MONOLITHIC 3D INTEGRATION OF ASYNCHRONOUS SYSTEMS

A Thesis
Presented to
The Academic Faculty

By

Neela Lohith Penmetsa

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2014

MONOLITHIC 3D INTEGRATION OF ASYNCHRONOUS SYSTEMS

Approved by:

Dr. Sung Kyu Lim, Advisor
Professor, School of ECE
Georgia Institute of Technology

Dr. Saibal Mukhopadhyay
Associate Professor, School of ECE
Georgia Institute of Technology

Dr. Arijit Raychowdhury
Associate Professor, School of ECE
Georgia Institute of Technology

Date Approved: December 1st 2014

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The ever increasing demand for computing devices has been driving the innovation in consumer electronics and its enabling technologies. This has lead to active research and development in to alternative technologies. From the manufacturing perspective three-dimensional integration has recently come in to light showing potential, while from the circuit design perspective asynchronous systems are considered an interesting alternative.The goal of this thesis is to study the impact of 3D integration on asynchronous circuits and explore the benefits in power, performance and area compared to traditional two dimensional integration. To enable this study we develop a fully automated asynchronous design methodology and 3D integration flows for asynchronous circuits. This study is also a first one to explore the mutual benefits of asynchronous circuits and 3D integration. In this thesis, for the first time, we show that using 3D integration of a de-synchronized system can help achieve better power, performance and area compared to its traditional synchronous implementation.

CHAPTER 1

INTRODUCTION

The ever increasing demand for computing devices has been driving the innovation in consumer electronics and its enabling technologies. A significant effort by the semiconductor industry and the academia has gone in to the process of miniaturization of ICs by scaling the device and interconnect, which is now around 14nm node. Though several roadmaps predict [2] further scaling down to 10nm and 7nm, numerous obstacles in manufacturability and design process must be overcome before we get there. It is becoming increasingly difficult to deal with ultra-deep submicron issues while not compromising on design metrics such as performance, power, area, cost and time to market.

This has lead to active research and development in to alternative technologies. Some of these technologies attempt to address these challenges through newer manufacturing and fabrication processes while others through circuit design innovations. Several circuit design innovations have explored ways to address the challenges from process scaling. Of all the challenges that came up due to scaling, dealing with variability and power are the most fundamental. One approach to tackling variability issues in modern VLSI circuits is to exploit asynchronous design techniques. Asynchronous circuits are adaptive to variations [3] and can operate at a variety of voltage and frequency points [4]. However their disadvantages include greater design complexity and a negative impact to power, performance and area metric compared to their synchronous counterparts. This apart, lack of automation techniques for design and testability hindered their adoption in to the main stream.

Disruptive innovations in manufacturing technologies have given rise to three dimensional (3D) integration through die-stacking and monolithic integration. 3D ICs are expected to provide very high memory bandwidth and achieve better power and performance than traditional two-dimensional (2D) integration. The goal of this thesis is to study the

impact of 3D integration on asynchronous circuits and explore the benefits in power, performance and area compared to traditional two dimensional integration. To enable this study we develop a fully automated asynchronous design methodology and 3D integration flows for asynchronous circuits. This study is also a first one to explore the mutual benefits of asynchronous and 3D integration.

CHAPTER 2

BACKGROUND AND MOTIVATION

This chapter contains background information on two broad topics related to the thesis. The first is a description of de-synchronization theory with a survey of de-synchronized designs and the second is a description of 3D integration flow with focus on monolithic integration.

2.1 De-synchronization

De-synchronization is a methodology to covert a synchronous netlist to an asynchronous equivalent netlist. In this methodology, the global clock network is replaced automatically with a network of latch controllers which are connected in such a way that the data flow remains the same. It has been proved that [5] individual sequential element will have the exact data sequence as its synchronous counterpart. This allows for the application of standard synchronous testing techniques. Another important advantage of this methodology is that it can use existing commercial quality EDA tools thus the designer is not forced to have an in-depth understanding of asynchronous communication protocols. It is the only asynchronous methodology which can make use of exactly same RTL and standard cell libraries as a traditional flow. In this process a synchronous netlist is obtained by synthesizing RTL with synchronous constraints. The netlist is then modified to a de-synchronous netlist before proceeding with place and route flows. A detailed description of the implementation is given in the next chapter.

