floorplanning, Placement, Partition, Wirelength, Bounded Sliceline Grid.

1. INTRODUCTION

There are many positive and few negative consequences of the momentum developed recently by the microelectronics industry. One of the negative consequences is associated with the increase of chip size of modern ICs. Such an increase results in the increase of the worst-case interconnection length. And wires cannot scale as well as transistors do and delay of long on-chip wires become major components of the total signal delay budget [1]. Thus, theoretically possible advantages of new IC technologies are diminished in a substantial amount. In addition, since interconnection delay is very hard to predict before layout is generated, synthesis-based VLSI design methodologies may often have difficulty in achieving timing convergence. This may lead, in turn, to excessive number of iterations between logic and physical design. The key question is, therefore, how to address the above negative trends, preserving at the same time, momentum in the increase of the functionality of modern ICs.

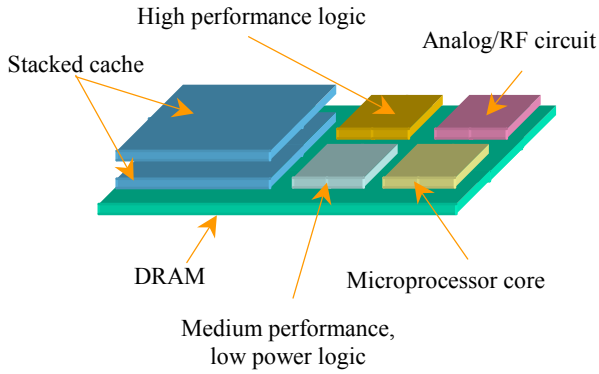

Figure 1. Example of 2.5-D system.

To find the answer to the above question one should notice first that the "excessive wire length problem" is inherent for the monolithic integration scheme. Therefore, it is natural to seek the solution by considering non-monolithic system integration schema. One of them can be, for instance, an assembly of a vertical stack of monolithic IC chips, called 2.5-D integration schema (see e.g.[2]). The purpose of this paper is to analyze whether or not 2.5-D system integration strategy can address "excessive wirelength problem" inherent for 2-D (i.e. monolithic) ICs.

This work was supported by MARCO Gigascale Silicon Research Center.

Permission to make digital or hard copies 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 and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ISPD'01, April 1-4, 2001, Sonoma, California, USA.
Copyright 2001 ACM 1-58113-347-2/01/0004…$5.00.

To appreciate both complexity and the potential advantages of the 2.5-D integration schema it is useful to consider in more detail how 2.5-D integration might be accomplished. Of course, in the first step the designed VLSI system must be partitioned into a number of sets that will contain components allocated to a specific layer of the stack. Partitioning process could be guided by taking into account system's functional boundaries or some other distinguishing attributes of its elements [2]. Then components of each set could be fabricated using specific technology, optimum for this set and manufactured at different foundry. Next system components should be stacked together, for instance, in the manner illustrated in Figure 1. The inter-die communication and power distribution might be accomplished through "vertical" connectors between stacked dies, which we call here "2.5-D vias".

Note that the most important feature of the 2.5-D strategy described above should be the **co-design** of all system components and technologies that are applied for each component's fabrication. An example of recently developed IC die stacking technology used for manufacturing of multi-processor systems is reported in [3].

To study the earlier stated question related to "excessive wire length problem" of traditional monolithic (2-D) integration strategy two independent investigations have been conducted. In each of them wire length distribution was analyzed in terms of two attributes: maximum wire length and total wire length. First one to assess the impact of the interconnect on the IC performance and the second to index the influence of wire length distribution on the consumed power. In each of the studies wire length distribution was obtained for 2-D and 2.5-D integration. In the first study wire length distribution was investigated for the case of floorplanning problem and in the second study for the placement problem. In each case new algorithm for 2.5-D design had to be developed.

The remainder of this paper describes both algorithms and the results obtained by conducting floorplanning and placement for a set of benchmark circuits [4] characterized in tables 1 and 2.

