Simultaneous Scheduling, Binding and Layer Assignment for Synthesis of Vertically Integrated 3D Systems

Madhubanti Mukherjee and Ranga Vemuri
Department of ECECS, University of Cincinnati, Cincinnati, OH 45221-0030
Email: {mmukherj,ranga}@ececs.uc.edu

ABSTRACT

Three dimensional vertically integrated systems allow active devices to be placed on multiple device layers. In recent years, a number of research efforts have addressed physical synthesis issues for such systems. Such efforts showed a significant reduction in interconnect lengths. In order to effectively synthesize designs for 3D systems, it is necessary to take layer assignment for resources into consideration at higher levels of the design abstraction. We address the layer assignment problem as a part of a physical aware behavioral synthesis flow. We propose a 0-1 linear program formulation to perform simultaneous and optimal scheduling, binding and layer assignment for synthesizing designs for three-dimensional vertically integrated systems. The objective is to minimize inter-stratal via and the interconnect length in the critical path while taking thermal gradient between layers into account (which has been shown to be of particular concern for 3D systems). Floorplanning is performed for the synthesized design in order to estimate interconnect lengths. Results show a reduction of approximately 37% in total interconnect lengths on an average, compared to a traditional two-dimensional implementation when 2-5 layer implementations are examined.

1. INTRODUCTION

Rapid technology scaling has led to increased interconnect delays and power consumption. Although Moore's law has been accurate in it's prediction thus far, technology scaling is likely to reach a barrier once 22nm physical gate length is reached [20]. Many research efforts in recent years have concentrated on innovative strategies at the device and fabrication level to alleviate this. Among them, vertical integration of silicon in the third dimension provides a lot of promise to improve interconnect performance [10, 1].

Vertical three-dimensional integration refers to stacking of multiple active device layers by using wafer bonding [9] with vertical interconnects between them (vias). Fig. 1 [4] shows the cross-section of such a system with two active device layers. Such vertical integration has been shown to achieve significant improvements in interconnect lengths [11] and power dissipation. Joyner et al. [12] show that reduction in gate pitch and interconnect lengths in a 3D integrated design leads to an overall reduction in interconnect power. Such circuits have been shown to be extremely cost effective commercially as well [1].

This work is sponsored in part by the Ohio Board of Regents Ph.D. Enhancement Program, and in part by the DAGSI (Dayton Area Graduate Studies Institute) under the DAGSI/AFRL Research Program, Contract Number IF-UC-00-07
.

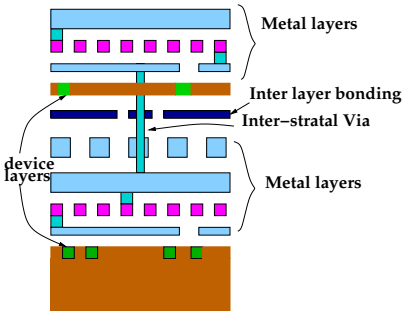

Figure 1: Vertically stacked 3D systems

In a vertically integrated 3D system, assignment of resources to layers is a part of the physical synthesis problem. Zhang et al. [21] have demonstrated the strong relationship of system partitioning with scheduling and binding in the past. As we explain in later sections, in addition to partitioning the resource set into different groups, layer assignment must also address the issue of thermal gradient between device layers as that is an extremely important issue for 3D systems. Further, inter-layer via calculations are more complex compared to the cut-set used for partitioning. All these factors make it essential to solve the scheduling, binding and layer assignment problems simultaneously in order to generate optimal designs.

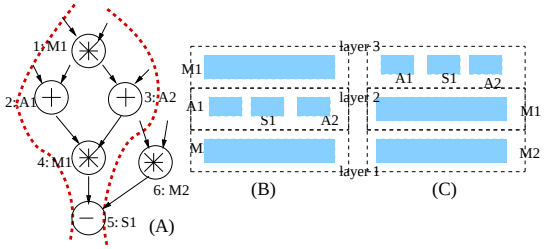

Figure 2: DFG segment and possible layer assignment

Fig. 2(A) shows a data flow graph (DFG) segment. The dashed lines show the two critical paths in the design. If operations 1 and 4 are both bound to the same multiplier, by assigning the multiplier M1 to one layer and the adders A1 and A2 to an adjacent layer, it is possible to reduce interconnect lengths significantly and also reduce the number of vias (as shown in Fig. 2(B)). On the other hand, exchanging the resource binding for operations 4 and 6 may not provide a similar advantage. Since multipliers consume more average power than an adder or a subtracter, the layer assignment shown in 2(B) has the possibility of having an non-decreasing power profile when traversed from layer 1 to 3. Although layer assignment

shown in Fig. 2(C) will have a non-increasing power profile, it leads to an increase in the number of inter layer vias. Power profile affects the thermal distribution of the system as explained in Section 2.1.

