3DHLS: Incorporating High-Level Synthesis in Physical Planning of Three-Dimensional (3D) ICs

Yibo Chen*
Synopsys Inc.
Mountain View, CA 94043
yibo.chen@synopsys.com

Guangyu Sun
Peking University
Beijing, China 100084
gsun@pku.edu.cn

Qiaosha Zou, Yuan Xie
Pennsylvania State University
University Park, PA 16802
{quz106, yuanxie}@cse.psu.edu

***Abstract*—Three-dimensional (3D) circuit integration is a promising technology to alleviate performance and power related issues raised by interconnects in nanometer CMOS. Physical planning of three-dimensional integrated circuits is substantially different from that of traditional planar integrated circuits, due to the presence of multiple layers of dies. To realize the full potential offered by three-dimensional integration, it is necessary to take physical information into consideration at higher-levels of the design abstraction for 3D ICs. This paper proposes an incremental system-level synthesis framework that tightly integrates behavioral synthesis of modules into the layer assignment and floorplanning stage of 3D IC design. Behavioral synthesis is implemented as a sub-routine to be called to adjust delay/power/variability/area of circuit modules during the physical planning process. Experimental results show that with the proposed *synthesis-during-planning* methodology, the overall timing yield is improved by 8%, and the chip peak temperature reduced by 6.6 °C, compared to the conventional *planning-after-synthesis* approach.**

I. Introduction

To further improve integration density and to tackle the interconnect challenge as technology continues scaling, researchers have been pushing forward three-dimensional (3D) IC stacking [1], [2]. In a 3D IC, multiple device layers are stacked together with direct vertical interconnects through substrates. 3D ICs offer a number of advantages over traditional two-dimensional (2D) design, such as (1) higher packing density and smaller footprint; (2) shorter global interconnect due to the short length of through-silicon vias (TSVs) and the flexibility of vertical routing, leading to higher performance and lower power consumption of interconnects; (3) support of heterogenous integration: each single die can have different technologies.

As we pack more and more transistors into a single chip, the pace of productivity gains has not kept up to address the increases in design complexity. Consequently, we have seen a recent trend of moving design abstraction to a higher level, with an emphasis on **Electronic System Level (ESL)** design methodologies. A very important component of ESL is raising the level of abstraction of hardware design. High-level synthesis (HLS) provides this component by providing automation to generate optimized hardware from a high-level description of the function or algorithm to be implemented in hardware. HLS generates a cycle-accurate specification at the register-transfer level (RTL) that is then used in existing ASIC or FPGA design methodologies.

Conventional thinking on HLS for 3D ICs focused on such a *planning-after-synthesis* flow that HLS was done first to generate the block-level (function unit) implementation of circuits, and 3D integration is used to put together blocks such as adders and multipliers. For any unsatisfactory in the integration, HLS is redone and then physical design follows. The problem here is that these blocks might be very small in size and this leads to a fine granularity in 3D integration. The physical planning (layer assignment and floorplanning) at this granularity might be impractical in real designs for the following reasons: (1) Very fine-grain integration of 3D ICs splits modules to different layers and creates more inter-layer connections, which is not desirable as the delay and area overheads on through-silicon vias (TSVs) is not negligible. (2)In typical designs there would be thousands of such function-level blocks. Optimal physical planning for such a input scale could be challenging and time-consuming, which complicates the early-stage design exploration and increase time to market of a design.

A new thinking is to bring 3D integration to a higher level, to address the integration of modules in a system, instead of function units inside a module. In this way, the combination of HLS and 3D IC design might be a different story. HLS is implemented as a sub-routine to be called to adjust delay/power/area of circuit modules during the physical planning process. In this *synthesis-during-planning* flow, HLS is first performed to estimate the delay, power and area of architectural modules in the system, such as ALU, FPU, *etc*. Then 3D partition and floorplanning are done to integrate these modules, and timing/power/variability/thermal analysis are performed on the planned results. If the constrains are not met, the modules on critical paths or hot spots are picked out and sent back to HLS. In this iteration, HLS is to re-explore the design space of such modules on certain directions and to generate new implementations with different (delay, power, variability, area) properties. Modules with new internal architectures provide new opportunities for the 3D integration and therefore it's likely to improve the design towards the constraints. With multiple iterations it will generate designs that are ready for low-level implementation. This flow can avoid the problem and concerns on very fine-granularity 3D integration, and can be done quickly to get a reasonable 3D implementation in the early stage of a design process.