Figure 1 is an example of a simple synchronous design where data flows through sequential cells controlled by a clock. The clouds indicate combinational logic. Dashed lines are marked on the circuit represent different regions of the circuit where each region representing combinational logic and the sequentials it is driving. The regions are manually specified or can be derived automatically by clustering algorithms with in the tool. The

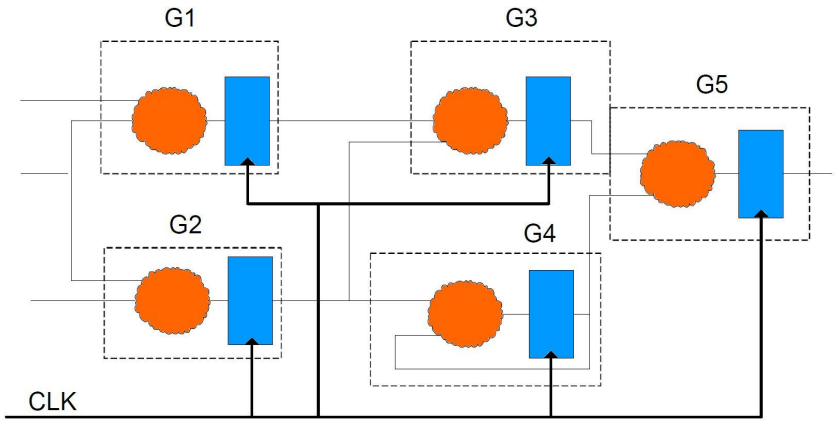

Figure 1. Sample synchronous circuit [1]

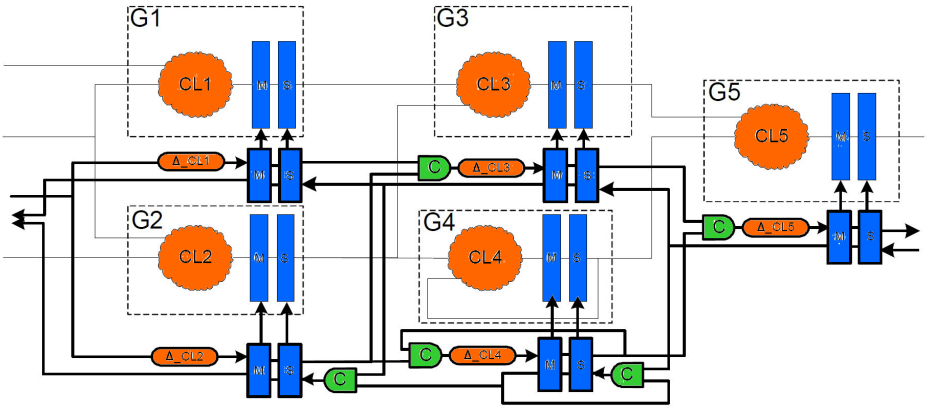

Figure 2. De-synchronized version of the sample circuit [1]

clustering process is a trade-off with the granularity of de-synchronization. As each region will have its own latch controller, more regions would imply more controllers which leads to a huge area over head. Figure 2 shows the fully connected de-synchronized version of the synchronous circuit. The final netlist contains c-elements, latch controllers and delay elements which are described in the next sections.