In general, the growing complexity of designs has necessitated the use of a hierarchical design strategy for the synthesis of digital circuits. Hierarchical synthesis is performed in three distinct stages as shown in Fig. 3: behavioral synthesis which or high level synthesis (HLS), logic synthesis and physical synthesis. Although a hierarchical design strategy simplifies the design problem, it also limits the design space as decisions taken at each level of the hierarchy limits the design space. This has led to physical-synthesis aware synthesis strategies wherein the decisions taken at higher levels rely on forward looking estimates of the final physical implementation. The tight coupling between HLS decisions and the final physical layout have been demonstrated by a number of researchers in the past [14, 19]. The vertically integrated 3D system is no exception and in fact even more so because of the additional complexities of thermal gradients and inter layer via. Hence, in order to design high performance designs for such systems, it is necessary to examine layer assignment at the earliest stages of the design hierarchy.

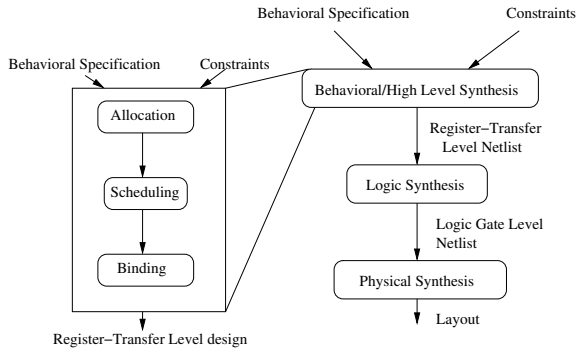

Figure 3: Hierarchical synthesis flow

In this work, we address the layer assignment problem as a part of behavioral synthesis. We propose a 0-1 linear programming formulation to perform simultaneous scheduling, binding for operations and layer assignment for resources to synthesize designs for vertically integrated 3D systems. The goal is to minimize interstratal via and the critical path length while taking thermal gradient between layers into account. Floorplanning is performed for the synthesized design in order to estimate interconnect lengths. The rest of this paper is organized as follows: Sec. 2 discusses some aspects specific to the synthesis of these systems. We outline the 0-1 LP formulation in Sec. 3. Sec. 4 describes the experimental framework and talks about the floorplanner used in this work. Experimental results are provided in Sec. 5. Finally, we conclude in Sec. 6.

2. ISSUES IN 3D SYSTEMS DESIGN

Rahman et al. [17] argues the infeasibility of integrating more than 4 or 5 strata because of the cost and complexity of integrating a large number of strata coupled with the congestion caused by interstratal interconnects. For synthesis purposes as well, it is safe to assume the use of a fixed number of strata (in the range of 3-5).

2.1 Thermal Considerations

The relationship of thermal profile of a system with its performance and reliability is well established. This also has specific implications for designs targeting vertically integrated 3D systems. For a design with L active layers, the temperature rise of the l^{th} active layer above the ambient is given by the following equation [13]:

$$\triangle T_j = \sum_{i=1}^{l} [R_i (\sum_{k=i}^{L} \frac{P_k}{A})]$$

where P_i is the power dissipation of the i^{th} layer and R_i represents the thermal resistance between the i^{th} and $(i-1)^{th}$ layers. Clearly, temperature rise in the l^{th} increases with power dissipation in that layer [13]. In order to limit this increase, we propose enforcing a non-increasing power gradient between i^{th} and $(i-1)^{th}$ layers. It should also be noted that decreasing power gradient would cause an imbalance in the active device areas among the layers by placing larger number of resources in layer closest to the lowest substrate. Hence it is also necessary to ensure that total active area of the design does not increase because of an area imbalance between layers.

2.2 Layer Assignment

The layer assignment problem for vertically integrated 3D systems might appear similar to partitioning in traditional 2D designs at first glance. However, the issues are quite different for 3D systems. First of all, partitioning essentially increases interconnect lengths for the set of nets crossing partition boundaries. On the contrary, assigning communicating resources to different layers can reduce interconnect lengths. This will be elaborated in Section 3.2 where this characteristic is used to minimize the critical path length. Secondly, the cost function to determine the number of interconnects crossing layer boundaries is dependent on the number of layers it *crosses*.