*This work was done when the first author was pursuing Ph.D. degree at The Pennsylvania State University.

This work is supported in part by SRC grants, and NSF 0643902, 0702617, 0903432, and 1017277.

978-3-9810801-8-6/DATE12/©2012 EDAA

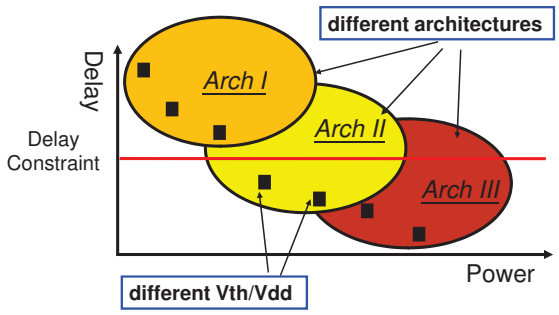

Fig. 1. The design space of a module in high-level synthesis

II. Related Work

Most of previous work on layout-aware high-level synthesis only handled 2D circuits [3], [4]. These approaches typically use a loosely coupled independent floorplanner for physical estimation. Gu *et. al* [5] proposed a incremental exploration framework of the combined physical and behavioral design space, which enables maintaining physical-level properties across consecutive physical estimations during behavioral synthesis. Tightly integrating the high-level and layout-level phases of synthesis is necessary to ensure convergence of the synthesis flow.

Only a few previous work have been reported on high-level synthesis aimed at 3D layouts. Mukherjee *et. al* [6] has addressed the layer assignment problem during high-level synthesis for 3D ICs. However, their approach separates the high-level synthesis from the layer assignment step, and the 0-1 integer linear programming formulation in their approach is typically unable to explore large design space due to the computation complexity. Krishnan *et. al* [7] proposed a 3D-layout aware binding algorithm for high-level synthesis of 3D ICs. While these work addressed the synthesis of 3D IC in various aspects, the major drawback is the granularity of the objects for physical planning. Tackling the physical planning problem at functional blocks level would be trivial as we stated in the previous section.

Our work in this paper is substantially different from the existing ones, in such a way that physical planning at the granularity of modules operates in an outer loop, while HLS is employed as a refining tool in the inner loop to optimize the delay/power/area of modules, so that the modules can be fit better in the physical planning of 3D ICs.

III. The Problem Definition

This section introduces preliminaries on high-level synthesis and 3D physical planning, and then presents the problem formulation of this paper.

A. High-Level Synthesis

High-level synthesis (HLS) is the process of translating a behavioral description into a register level structure description. Scheduling and resource binding are key steps during the synthesis process. The scheduler is in charge of sequencing the operations of a control/data flow graph (CDFG) in correct order and it tries to schedule as many operations as possible in the same control step to extract more parallelism. The binding process binds operations to hardware units

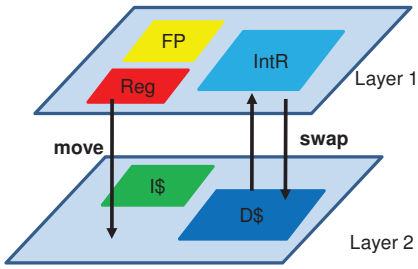

Fig. 2. 3D physical planning including layer assignment and floorplanning

in the resource library to complete the mapping from abstracted descriptions of circuits into practical designs.