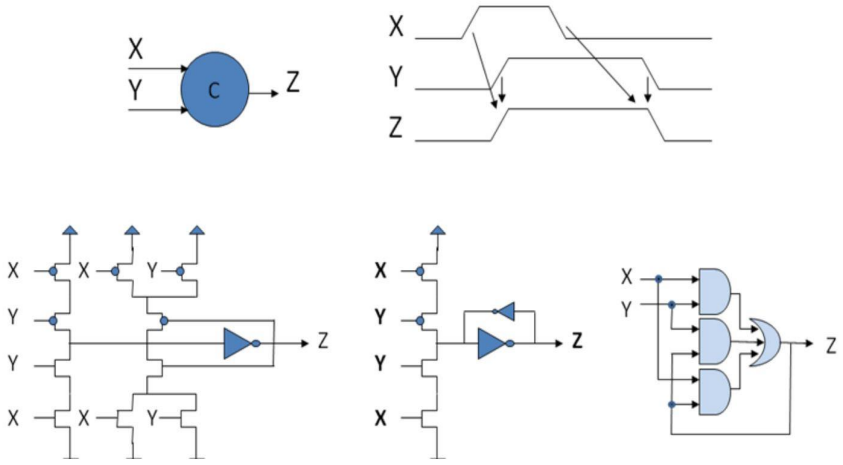

Figure 3. C-element realization styles

2.2 Elements of De-synchronization

2.2.1 The Muller C element

To design asynchronous systems with correct behavior, one must take a look at when signals are required to be valid. In synchronous systems the clock edge is used as an indicator of when all the signals are required to be valid. If the signals are not stable, its a hazard and the data is not acknowledged by the system. In asynchronous systems there is no such clock indicating the validity of signals, hence there must be an alternative way to keep track of various events in an asynchronous system. In the absence of clock means that, in many circumstances, signals are required to be valid all the time, that every signal transition has a meaning and consequently hazards and races must be avoided.

For this purpose a special gate is needed which can keep track and synchronize events in the circuitry. Muller C-elements are better in this regard for their state-holding capabilities like an asynchronous set-reset latch. For both inputs at logic 0 the output is 0 and for both inputs at logic 1 the output is 1 for other combinations the output does not change. The C-element and its truth-table are shown in Figure 3. Further details on its usage in asynchronous circuits is discussed in latch controller section.