3. HLS WITH LAYER ASSIGNMENT

We formulate the simultaneous scheduling, binding and layer assignment problem as a 0-1 Linear program (0-1LP). We chose to solve the problem as a 0-1 linear program as such solutions tend to be optimal given a judicious choice of the minimization/maximization objective. Although solving such problems can be time consuming, it is acceptable to use this approach in the current context since the size of DFG's at the behavioral level are not too large and the number of stacked layers are limited as well. The variables, constraints and objective of the formulations are described in detail in this section. In addition to performing scheduling, binding and layer assignment, we also generate constraints that can be used by a placement/floorplanner that follows HLS. Fig. 4 shows the proposed flow.

3.1 Variables and Constraints

The following variables are used for describing the problem:

- L denotes the number of active device layers in the design.
- N denotes the total number of operations in a design.
- R_{max} is the total number of resources available for design implementation.
- A_{max} denotes the maximal area allowed for every layer.
- E_i denotes the ASAP schedule for node v_i.
- L_{max} denotes maximum ALAP time step among all operations. Which is also the maximal allowable latency constraint for the design.
- L_i denotes the ALAP schedule for node v_i. The ALAP schedule is determined by performing time-constrained scheduling for the DFG given a maximal allowable latency constraint (schedule length) for the design.
- $v_i \in V$ denotes the set of all operations.
- $r_k \in R$ denotes the set of all available resources.

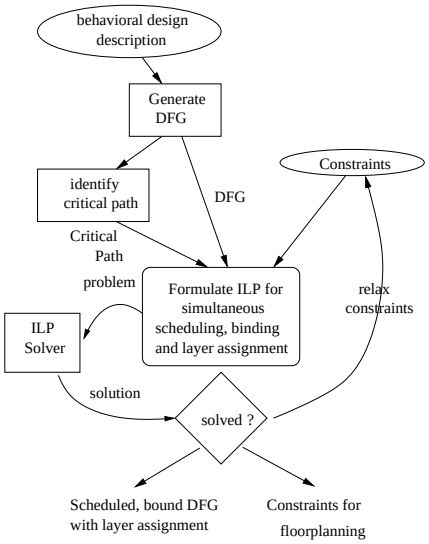

Figure 4: Simultaneous scheduling, binding and layer assignment

- v_{ijkl} is a 0-1 variable that models if an operation v_i is scheduled in control step j, bound to resource k and assigned to layer l. v_{ij}, v_{ik}, v_{il} can be derived from v_{ijkl}. These variables model if an operation is scheduled in control step j, bound to resource k, assigned to layer l, respectively. The relations between them are:

$$v_{ij} = \sum_{l=1}^{L} \sum_{k=1}^{R_{max}} v_{ijkl} \qquad v_{ik} = \sum_{l=1}^{L} \sum_{j=E_i}^{L_i} v_{ijkl}$$

$$v_{il} = \sum_{k=1}^{R_{max}} \sum_{j=E_i}^{L_i} v_{ijkl}$$

- A resource r_{kl} is a 0-1 variable that models if a resource r_k is assigned to layer l.
- p_k denotes the average power dissipation of resource r_k. A_k denotes the area of resource r_k. Both of these area known values obtained from the resource specifications.
- d_l models the average power dissipation in layer l.
- $CPath$ in the set of operation nodes in the critical path.

3.1.1 Uniqueness constraints

An operation v_i can be scheduled in only control step, bound to one resource and assigned to one layer. This is an essential set of constraints the value of which determines the binding information, control-step assignment and layer allocation for an operation node.

$$\sum_{l=1}^{L} \sum_{k=1}^{R_{max}} \sum_{j=E_i}^{L_i} v_{ijkl} = 1, \ \forall \ v_i \in V$$

3.1.2 Dependency Constraint

If there is a dependency from node v_{i_1} to v_{i_2} ($v_{i_1} \rightarrow v_{i_2}$), v_{i_2} has to be scheduled at a control step greater than the one in which v_{i_1} is scheduled. In order to enforce this constraint, for every possible schedule of v_{i_1}, the sum of $v_{i_1 j_1 kl}$ + $v_{i_2 j_2 kl}$ where $j_2 \leq j_1$ should be less than or equal to 1. If v_{i_1} is scheduled in time step j_1, scheduling v_{i_2} at $j_2 < j_1$ violates this constraint.