The resource library consists of hardware units with different delay and area properties, which make it possible to perform design exploration to get more optimized result during the synthesis process. Figure 1 shows the design space exploration for a specific module used in high-level synthesis. `ARCH I-III` denotes different architectural implementations of the module with the same functionality, while the black squares inside each architecture shows the design points with different module-level and circuit-level optimizations (multi-Vth/Vdd, gate sizing, *etc*.), which lead to different delay and power values of the same architecture. Therefore, given the delay constraint shown in Figure 1, there are two architectures available and each architecture has two viable design options. The synthesis tool can choose between these options for a best fit of all design considerations.

B. 3D Physical Planning

Physical planning is a key step in 3D IC design. It usually involves layer assignment and floorplanning of modules on each layer. Layer assignment, which is unique in 3D IC design, assigns blocks and modules to different layers in order to balance the area split, to mitigate the thermal crisis, as well as to reduce the interconnect wirelength. Figure 2 shows the physical planning process for a 3D microprocessor. During the planning process, modules can be moved around within a layer as well as between layers, to achieve the best performance in terms of delay/power/area/*etc*. A coarse-grain physical planning methodology can generate balanced assignment and placement of modules at the early stage of the design process, and provide confidence and guidance for the succeeding design steps.

C. Problem Formulation

During 3D physical planning, several operations can be performed to change the location, layer assignment and aspect ratio of any module. However, the delay/power/area of such modules are fixed, leading to a limited design space. Meanwhile, high-level synthesis can generate architectures and design points of a give specification with different delay/power/area values. Therefore, if high-level synthesis can be incorporated into the physical planning process, the design space will be significantly enlarged and the quality of the design decision will be greatly improved. With high-level synthesis as a tuning knob of each module in the design, a framework with two levels of optimization loops can be established, in which the physical planning serves as the outer loop and evaluates all potential movements, and high-level synthesis acts as a fine tuning tool to

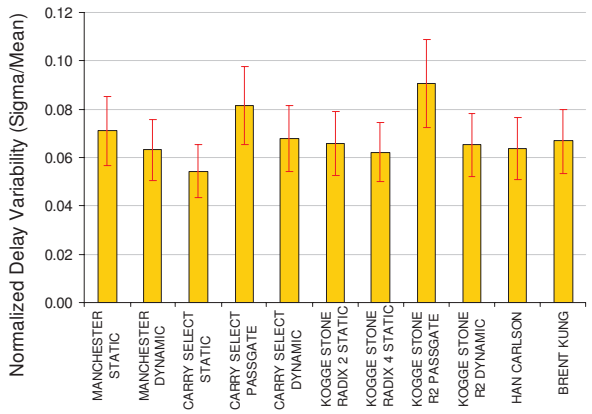

Fig. 3. The delay variation for 16-bit adders in IBM Cu-08 technology (Courtesy of IBM).

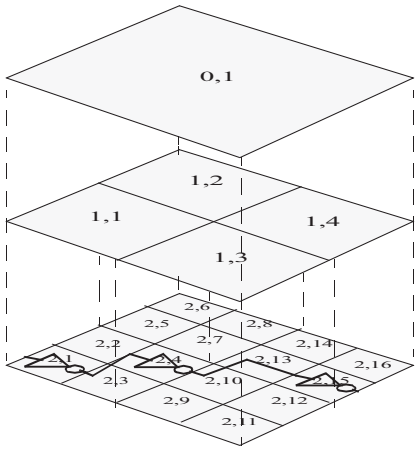

Fig. 4. The process variability model with spatial correlation [12].

facilitate the movements. Within such a framework, several design considerations including performance, process variability and thermal efficiency of 3D ICs can be effectively addressed:

1) Mitigating the Impact of Process Variations: CMOS process variability is a major challenge in deep-submicron SoC designs. The variations in transistor parameters are complicating both timing and power consumption prediction. Previous work has shown that different architectural implementations of the same function modules, might have different immunity to process variability [8]. As shown in Fig. 3, the normalized delay variability varies with different adder architectures. Consequently, architectural components such as ALU will exhibit similar behavior when they are built up with these circuit modules. Therefore, besides delay and power, the variability becomes a new metric to explore and to optimize during higher levels of the design hierarchy.