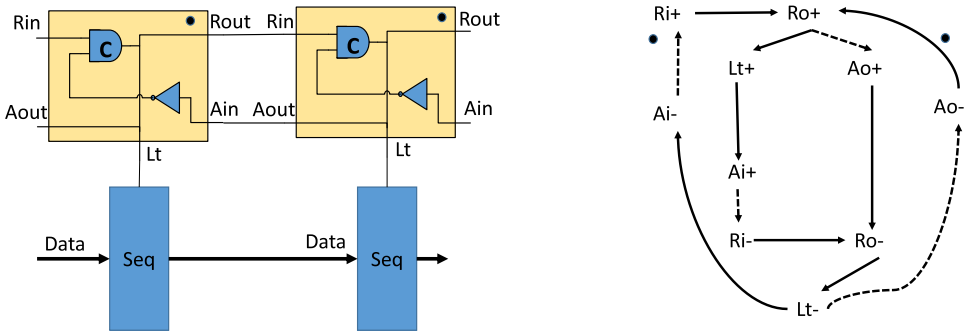

Figure 4. Simple 2-phase protocol

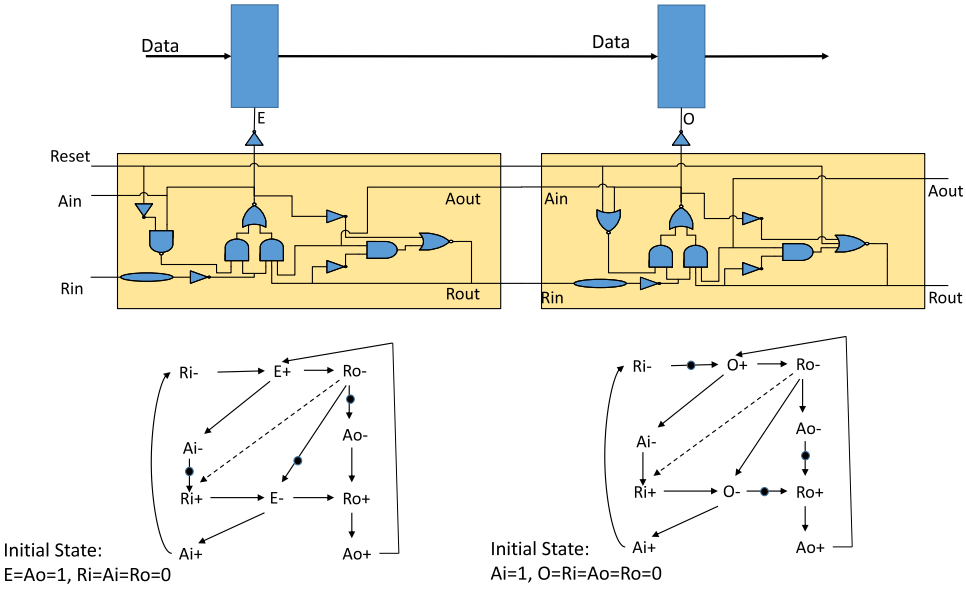

Figure 5. 4-phase semi-decoupled controller

2.2.2 Latch Controllers and Handshake Protocols

The main element used in the controller network is a latch controller which is an asynchronous circuit implementing a handshake protocol [6]. A simple 2-phase latch controller can be seen in Figure 4. On the left hand side of the figure, the signal Rin, i.e. the input request, indicates that the group of the predecessor controller(s) has (re)finished computing the output data, while the signal Ain, i.e. the input acknowledgment, signals a response to indicate that this group has processed its current data and they can be replaced by new ones. On the right hand side, we have the corresponding signals communicating with the successor controller(s). Thus, signal Rout, i.e. the output request, informs the target controller for the validity of this group's output data, while signal Aout, i.e. the output acknowledgment,

indicates that the target group has processed these data. Signal Lt, i.e. the latch enable, is used for driving a set of latches, while the signal Reset in a 4-phase controller, is used for the controller's initialization.

For flow-equivalent operation controllers may implement any handshake protocol suitable for De-synchronization [5], e.g. semi-decoupled, fully-decoupled or de-synchronization controller types are all valid. Signal Transition Graphs (STGs) of 2-phase and 4-phase protocols can be seen in along with their corresponding controllers. STGs are constrained PetriNets, which represent the signal dependencies and sequence. Figure 5 shows a 4-phase semi-decoupled controller. In this thesis, 2-phase controllers are used as they have been shown to be faster and simpler to construct. However the developed methodology can easily be extended towards using a different controller with minor modifications.

2.2.3 Delay Elements

The de-synchronized circuit has to respect setup constraints of its sequential elements. This implies that the combinational logic clouds have to be given enough time to compute their data. Since the request signal is the one that indicates when the logic has finished computing and there are valid data, these signals have to be appropriately delayed for so long as the combinational logic's critical path delay. There are two possible methods to achieve this, i.e. using delay elements to mimic the delay of the combinational logic or modifying the combinational logic and embed completion detection.

This work uses delay elements for mimicking the logic's delay. In this approach the request signals pass through a delay element before reaching the target controller. Thus, there is one delay element for each circuit region. A set of timing constraints are created such that the delay elements have a higher timing delay compared to its corresponding combinational paths. The delay elements can be implemented using straight forward inverter buffer chains or using a multiplexer to select the required delay as shown in Figure 6. The select value of the multiplexer can emulate data dependent delays.