$$\sum_{l=1}^{L} \sum_{k=1}^{R_{max}} v_{i_1 j_1 kl} + \sum_{j_2=E_{i_2}}^{j_1} \sum_{l=1}^{L} \sum_{k=1}^{R_{max}} v_{i_2 j_2 kl} \leq 1$$

for $j_1 = E_{i_1} .. L_{i_1}$ where $v_{i_1} \rightarrow v_{i_2}$

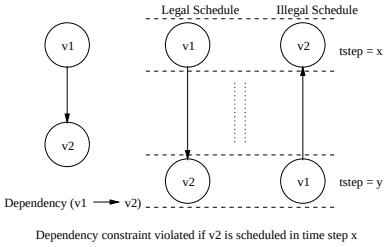

Figure 5: Dependency constraint illustration

3.1.3 Resource Usage Constraint

Every resource can be used a maximum of one time in every control step. This ensures that for all the operation that are bound to the same resource, at most one can be executing at any control-step.

$$\sum_{l=1}^{L} \sum_{i=1}^{N} v_{ijkl} \leq 1, \ \forall \ j = 1..L_{max}, \ \forall \ k = 1..R_{max}$$

3.1.4 Layer Area Constraint

The sum of the areas of resources assigned to a layer must be less than or equal to the maximum area for a layer. Zhang et al. [21] modeled partition size bounds in terms of the maximum number of nodes allowed in a partition. However, when performing layer assignment for resources, unless the active area in each layer is balanced, there is an effective increase in total silicon area (total silicon area is the product of L and the area of the layer with largest area).

$$\sum_{k=1}^{R_{max}} A_k r_{kl} \leq A_{max}, \ \forall \ l = 1..L$$

In order to ensure active area balancing A_{max} is derived from the allocated resource set by using the following relationship. $A_{max} = \sum_{k=1}^{R_{max}} A_k / L + A_{k,max}$ where $A_{k,max}$ is the size of the largest resource in the design.

3.1.5 Layer Assignment for Resources

A resource can be assigned to one and only one layer. This is a an uniqueness constraint for layer assignment of resources.

$$\sum_{l=1}^{L} r_{kl} = 1, \forall r_k \in R$$

3.1.6 Power Constraint

A decreasing power gradient is maintained from the lowest to the highest layers in order to control the thermal gradient in the design. p_k is the average power dissipation of resource r_k. The product $p_{k_1} r_{k_1 l_1}$ for a layer l_1, evaluates to p_{k_1} if the resource r_{k_1} is assigned to layer l_1. The value of d_{l_1} is equal to the sum of average power of resources in layer l_1. Non-increasing power gradient between layers l_1 and l_2 where $(l_2 > l_1)$ is enforced by the condition that for every pair of *adjacent layers*, the average power of resources in the upper layer must be less than that of the lower layer.

$$\sum_{k=1}^{R_{max}} p_k r_{kl} - d_l = 0, \ \forall \ l = 1..L$$

$$d_{l_1} - d_{l_2} \geq 0, \ \forall l_1, l_2 \ \text{ where } (l_2 > l_1)$$

3.1.7 Correctness Constraint

In every control step, v_{ijkl} for the operations bound to a resource r_k must match r_{kl} for that control step and that layer. This is necessary to ensure that the r_{kl} and v_{ijkl} correspond to each other. For example, if operation v_1 is bound to resource r_5 and is assigned to layer 3, $v_{1j_153} = 1$ and so, $r_{53} = 1$.

$$\sum_{i=1}^{N} \sum_{j=E_i}^{L_i} v_{ijkl} - r_{kl} = 0 \quad \forall \ r \in R, \ \forall \ l \in L, \ \forall \ l = 1..L$$

3.2 Objective

The objective is to reduce communication costs in terms of the number of vias, while reducing interconnect length in the critical path. Since the critical path in a DFG dictates the latency of the design, minimizing the interconnects along the critical path would reduce design latency. This is in contrast to increasing the clock speed of the design, but both ultimately serve similar purposes.

To minimize vias, we need to assign communicating operations to the *same layer* whenever possible. Minimization of the interconnect length in the critical path is motivated by the fact that two communicating operations have smallest separation when one is placed above the other in the z-dimension. We propose *physical chaining* of the resources in the critical path along the layers of the design. Physical chaining prevents them from being clustered in two adjacent layers (which would lead to increase in intra-layer interconnect length) while reducing interconnect lengths by providing vertical proximity. This is illustrated in Figure 6. The objective is formulated as a maximization goal with α and β being *normalized* constants used for weighing the two factors in order to perform a trade-off.