Researchers have then proposed optimization techniques in behavioral or high-level synthesis to reduce the variability of synthesized results, with the price of increased power or silicon area [9]–[11]. These techniques can mitigate the impact of process variations at the early stage of a design flow. However, most of these optimizations are physical-unaware – the work on high-level is unable to address the impact of interconnects and spatial correlations of process variability. Combining high-level synthesis and physical planning will lead to an effective way to address the process variability of 3D ICs. In our proposed framework, after the initial physical planning is done, the interconnect delays are extracted and the process variations of all components are re-evaluated with the updated spatial correlations. With such information, the near-critical components can be identified, and HLS is then called to optimize such modules to reduce the variability. The optimization will result in alternatives with different power/area, and these may violate the delay/power/thermal constraints so a new iteration of physical planning is required.

To bring the process-variation awareness to the high-level synthesis flow, we first introduce a new metric called *Parametric Yield*. The parametric yield is defined as the probability of the synthesized hardware meeting a specified constraint $Yield = P(Y \leq Y_{max})$, where Y can be delay or power. We assume that each architectural blocks are separately by registers. Given the clock cycle time t_{CLK}, the timing yield of the entire system, $Yield_t$ is defined as:

$$Yield_t = P(t_1 \leq t_{CLK}, t_2 \leq t_{CLK}, \ldots, t_n \leq t_{CLK}) \quad (1)$$

where $P()$ is the probability function, $t_1, t_2, \ldots, t_n$ are the arrival time distributions of each architectural block $B_1, B_2, \ldots, B_n$, respectively.

In each iteration, we use the model presented in [12] to model the spatial correlations of the process variability. As shown in Fig. 4, an independent random variable $L_{l,r}$ is associated with each region (l, r) to represent a component of the total process variation. The total intra-die variation of a given block will be the sum of $L_{l,r}$ variables corresponding to the (l, r) regions that intersect with the block. Such spatial correlations are then fed to a gate-level statistical timing analysis tool to recalculate the yield of each architectural block.

2) Improving Thermal Efficiency: Heat dissipation is one of the key challenges in the design of 3D ICs, due to the stacking of multiple active layers and the improved logic density. Therefore, the thermal efficiency problem has to be addressed from the very beginning of the 3D IC design. In our proposed physical planning framework, a temperature estimator could be integrated into the flow to evaluate the temperature profile of the design after any optimistic move. At each iteration, the framework will identify the modules with hot spots. Such modules will be moved around the chip for a better head conduction or be replaced with their alternatives. At this point, the design space of such modules could be explored, and candidates with lower power consumption, thus lower heat dissipation could be chosen as the replacement. After the movement or replacement of these critical modules, a new iteration will be initiated to optimize the other modules in order to meet the design constraints.

For quick temperature estimation, we model 3D ICs as resistor-capacitor structures in a thermal RC model similar to [13]. For a microarchitectural unit, heat conduction to the thermal package and to neighboring units are the dominant mechanisms that determine the temperature. The RC model consists of a vertical and a lateral convection model. Figure 5 shows the block level RC modeling, where each modules is modeled as a block with its power dissipation and thermal resistance. The vertical thermal resistance (R_V) captures heat flow from one layer to the next, moving from the source die

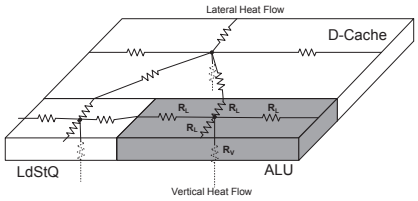

Fig. 5. Block mode thermal RC modeling for 3D ICs.

through the package. The lateral thermal resistance (R_L) captures heat diffusion between adjacent blocks within a layer, and from the edge of one layer into the periphery of the next area. The temperature of each block can then be estimated as:

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1n} \\ R_{21} & R_{22} & \dots & R_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ R_{n1} & \dots & \dots & R_{nn} \end{bmatrix} \times \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix}. \quad (2)$$

where $T_{1\dots n}$ and $P_{1\dots n}$ denotes the temperature and power consumption values for blocks $1 \dots n$, respectively. $\mathbf{R} = R_{1\dots n,1\dots n}$ is the transfer matrix built from R_V and R_L of each block.