Table 1. Placement benchmark circuits statistics

design	# cells	# nets
fract	149	147
primary1	833	904
struct	1952	1920
primary2	3014	3029
industry1	3085	2593
biomed	6514	5742
industry2	12637	13419
industry3	15433	21940
avqsmall	21918	22125
avqlarge	25178	25385
golem3	100312	144949

Table 2. Floorplanning benchmark circuits statistics

design	# macros	# nets
ami33	33	123
ami49	49	409

2. FLOORPLANNING

We implemented both the 2-D and 2.5-D floorplanners using the Bounded Slice-line Grid (BSG) structure proposed by Nakatake et al. in [5]. BSG is a representation for non-slicing floorplan, which provides a large solution space including the optimal one and allows rapid exploration of this space. In our implementation, we maintain a BSG structure for each level of the 2.5-D system.

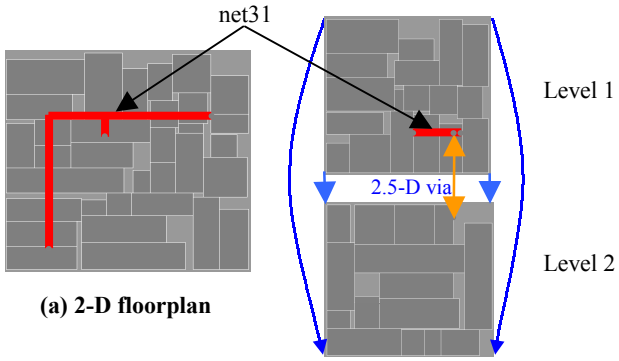

Figure 2. 2-D and 2.5-D floorplan of ami49 circuit.

The optimization is accomplished by using simulated annealing engine with the cost function given in (Eq. 1) which is the weighted sum of three terms, total wire length, floorplan area, and total number of 2.5-D vias.

$$Cost = \lambda \cdot wirelength + \gamma \cdot chip\_area + \mu \cdot num\_2.5D\_vias \quad (1)$$

In our implementation, we choose λ and γ so that the first two terms roughly equal. Thus the floorplanner puts roughly the same effort to optimize wirelength and chip area. The third term is used to indirectly control the number of 2.5-D vias. The value of μ is used to represent "cost" of 2.5-D vias. During optimization each new intermediate floorplan is generated randomly by the following three ways: displacing a cell, rotating a cell, and swapping two cells. (Note that inter-level move is allowed when 2.5-D floorplanning is performed.)

In our experiments, we used two well-known MCNC benchmark circuits, ami33 and ami49, which have 33 and 49 modules, respectively [3]. Both 2-D and 2.5-D floorplans of ami49 are shown in Figure 2. The 2.5-D floorplan consists of two monolithic floorplans, one on top of another. The wirelength reduction is clearly illustrated. For instance, in the 2-D floorplan, net31 has to span a relatively long path to connect three blocks. However, in the 2.5-D floorplan, the length of net31 is greatly reduced due to the help a 2.5-D via. Table 3 compares the results of monolithic and 2.5-D floorplans. The area values listed are the packing area for all modules not including routing area. Hence the area reduction is insignificant. However, we observe 20% and 30% total wire length reduction, respectively. Meanwhile, worst-case wirelengths in 2.5-D floorplan are 8% and 33% shorter. As the wires handled at the floorplanning level are global ones. Thus they tend to be in the critical path and have a significant impact

Table 3. 2-D and 2.5-D floorplans of ami33 and ami49.

	2-D		2.5-D		reduction	
	ami33	ami49	ami33	ami49	ami33	ami49
area of level 1	/	/	710, 892	22, 278, 928	/	/
area of level 2	/	/	571, 438	20, 921, 628	/	/
total area	1, 316, 140	44, 096, 472	1, 282, 330	43, 200, 556	3%	2%
longest wirelength	2, 923	12, 005	2, 688	8, 099	8%	33%
total wirelength	81, 351	894, 100	64, 713	625, 769	20%	30%

on the system delay, this reduction implies a significant improvement in system performance.

3. PLACEMENT

While the floorplanning study described above was good enough to provide the answer to the question discussed in this paper, relevant for the systems in which its functional blocks (or IP cores such as embedded memories) are large, it was not good to provide sufficient insight needed for the systems implemented using low granularity components. Therefore, in this section of the paper we are discussing traditional placement procedure implemented for 2-D and 2.5-D systems built with ordinary standard cells.