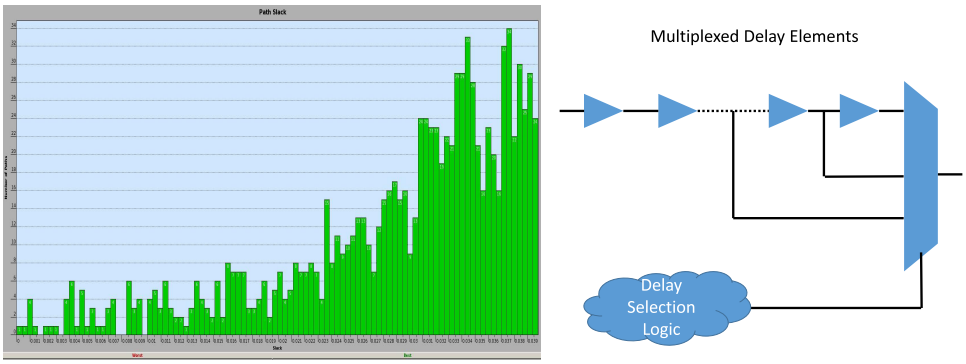

Figure 6. Slack distribution of a datapath and multiplexed delay elements

2.3 3D Integration

As demands accelerate for increasing density, higher bandwidths, and lower power, many IC design teams are looking up to 3D ICs with through-silicon vias (TSVs). 3D ICs promise more than Moore integration by packing a great deal of functionality into small form factors, while improving performance and reducing costs. 3D IC packages may accommodate multiple heterogeneous dies such as logic, memory, analog, RF, and micro-electrical mechanical systems (MEMS) at different process nodes, such as 28nm for high-speed logic and 130nm for analog. This provides an alternative to system-on-chip (SoC) integration, potentially postponing an expensive move to a new process node for all of the functionality developers want to place in a single package. Three-dimensional (3D) IC is expected to provide extremely high chip-to-chip bandwidth and achieve higher performance than traditional two-dimensional (2D) ICs. Black et al. studied the potential to achieve 15% power reduction as well as 15% performance gain of a high performance microprocessor by a 3D floorplan [7]. Kang et al. demonstrated 25% dynamic and 50% leakage power reduction in 3D DRAM [8].

2.4 TSV vs Monolithic integration

Currently, through-silicon vias (TSVs) enable 3D ICs, allowing vertical stacking of multiple dies fabricated separately. An emerging alternative is monolithic 3D that enables orders of magnitude higher integration density due to the extremely small size of the monolithic

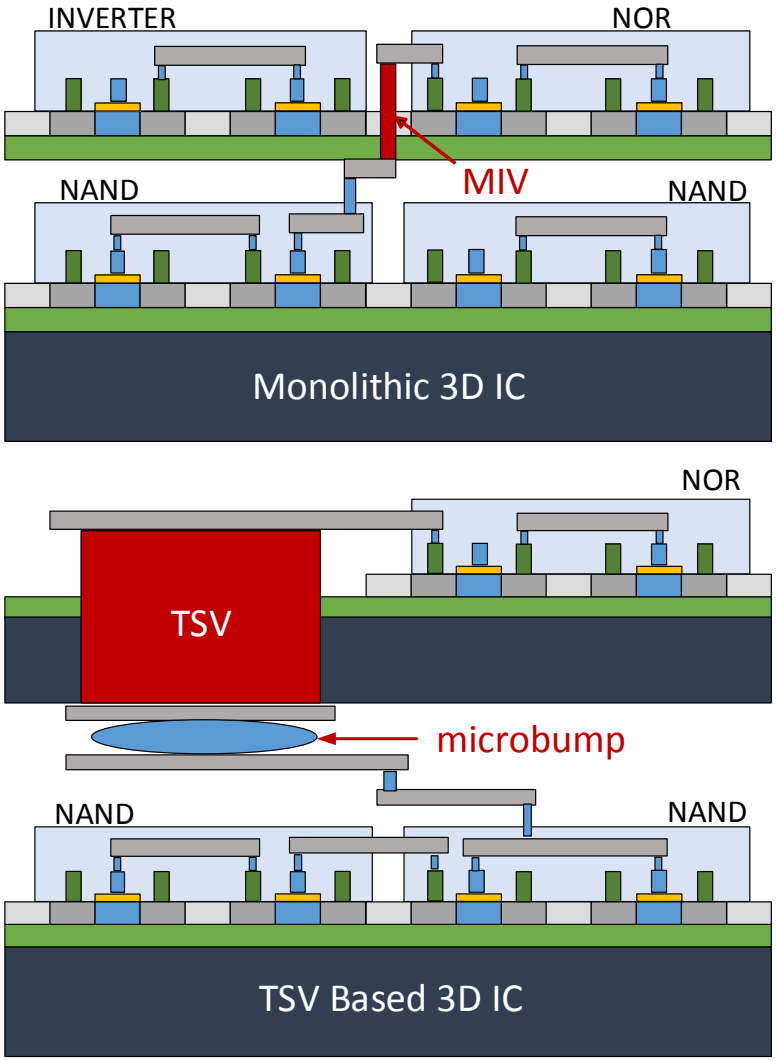

Figure 7. Comparison of TSV based die-stack and monolithic 3D integration

inter-tier vias (MIVs). Figure 7 shows both the 3D integration technologies. In monolithic 3D integration technology, one fabricates two or more tiers of devices sequentially, instead of bonding pre-fabricated dies. This eliminates the need for die alignment, enabling smaller via sizes. Overall, monolithic 3D ICs offer several advantages over traditional 3D ICs: (1) the small size of MIVs enables ultra-high integration density, considerably reducing silicon area and cost, (2) the significantly reduced MIV parasitics help improve the power performance envelope, and (3) the manufacturing process is entirely foundry-driven, and does not involve a packaging house for the processing of backside redistribution layers and micro-bumps. This enables tighter process control, potentially leading to a faster ramp up

once the technology is mature.