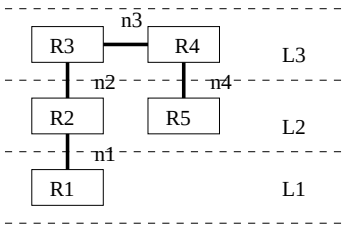

Figure 6: Illustration of the critical path objective

$$Objective = \alpha.Vias + \beta.CrtPath$$

$Vias = \sum_{l=1}^{L} v_{i_1 l} * v_{i_2 l} \ \forall v_{i_1}, v_{i_2}$ where $v_{i_1} \rightarrow v_{i_2}$. The product $v_{i_1 l} * v_{i_2 l}$ evaluates to unity only if the two operations are assigned to the same layer. If there are DFG_{edges} edges in a data flow graph, the maximum value that $\sum_{l=1}^{L} v_{i_1 l} * v_{i_2 l}$ can have, is DFG_{edges}. In order to normalize the contribution of vias, the value of α is set to $1/DFG_{edges}$. Hence, the sum of products term $\sum_{l=1}^{L} v_{i_1 l} * v_{i_2 l}$ can take the maximum value of 1 when all pairs of communicating resources are assigned to the same layer.

Besides the layer area constraints, what prevents all operations (hence all resources), to be assigned to the same layer is the *CrtPath* objective. The $CrtPath$ objective tries to maximize assigning communicating operations in the critical path in adjacent layers in a physical chain. This is based on the hypothesis that inter-layer wires can be shorter compared to intra-layer wires (by performing constraint-driven placement/floorplanning wherein communicating resources across layers are placed close together in the xy-direction).

$$CrtPath = \sum_{l=1}^{L} (\sum_{i_1=1, i_2=i+1}^{P} v_{i_1 l_1} * v_{i_2 l_2} + v_{i_2 l_2} * v_{i_3 l_3} + .. + v_{i_{p-1} l_{p-1}} * v_{i_p l_p}) \ \forall v_{i_1}, v_{i_2}$$

where $v_{i_1} \rightarrow v_{i_2}$, $(v_{i_1}, v_{i_2}) \in CPath$, $P = CPath_{length}$ and $l_1, l_2, ...l_p$ form a chain of length P in either of the form m, m+1, m+2,... ,L-1, L, L-1, L-2,.....,3, 2, 1 or of the form m, m-1, m-2,...2, 1, 2, .., L-1, L. Fig. 6 shows the preferred physical chaining of resources in the critical path. The term has a maximum value when every pair of resources in the critical path are assigned to layers adjacent to each other in a physical chain. Since the maximum value of this term can be equal to the critical path length, the normalization factor β used is $1/P$.

While the first part of the objective tries to minimize the vias, the second part tries to place resources in different layers causing more vias to be introduced in the system. Maximization of the sum of the two objectives allow us to perform a trade-off between interconnect lengths and vias. Theoretically the maximum value that the objective function can take is 2 but for a realistic system the objective function is maximized by a trade-off between the two goals.

Ilog Cplex [3], a Linear Programming Package, has been used to solve the LP formulation. We have used Task Graphs for Free (TGFF) [18] to generate a number of pseudo random data flow graphs for use as benchmarks for validating the proposed approach. We were able to handle DFGs up to the size of 44 nodes in a reasonable amount of time (the longest time was 56 seconds). TGFF has been used earlier [7] for scheduling and binding research. It allowed us to examine various types of DFGs of different sizes and examine the applicability of our approach in a very generalized context.

3.3 Linearization

There are a number of different linearizartion techniques that can be used to convert non-linear equations to a linear form [8, 2]. We use Fortet's linearization method [2]. For every term $t_1 * t_2$ a new variable t_3 is introduced. The relation between these variables is:

$$t_1 + t_2 - t_3 \leq 1 \quad , \ -t_1 - t_2 + 2 * t_3 \leq 0$$

The first equation forces t_3 to be 1 when both t_1 and t_2 are 1. The second equation forces t_3 to be 0 when either t_1 or t_2 is 0.