To accomplish our goal we have chosen to use ordinary placement algorithm and develop on top of it 2.5-D new placer. To extend the 2-D algorithm to 2.5-D, several options can be considered.

When 2.5-D via is very "expensive" (in terms of fabricating cost or consumed chip area), we should minimize total number of them. To do this, we may first carry out inter-level partitioning, which means we first min-cut the circuit netlist into a certain number of parts and assign them to different levels. Then we have two options: we can run 2-D placements simultaneously or sequentially.

On the other hand, when 2.5-D via is very inexpensive and its cost is comparable with a traditional via, the 2.5-D placement can be performed just as in the 2-D case except that now layout region consists of a set of levels. In this case partitioning is stopped always when there are a given number of cells left in each sub-circuit. At this stage we can assign each cell to either of layers. Obviously, the low cost via assumption in many cases may be too optimistic. In fact, for practical problems, the partitioning scheme should be designed to address cost of the via being in-between of the discussed above two extremes.

The idea of partition-based placement is rooted in the assumption that loosely connected cells can be placed farther away. Therefore, if the worst-case intra-level delay of this sub-region is less than the inter-level delay, or more generally the intra-level connection cost is less than inter-level connection cost, inter-level partitioning should be done and partitioned sub-circuit should be assigned to different levels. Otherwise, we do partitioning and assignment the same as in the 2-D placement process.

In the actual implementation of the placer discussed in this paper we used the first (layer assignment priority) approach. The reason for such a decision was a need to consider the worst-case scenario. If for such a scenario we could demonstrate positive characteristics of the 2.5-D integration schema then we could conclude for all other scenarios that 2.5-D will produce better results than equivalent 2-D scenario.

Another consideration is the balance constraint, i.e. the total cell area difference between two partitions. On one hand, too tight a constraint will over constrain the partitioning problem and result in poor solution quality. On the other hand, too loose a constraint will lead to unmatched areas of two levels and thus fewer nets can enjoy the benefit of vertical interconnections. In this work, we set balance constraint to 10%, which means cell area of each partition should be within 45~55% of the total cell area before partitioning.

In our implementation of the 2.5-D placement tool we have adopted Capo placer [6]. Capo is a partitioning based placer, which integrates many new techniques like multi-level partitioning, optimal end-case partition and white space allocation. In our work, the input netlist is first bi-partitioned by the multi-level partitioning engine into two sub-netlists. Next each of these sub-netlists is assigned to a different level and then we sequentially place each level. In partition based placement, terminal propagation is an effective technique to improve placement quality [7]. We adopted this idea in the 2.5-D placer extension. This means that we consider cells not only in current level but also in another level during the terminal propagation process. Thus, we still take into account global information of the whole circuit while performing intra-level placement. This also implies that a long path has opportunity to be "folded" in the 2.5-D space. When cells are first assigned to a device level, they are located at the center of the current level. This implies that terminal propagation information is not accurate when there are un-placed levels. To overcome this problem, a simple "cycle" technique is used, which means we sequentially place each level then we redo this sequence of placements utilizing the updated cell locations. In our experiments, we found a cycle of two iterations is sufficient to achieve convergence.

Table 4. Total wirelength and longest wirelength comparison

design	# of Cells	total wire length (2-D)	total wire length (2.5-D)	total wire length reduction	longest wire length (2-D)	longest wire length (2.5-D)	longest wire length reduction
fract	149	76862	66635	13.31%	4672	3781	19.07%
primary1	833	1.191E+06	9.170E+05	23.01%	1.214E+04	9.676E+03	20.32%
struct	1952	9.697E+05	8.331E+05	14.09%	1.657E+04	1.131E+04	31.72%
primary2	3014	4.780E+06	4.016E+06	15.98%	3.202E+04	2.519E+04	21.32%
industry1	3085	1.883E+06	1.571E+06	16.54%	1.801E+04	1.291E+04	28.30%
biomed	6514	5.382E+06	4.509E+06	16.23%	5.693E+04	4.031E+04	29.19%
industry2	12637	1.982E+07	1.797E+07	9.34%	7.926E+04	5.690E+04	28.21%
industry3	15433	5.268E+07	4.889E+07	7.20%	1.346E+05	1.044E+05	22.49%
avqsmall	21918	6.974E+06	5.877E+06	15.73%	5.397E+04	3.618E+04	32.95%
avqlarge	25178	7.526E+06	6.075E+06	19.29%	5.592E+04	3.822E+04	31.66%
golem3	100312	1.198E+08	9.421E+07	21.37%	1.733E+05	7.666E+04	55.77%
average	-	-	-	15.64%	-	-	29.18%