With the proposed fast temperature estimator, the physical planning flow is able to address the thermal issue by iteratively identifying the hot spots and optimizing the corresponding modules through design space exploration, creating a balanced thermal profile for the whole IC stack.

IV. Iterative Physical Planning Framework

The synthesis-during-planning flow proposed in this paper is formulated as an iterative optimization problem, in which the physical planning process works at the outer loop, identifying the critical modules and optimizing them by calling the high-level synthesis process.

The framework is based on a simulated annealing engine, as shown in Fig. 6. The optimization starts with a system design specification, which defines the architectural modules used in the system as well as the data flows between them. Firstly, such modules are synthesized using the built-in high-level synthesis sub-routine. From the synthesis results, the delay/power/variability/area of the modules are extracted and then fed to the physical planning process. The physical planner first assigns modules to different layers of the 3D IC stack, and then performs floorplanning on each layer. After the floorplanning, the delay and power consumption of the whole stack is updated with the interconnect information, while the parametric yield and the thermal profile are updated with the spatial information. A cost function incorporating all the constraints is then evaluated. New annealing moves will be generated for those modules on the critical "spots", which could either be modules on the critical timing paths, or modules of hottest spots in thermal profile. Those critical modules are sent to the high-level synthesis sub-routine, in which the delay/power/variability of the module are optimized through design space exploration. With the updated properties of the critical modules, a new iteration of physical planning is required. The iterative process continues until the cost is significantly low or the iteration counts reach a preset bound.

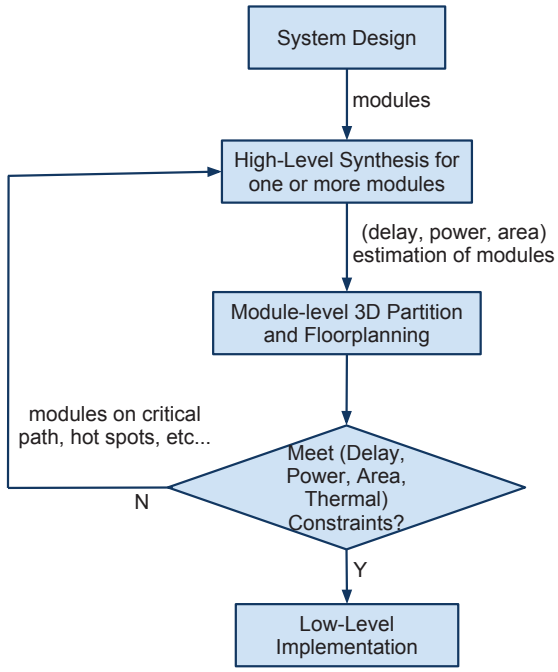

Fig. 6. The iterative planning and synthesis framework

A. On-Demand High-Level Synthesis as a Tuning Knob

In the highl-level synthesis sub-routine, the delay, power, and variability of the input module can be optimized via a set of techniques. On unit level, different function units with the same functionality can be chosen to facilitate the module; on circuit level, techniques such as multi-Vdd/Vth and device sizing can be used to tune up the modules. These optimizations can be done on the spot, however, they can be time consuming depends on the size of the modules.

In order to improve the design efficiency, for each module used in the design, several candidates can be generated *a priori* using the high-level synthesis process, and a look-up table can be built with multiple choices of delay/power/variability for the given module. In such a way the call to the high-level synthesis sub-routine could be very fast and the running time of the whole optimization framework can be reduced.

B. Incremental 3-D Floorplanning Operation

Layer assignment and floorplanning are the key steps in the proposed framework. The floorplanner used in the physical planning process is based on the work in [14], which simultaneously assign blocks to each layer and perform floorplanning operations. There are six perturbation operations used in the algorithm:

1) Node swap, which swaps two modules.
2) Rotation, which rotates a module.
3) Move, which moves a module around.
4) Interlayer swap, which swaps two modules at different layers.
5) Interlayer move, which moves a module to a different layer.
6) Replace, which replace a module with its alternatives.