4. CONSTRAINT GENERATION FOR FLOORPLANNING

While trying to minimize vias, we prefer assigning communicating resources to layers close to each other; ideally in the same layer. Consider the situation illustrated in Fig 7. R_1 and R_2 communicate with each other. ($R_1 \rightarrow layer_1$, $R_2 \rightarrow layer_1$) is a preferred assignment compared to ($R_1 \rightarrow layer_1$, $R_2 \rightarrow layer_2$). Further, the second mapping is preferred over ($R_1 \rightarrow layer_1$, $R_2 \rightarrow layer_3$). Let the floorplan generated in each case result in interconnects Net_1, Net_2 and Net_3 respectively. But it is possible for the interconnect lengths to have the following relation: $Net_1 > Net_2 > Net_3$ depending on the placement of resources within every layer. Thus, an optimal solution should be able to consider floorplanning as a part of the scheduling, binding and layer assignment problem. However, inclusion of placement/floorplanning in the LP formulation increases the complexity of the problem manifold. Hence, we chose to perform simultaneous scheduling, binding and layer assignment

and generate the optimization objective and constraints that can be used by a floorplanner.

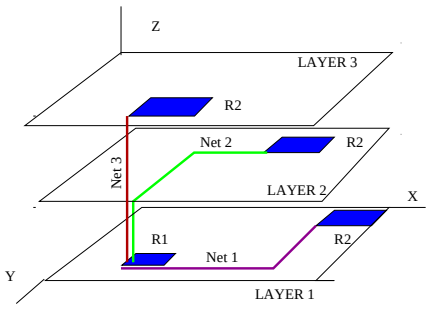

Figure 7: Multi-layer floorplanning and interconnect lengths

Floorplanning is performed on all the synthesis results in order to obtain wirelength estimates. The floorplanner used is based on that proposed by Kim et al. [15] which proposes a linear programming formulation for generating optimal floorplans. We perform floorplanning for the scheduled and bound DFG for different values of L (number of design layers) and compare the interconnect length estimates with that obtained using traditional 2-D floorplanning on the same design using the above floorplanner. To accommodate simultaneous floorplanning for all the layers for a vertically integrated system, the following constraints/objective are generated:

- No-overlap constraints are specified **only between resources assigned to the same layer**. This allows us to generate the floorplans for all the layers simultaneously.
- The minimization objective for the floorplanner includes reduction of the two-dimensional bounding box enclosing every interconnect that span multiple layers. This corresponds to minimizing $N_x + N_y$ in Fig. 8 as that would lead R_1 and R_2 to be placed directly above one another. This results is placing connected resources that are assigned to different layers with minimal x-y separation between them and ensures that the significant contribution to the total interconnect is due to the via length (which we have no control over).

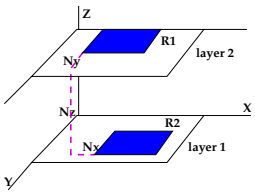

Figure 8: x-y minimization for inter layer communication

Wirelength estimation models for vertically integrated 3D systems is not quite well developed. Deng et al. [6], used half-perimeter wirelengths while studying the interconnect characteristics of vertically integrated systems but did not take via lengths into account. Their contention was that the contribution of via-lengths to complete interconnect length is minimal. Das et al. [5] on the other hand used used the 3-D bounding box metric. We have used a similar model for estimating interconnect lengths. Since empirical data about via lengths is not readily available, we have varied the contribution of a via to an interconnect to be 10%, 25% and 40% of the maximum height or width of the resource set used in the design. we will refer to this as $R_{(x,y)_{Max}}$ in the following section. Further details are not provided due to limitation of space.