The above-described placer was used, as before, to assess the potential benefits of 2.5-D integration schema. In Table 4, we list both total wirelength and worst-case wirelength of the 2-D and 2.5-D placements. The 2-D placements are done by Capo. We assume a fixed-die model, which means the aspect ratio and channel height fixed before placement. Aspect ratio is fixed at 1 for all designs. Wirelengths in this paper are weighted pin-to-pin half-perimeter wirelength measured by the formulas of [8], which utilizes empirical data to estimate the wirelength of multi-pin net. Also we do not take into account the cost of 2.5-D via in terms of wirelength or interconnect delay when calculating wirelength due to lack of data.

Analyzing the results in Table 4 one could find that by placing circuit into two levels, on average we could reduce total wirelength by 16% and worst-case wirelength by 29% compared with 2-D placement. As for the total wirelength, more than 13% reduction is observed in 9 out of the 11 circuits. Note that in the biggest design, golem3, we can achieve a two-fold reduction in the longest wirelength. We also find that those two circuits, industry2 and industry3, which have the two lowest reductions, happen to have highest average node degrees. Generally highly connected hypergraph is more difficult to be partitioned.

It is also useful to compare wirelength distributions obtained for 2-D and 2.5-D placements, described by the histogram shown in Figure 3. Obviously, wires longer than 80,000 units do not appear in the 2.5-D placement. This suggests that the placer indeed has the capability of picking up long wires and "folding" them in a two-level layout domain. This also confirms the hypothesis that 2.5-D integration offers the opportunity to achieve top performance, which is not possible in traditional 2-D scheme. Observe also the reduction in total wirelength, which is mainly due to the smaller number of semi-global wires. Though these wire have less obvious impact on system timing, they will affect power consumption by acting as a capacitive load. Hence the shorter total wirelength also promises reduced system power consumption. Finally, note that in the 2.5-D placement there are more local wires, which have predictable wire loads. Because as we have mentioned before one of the most critical problems in current VLSI design flow is that wire load is extremely difficult to predict during synthesis, 2.5-D integration may also help with design process.

Finally, we need to mention that the wirelength reduction achieved between 2-D and 2.5-D integration schemes indicate a chance for a substantial reduction in the total chip area. This should be possible because narrower channels could be used to accommodate “less amount” of metal needed to completely wire the designed 2.5D integrated system. Such effect is not shown in the examples described in Table 3. because we decided to use in our computations unchanged channel width for both 2-D and 2.5D placements.

4. CONCLUSION

In this work, we developed physical design tools for 2.5-D integration scheme, including a 2.5-D standard cell placer and a 2.5-D floorplanner. With these tools, we were able to study wire length distribution of 2-D and 2.5-D schemes using a set of benchmark circuits. In our experiments, for floorplanning and placement cases, we achieved considerable reductions in both total wirelength and worst-case wirelength by assuming the 2.5-D scheme. These reductions can be translated into improved system timing, lowered power consumption and shortened design time. Meanwhile, because routing consumes silicon area, the reductions