The first five operations are from the original work in [14]. Operation (6) is added to facilitate the design space exploration during the physical planning process.

C. Cost Function

The optimization is guided by several cost factors including the chip foot-print, the total interconnect length, the overall timing yield of the stack, and the thermal efficiency. The cost function can be written as

$$cost = \alpha * area + \beta * wl + \gamma * yield + \theta * temp \quad (3)$$

1) Total Interconnect Wirelength: With the continuous technology scaling, interconnect has emerged as the dominant source of circuit delay and power consumption. Three-dimensional (3D) ICs have recently recognized as a promising means to mitigate the interconnect-related problems [1], [2]. During the physical planing process interconnect wirelength should be minimized in order to maintain the performance benefits brought by 3D integration.

2) Chip Foot-print: One problem of 3D floorplanning is the final packed area of each layer must match to avoid penalties of chip area. For example, assuming two layers L1 and L2, if the final width of packed modules of L1 is larger than the final width of packed modules of L2 and the height of L1 is smaller that of L2, a significant portion of chip area is wasted due to the need for the layer dimensions to match for manufacturing. Thus, care must also be taken in both stages of our algorithm so that the dimension of each layer will be $compatible$.

3) Overall Timing Yield: $yield$ in Equation (3) denotes the overall timing yield of the design. As we mentioned in Section III-C, the overall timing yield depends on not only how the modules are implemented, but also where the modules are placed. For each synthesized module, the gate-level netlist is provided by the high-level synthesis tool. A gate-level statistical timing analysis tool, PrimeTime VX, is then called to analyze the arriving time distributions of each module, taking into account the spatial correlation of the process variations. The overall timing yield is then computed from the yield values of each module according to Equation (1).

4) Thermal Efficiency: $temp$ in Equation (3) denotes the peak temperature of the 3D chip. Peak temperature is an effective metric of the thermal efficiency. Since block-level thermal evaluation is used during the optimization iterations, the peak temperature is actually the average temperature of the hottest block in the stack.

The weighing factors α, β, γ, and θ are carefully chosen so that the cost factors can co-exist with their respect contributions to the total cost functions.

V. Experiment Analysis and Case Studies

In this section, we first present the experiment results of the design space exploration for delay/power/variability in high-level synthesis, and then demonstrate the effectiveness of our proposed *synthesis-during-planning* approach on several benchmarks.

A. Results for Design Space Exploration in High-Level Synthesis

We first show the variation-aware delay and power characterization of function units. The characterization is based on NCSU FreePDK 45nm technology library [15]. Variations on two device parameters, *channel length* and *oxide thickness*, are set with relative deviations (σ/mu) to be 5%, respectively. The voltage corners for the characterization are set as: $V_{th}^L = 0.37V$, $V_{th}^H = 0.56V$, $V_{dd}^L = 0.9V$, $V_{dd}^H = 1.1V$. The characterization results for five

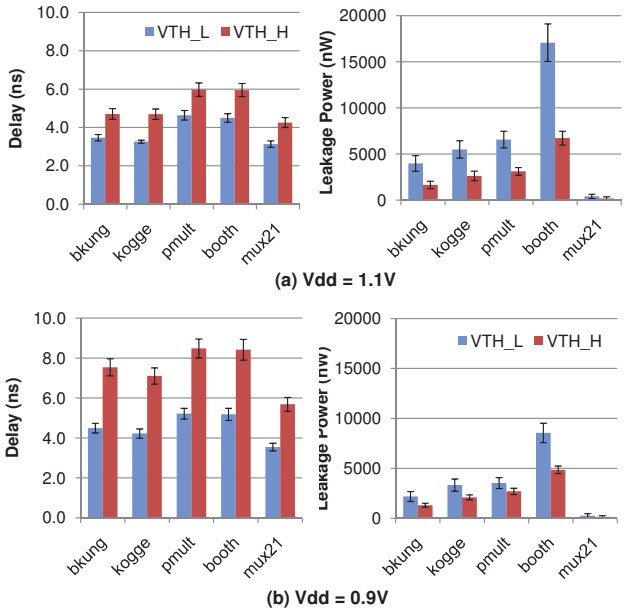

Fig. 7. Delay and leakage power characterization of function units with multi-V_{th}/V_{dd} and variation awareness.