2.5 Monolithic 3D IC design flow

This section presents the sign-off CAD methodology for monolithic 3D ICs [9]. This methodology is based on the fact that the z-dimension is negligible in monolithic 3D ICs (only a few nm), which enables us to utilize commercial 2D IC tools to perform place and route for M3D. Consider a true 3D analytical placer that solves equations in the x,y, and z dimensions. Since we consider only the rectilinear half-perimeter wirelength (HPWL), each axis is independent of the other, and is therefore solved independently. Now, since the z dimension is so small (and discrete), all z solutions for a given x and y solution will have more or less the same HPWL. This implies that a 2D placer can be used to first find the x and y solutions, and the z location can be determined as a post-process. Note that this entire process is contingent on the 2D placer being able to place all the gates in a monolithic 3D IC footprint, which is half the foot-print area of a 2D IC. This requires several techniques to utilize the commercial 2D IC tool. In addition, memory complicates the issue, as they are pre-placed in both tiers, and this somehow needs to be fed into the commercial tool. The design flow is described in the next chapter in detail. First, in order to utilize the 2D tool to handle all the standard cells in a reduced foot-print, several technology files are scaled, and this process will be described in the next chapter. Next, memory handling requires several steps such as memory scaling, memory placement and memory flattening. Once this is done, the commercial 2D engine (Cadence Encounter) can be run on this shrunk 2D design. This result is then split into multiple tiers to obtain a DRC-clean sign-off design. Finally timing and power analysis is performed to obtain the design metrics.

2.6 Variation in 3D ICs

Die-to-die (D2D) and within-die (WID) variations in process parameters can lead to significant chip-to-chip variations in delay and power dissipation of ICs [10] . In 2-D ICs,

within-chip variation is determined by WID variations only. A three-dimensional (3-D) IC is composed of separate dies from different wafers and lots. Therefore, in a 3-D IC, both WID and D2D variations contribute to within-chip variations [11] . Moreover, variations in RC properties of through-silicon vias (TSVs) also add to total delay variations in 3-D ICs [6][9]. Hence, methodologies are required to reduce the effect of within-chip and chip-to-chip variations in 3-D ICs.

The performance and functionality of a digital circuit depend on the variations in logic delays and clock skews. The clock skew is defined as the difference between arrival times of the clock signal at different flip-flops. A higher clock skew worsens performance and/or robustness of a design.In 2-D ICs, WID variations change the delay difference between various branches of the clock tree, leading to increased clock skews. The D2D variation changes the delay of the entire clock tree and, hence, does not affect the clock skew significantly. On the contrary, clock skews in 3-D ICs are affected by both D2D and WID variations as both of them lead to within-chip variations.

Monolithic 3D ICs differ from TSV-based 3D ICs in that tiers are fabricated sequentially. The devices and interconnects of the top tier are fabricated on top of an already existing front end-ofline (FEOL) and back end-of-line (BEOL). During the processing of the top tier, care must be taken to prevent damage to the devices and interconnects of the bottom tier. If, however, we wish to use copper on the bottom tier, laserscan anneal has been proposed for the dopant activation on the top tier. This method only results in localized heating, thereby preventing any damage to the devices and interconnects on the bottom tier. However, this process results in degraded transistors, and the PMOS and NMOS performance degrade by 27.8% and 16.2% respectively [12].

Handling variation in 3D IC is extremely important as this might offset the performance benefit arising due to 3D integration. This is one of the main motivating factors to explore asynchronous circuits for 3D integration as they do not have a global clock and are proven to operate reliably when subjected to process variations.