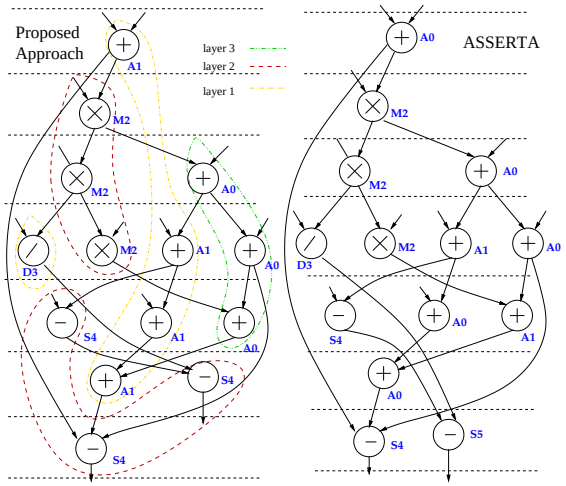

Figure 9: Comparison with ASSERTA

5. EXPERIMENTAL RESULTS

We first illustrate the difference between the scheduling and binding obtained by the proposed methodology with a traditional approach. We compare the scheduled and bound DFG generated by our approach with that generated by a heuristic HLS approach ASSERTA [16]. ASSERTA uses Force Directed List Scheduling followed by recursive-improvement based resource binding. Fig.9 shows the scheduling and binding performed by our proposed approach and by ASSERTA. We also show the assignment of resources to layers when maximum number of layers is about 3. In the proposed approach, resources A_1 and D_3 are in layer 1, resources M_2 and S_4 are in layer 2 and A_0 is in layer 3. If the same layer assignment was performed for the scheduled and bound DFG generated by Asserta with S_5 being assigned to layer 3 (for area balancing), the number of vias increase from 8 in the first case to 14 in the second.

Fig. 10 shows the variation of total interconnect lengths when inter-stratal via lengths were set to 40% of $R_{(x,y)_{Max}}$. Each group of data show the total interconnect lengths for a benchmark as the number of layers increase from 1 to 5. The benchmark sizes increase along the x-axis from 14 to 44 operation nodes. The proposed approach failed to generate a result for benchmarks with 39 and 44 nodes when trying to map to 5 device layers. This is possibly because the solver was unable to meet power gradient constraints given the area balancing factor for those designs. Fig.11 shows the reduction in interconnect lengths as a percentage reduction compared to a 2D floorplanner for different number of layers. The inter-stratal via lengths were set to 40% of $R_{(x,y)_{Max}}$ in this case as well. Although the percentage reduction for the smaller benchmarks is highest for 4 layers, as the size of benchmarks increase, 2 and 3 number of layers show the most reduction. Additionally, the reduction is more significant for larger benchmarks.

Fig. 13 shows the average reduction in interconnect lengths for the entire benchmark suite for different number of layers. The three line in the graph shows the average reduction when inter-stratal via lengths are set to 10%, 25% and 40% of $R_{(x,y)_{Max}}$. The important observation here is the existence of a point of diminishing returns. In each case, the average % reduction in interconnect length reaches a peak and then starts falling when additional layers are stacked. Note that there is still a improvement compared to a 2D implementation, but the % improvement falls. Fig. 12 shows the number of vias introduced in the design when the design is mapped to increasing number of layers. For 24 and 34, the number of vias jump significantly when stacked layers are increased from 3 to 4. This also

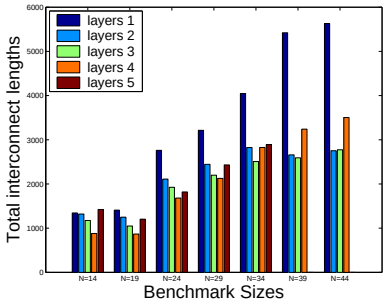

Figure 10: Net length variation

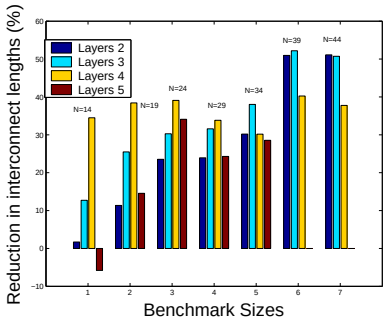

Figure 11: % Reduction in net length

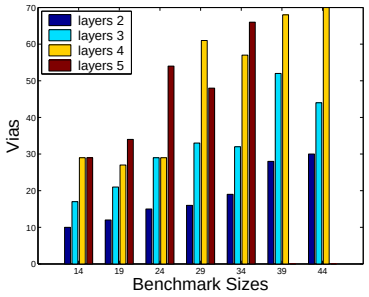

Figure 12: Change in # Vias

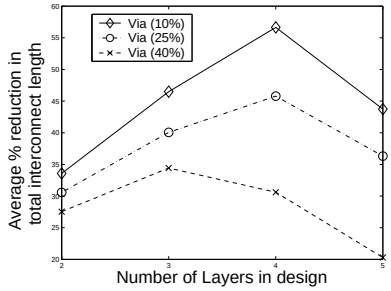

Figure 13: Average total interconnect length reduction with increasing number of stacked layers

validates the existence of such a point of diminishing returns.