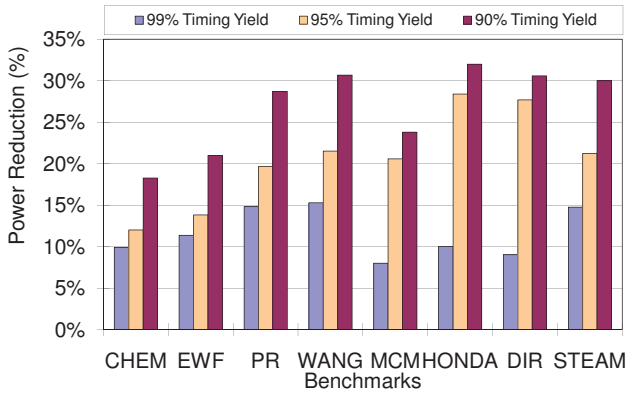

Fig. 8. Design space exploration on timing and power yield of the synthesized design.

function units, including two 16-bit adders *bkung* and *kogge*, two 8-bit×8-bit multipliers *pmult* and *booth*, and one 16-bit multiplexer *mux21*, are depicted in Fig. 7. In the figures, the color bars show the nominal case values while the error-bars show the deviations. It is clearly shown that with lower V_{dd} and/or higher V_{th}, power consumptions are reduced at the cost of delay penalty.

With the variation-aware multi-V_{th}/V_{dd} resource library characterized, we performed design space exploration for power reduction on a set of industrial high-level synthesis benchmarks. The dynamic power consumption of function units is estimated by Synopsys *Design Compiler* with multi-V_{th}/V_{dd} technology libraries generated by *Liberty NCX*. In this work with FreePDK 45nm technology, the dynamic power is about 2 times of the mean leakage power.

Fig. 8 shows the power reduction of the synthesized design compared with conventional single-voltage design. The average power reductions against conventional worst-case based design, under timing yield constraints 99%, 95% and 90% are 11.7%, 21.0% and 27.2%,

TABLE I
PHYSICAL PLANNING RESULTS OF 3D ARCHITECTURE

Circuit	Planning-after-Synthesis					Synthesis-during-Planning				
	wire (um)	area (mm^2)	peakT	timing yield	run time(s)	wire (um)	area (mm^2)	peakT	timing yield	run time(s)
$Alpha$	210749	15.49	126.01	85.2%	363	210833	15.44	117.48	94.3%	2718
$ami33$	27911	0.613	164.60	92.7%	113	27435	0.625	154.65	98.8%	1165
$ami49$	547491	18.55	130.22	90.1%	386	52090	18.72	124.19	95.7%	4590
hp	124819	4.45	125.67	88.7%	16	119863	4.51	120.39	96.5%	871
$xerox$	297440	9.76	127.21	91.3%	17	295768	9.60	119.64	97.2%	1615
$Average$	1	1	1	1	1	0.99	1.00	0.95	1.08	12.24

respectively. It is clearly shown that, the power reduction largely depends on how much timing yield loss is affordable for the design. This testifies the necessity of design space exploration for a well balanced timing yield and power trade-off.

B. Results of the Incorporated 3D Physical Planning Flow

For the experiments on 3D physical planning, we use an Alpha-like detailed microprocessor with the IVM verilog model [16], as well as several MCNC benchmarks. The processor model is synthesized with FreePDK 45nm technology library. The power density of each module is set between $1 - 5W/mm^2$. The designs are split into two layers of a 3D IC, and the delay and area overhead on through-silicon-vias (TSVs) are not accounted for the sake of brevity. The physical planning flow is implemented in C++ and experiments are conducted on a Linux workstation with a 3.2GHz dual-core CPU.