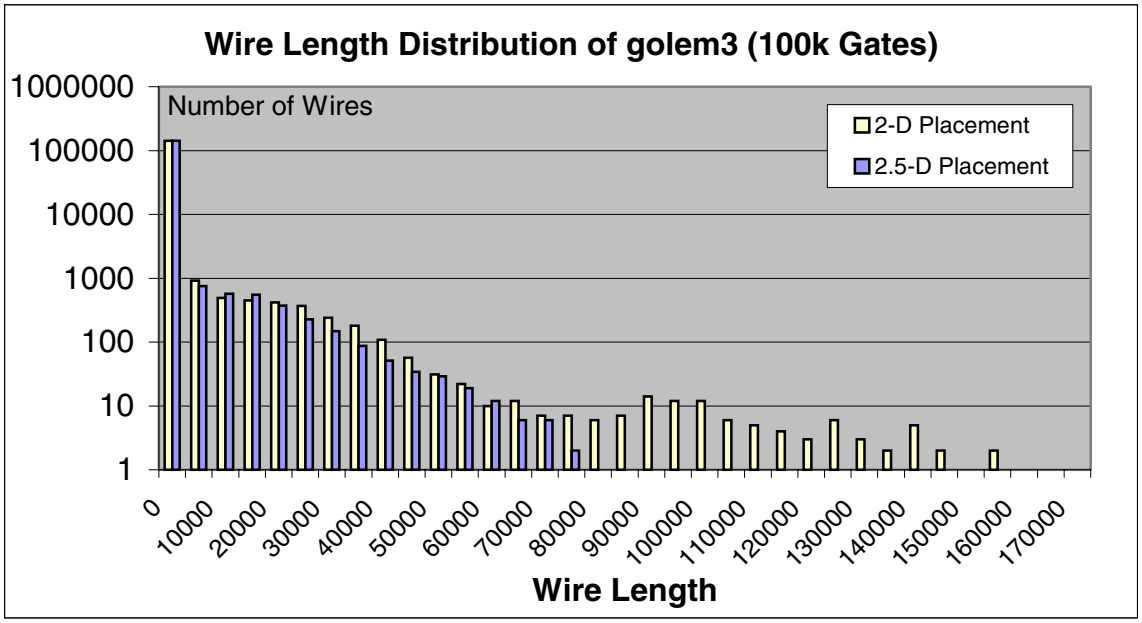

Figure 3. Wirelength distribution of design Golem3.

in wirelength also suggest opportunity to reduce chip area. Consequently, we conclude that 2.5-D integration scheme may offer substantial advantage over traditional monolithic approach.

In the future, we will further refine our placement and floorplanning methodology. For instance, we now use a top-down level assignment strategy. Hence it is interesting to investigate the effect of a bottom-up clustering approach because global wires may be identified more easily and thus we can "fold" them more efficiently. In addition, in our current implementation, during floorplanning we only consider block level assignment. Actually, another design choice is to fold a block into multiple levels. Hence we need to take into account this choice in future implementation. We are also going to assess the routability of 2.5-D placement. This may involve extending current routing tools to handle 2.5-D interconnection.

5. ACKNOWLEDGMENTS

The authors would like to express the appreciation Prof. Andrew Kahng and Prof. Igor Markov for the helpful discussions and providing Capo code as well as to MARCO for providing financial support.

6. REFERENCES

[1] M. Bohr, "Interconnect Scaling – the Real Limiter to High Performance ULSI," Proceedings of IEDM 1995, pp. 241-244.

[2] W. P. Maly, D. B. I. Feltham, A. E. Gattiker, M. D. Hobaugh, K. Backus and M. E. Thomas, "Multi-Chip Module Smart Substrate Systems ", IEEE Design & Test of Computers, Summer 1994, pp. 64-73.

[3] K. W. Lee, T. Nakamura, T. Ono, Y. Yamada, T. Mizukusa, H. Hashmoto, K. T. Park, H. Kurino and M. Koyanagi, "Three-Dimensional Shared Memory Fabricated Using Wafer Stacking Technology," Proceedings of IEDM-2000, 165-168.

[4] K. Kozminski, "Benchmarks for Layout Synthesis," Proceedings of DAC, 1991, 265-270.

[5] S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani, "Module placement on BSG-structure and IC layout applications," Proceedings of ACM/IEEE ICCAD, Nov. 1996, pp. 484-491.

[6] A. E. Caldwell, A. B. Kahng and I. L. Markov, "Can Recursive Bisection Alone Produce Routable Placements?" Proceedings of Design Automation Conf. 2000, 477-482.

[7] A. E. Dunlop and B. W. Kernighan, "A Procedure for Placement of Standard Cell VLSI Circuits," IEEE Transactions on Computer-aided Design of Integrated Circuits and System, No.4, Vol.1, 1985, 92-98.

[8] C. –L. E. Cheng, "RISA: Accurate and Efficient Placement Routability Modeling," Proceedings of ICCAD 1994, 690-694.