6. CONCLUSIONS

In this work we addressed the layer assignment problem for three-dimensional vertically integrated systems as a part of a physical aware behavioral synthesis flow. We outlined a 0-1 linear programming formulation to perform simultaneous scheduling, binding and layer assignment. In order to estimate total interconnect lengths, floorplanning was performed on the resulting scheduled, bound data flow graphs. The experiments were restricted to DFGs with up to 44 nodes since 0-1 LP has a time complexity exponential in the number of variables. On an average, reductions of 37% was obtained for total interconnect lengths (for different values of the inter-stratal via lengths) compared to a traditional two-dimensional implementation when 2-5 layer implementations are examined. We also observed that the gain was greater for larger benchmark sizes. A trend was established for the gain and we observed was the existence of a point of diminishing returns.

7. REFERENCES

[1] http://www.matrixsemi.com/index.shtml.

[2] P. Hansen et al. Constrained NonLinear 0-1 Programming. *ORSA Journal of Computing*, 5(2), 1993.

[3] I. Cplex". http://www.ilog.com/products/cplex/.

[4] S. Das, A. Chandrakasan, and R. Reif. Three-dimensional integrated circuits: performance, design methodology, and CAD tools. In *Proceedings. IEEE Computer Society Annual Symposium on VLSI*, pages 13–18, Feb 2003.

[5] S. Das, A. Chandrakasan, and R. Reif. Calibration of Rent's Rule Models for Three-Dimensional Integrated Circuits. *IEEE Transactions on VLSI Systems*, 12(4):359–366, April 2004.

[6] Y. Deng and W. P. Maly. Interconnect Characteristics of 2.5-D SystemIintegration Scheme. In *ISPD*, pages 171–175, 2001.

[7] G. Constantinides et al. Heuristic Datapath Allocation for Multiple Wordlength Systems. In *Proc. of DATE*, pages 791–797, 2001.

[8] C. H. Gebotys. Optimal synthesis of multichip architectures. In *Proc. of ICCAD*, pages 238–241, 1992.

[9] Guarini, K. W. et al. Electrical integrity of state-of-the-art 0.13 /spl mu/m SOI CMOS devices and circuits transferred for three-dimensional (3D) integrated circuit (IC) fabrication. In *Electron Devices Meeting, 2002. IEDM '02. Digest. International*, pages 943–945, 2002.

[10] J. A. Davis et al. Interconnect limits on gigascale integration (GSI) in the 21st century. *Proc. of the IEEE*, 89(3):305–324, March 2001.

[11] J. W. Joyner et al. Impact of three-dimensional architectures on interconnects in gigascale integration. *IEEE Transactions on VLSI Systems*, 9(6):922–928, Dec 2001.

[12] J. Joyner and J. Meindl. Opportunities for reduced power dissipation using three-dimensional integration. In *Proc. of the IEEE International Interconnect Technology Conference*, pages 148–150, June 2002.

[13] K. Banerjee et al. 3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. *Proc. of the IEEE*, 89:602–633, May 2001.

[14] K. Bazargan et al. Integrating scheduling and physical design into a coherent compilation cycle for reconfigurable computing architectures. In *Proc. of DAC*, pages 635–640, 2001.

[15] J.-G. Kim and Y.-D. Kim. A linear programming-based algorithm for floorplanning in vlsi design. *IEEE Transactions on CAD*, 22(5):584–592, May 2003.

[16] N. Narasimhan and R. Vemuri. Specification of Control Flow properties for Verification of synthesized VHDL designs. In *Formal Methods in CAD*, pages 327–345, 1996.

[17] A. Rahman and R. Reif. Thermal Analysis of Three-Dimensional Integrated Circuits. In *Intl. Interconnect Technology Conference (IITC)*, pages 157–159, 2001.

[18] Robert P. Dick et al. TGFF: task graphs for free. In *Proc. of the 6th international workshop on Hardware/software codesign*, pages 97–101. IEEE Computer Society, 1998.

[19] J. Um, J. Kim, and T. Kim. Layout-driven resource sharing in high-level synthesis. In *Proc. of ICCAD*, pages 614–618, 2002.

[20] V. Zhirnov et al. Limits to binary logic switch scaling-a gedanken model. *Proc. of the IEEE*, 91(11):1934–1939, Nov 2003.

[21] Z. Yang and R. Gupta. A case analysis of system partitioning and its relationship to high-level synthesis tasks. In *11th Intl. Conf. on VLSI Design*, pages 442–448, Jan 1998.