We compare the results with the conventional *planning-after-synthesis* flow as shown in Table I. The left half of the table displays the total wirelength, the chip footprint, the peak temperature of the chip, and the overall timing yield for the conventional approach, while the right half shows the results for our proposed approach, respectively. The cost factors are set in order that the total wirelength and the chip footprint are kept unchanged, while the rest two factors are to be optimized. Results over the benchmarks show that the proposed *synthesis-during-planning* approach can reduce the chip peak temperature by 6.6 °C on average, and improve the overall timing yield by 8%, without causing overheads on wirelength or chip area. The benefits come from the fact that low-power or low-variability alternatives of modules which fit the needs best, are placed on the critical spots, while the slightly expensive modules in terms of power or area, are introduced at non-critical locations. This demonstrates the effectiveness and necessity that extra high-level design space exploration is used during the physical planning of 3D IC design. Due to the iterative optimization, the run time of the proposed approach increases by about 10 times, but is still in an acceptable range for a design flow starting from system-level specifications and generating placed RTL netlists.

VI. CONCLUSION

This paper proposes an incremental system-level synthesis framework that tightly integrates behavioral synthesis of modules into the layer assignment and floorplanning stage of 3D IC design. Behavioral synthesis is implemented as a sub-routine to be called to adjust delay/power/variability/area of circuit modules during the physical planning process. Experiment results show that the proposed synthesis-during-planning approach outperforms the conventional planning-after-synthesis approach in reducing the chip power consumption, chip area and the impact of process variability.

REFERENCES

[1] W. R. Davis, J. Wilson, and et al. Demystifying 3D ICs: the pros and cons of going vertical. *IEEE Design & Test of Computers*, 22(6):498–510, 2005.

[2] Y. Xie, G. H. Loh, and et al. Design space exploration for 3D architectures. *J. Emerg. Technol. Comput. Syst.*, 2(2):65–103, 2006.

[3] Kim Daehong, Jung Jinyong, and *et al*. Behavior-to-placed RTL synthesis with performance-driven placement. In *ICCAD*, 2001.

[4] A. Stammermann, D. Helms, M. Schulte, A. Schulz, and W. Nebel. Binding, allocation and floorplanning in low power high-level synthesis. In *ICCAD*, 2003.

[5] Gu Zhenyu, Wang Jia, and *et al*. Incremental exploration of the combined physical and behavioral design space. In *DAC*, 2005.

[6] Mukherjee Madhubanti and Vemuri Ranga. Simultaneous scheduling, binding and layer assignment for synthesis of vertically integrated 3d systems. In *ICCD*, 2004.

[7] Krishnan Vyas and Katkoori Srinivas. A 3D-layout aware binding algorithm for high-level synthesis of three-dimensional integrated circuits. In *ISQED*, 2007.

[8] Yibo Chen, Yuan Xie, Yu Wang, and Andres Takach. Parametric yield driven resource binding in behavioral synthesis with multi-Vth/Vdd library. In *ASP-DAC*, 2010.

[9] J. Jung and T. Kim. Timing variation-aware high level synthesis. In *ICCAD*, 2007.

[10] F. Wang and *et al.* Variability-driven module selection with joint design time optimization and post-silicon tuning. In *ASPDAC*, 2008.

[11] G. Lucas, S. Cromar, and D. Chen. FastYield: Variation-aware, layout-driven simultaneous binding and module selection for performance yield optimization. In *ASPDAC*, 2009.

[12] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *ICCAD*, 2003.

[13] W. Huang, S. Ghosh, and et al. Hotspot: a compact thermal modeling methodology for early-stage VLSI design. *TVLSI*, 14(5):501–513, 2006.

[14] W.-L. Hung, G. M. Link, and *et al*. Interconnect and thermal-aware floorplanning for 3D microprocessors. In *ISQED*, 2006.

[15] NCSU. 45nm FreePDK. http://www.eda.ncsu.edu/wiki/FreePDK.

[16] IVM. http://www.crhc.illinois.edu/ACS/tools/ivm/about.html.