CHAPTER 3

DESIGN METHODOLOGY AND IMPLEMENTATION

This chapter presents the design and implementation of both synchronous and asynchronous versions of the AES encryption core using monolithic 3D IC technology. This experiment is done to study the power, area and performance savings compared to a traditional 2D IC implementation. It is mutually beneficial to combine the domains of asynchronous and 3D integration as their respective strengths and weaknesses complement each other. Asynchronous circuits supplement 3D ICs with better thermal control, power supply integrity and variation tolerance. In return, 3D ICs help manage the power, performance and area overheads of asynchronous circuits. Our study is based on GDSII layouts and industry standard sign-off analysis flows.

3.1 Benchmark Design

In this work a custom, high performance pipelined Advanced Encryption Standard RTL is implemented. The ubiquity and the importance of the AES core is the main motivation behind its selection. AES encryption cores are part of thousands of real products with a diversity of form factors ranging from ultra-low power sensor networks to high performance server processors. Typically, depending on the end product's target encryption rate, AES cores are designed for various throughput speeds. Figure 8 shows the top level architecture of the AES encryption standard. It takes a plain input text and an AES key and performs 10 rounds of data transformations on it to generate the encrypted output. The current AES implementation used for this work is optimized for encrypting 128-bit data packets into a 128-bit cipher text using an AES key of the same size. The design is a deep pipelined architecture, which dumps out encrypted data packets every clock cycle, with an input to output latency of 41 clock cycles. Standard data packets and their pre-encrypted ciphers are used to functionally validate the design.

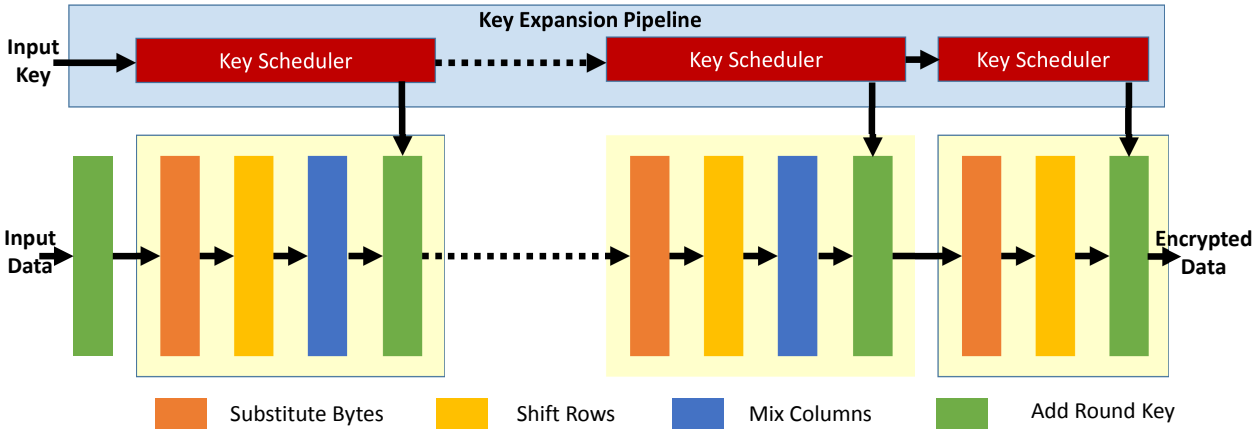

Figure 8. AES architecture

3.2 Logic Synthesis and De-synchronization Flow

As discussed in the previous chapter, this work uses a de-synchronization methodology [5] that presents a fairly simple framework to convert a synchronous gate-level netlist into an asynchronous one. A high level conversion flow is shown in Figure 9. First the AES RTL is synthesized using design compiler and traditional synchronous constraints. The final netlist after synthesis is de-synchronized using the following steps:

1. Modification of design hierarchies to facilitate de-synchronization.

2. Conversion of the flip-flops in synchronous design to latches. Here we split each flip flop in to its corresponding master and slave latches.

3. Automated region creation for de-synchronization. In this step, we assign each standard cell of the netlist to a de-synchronization region which is controlled by its corresponding handshake controller.

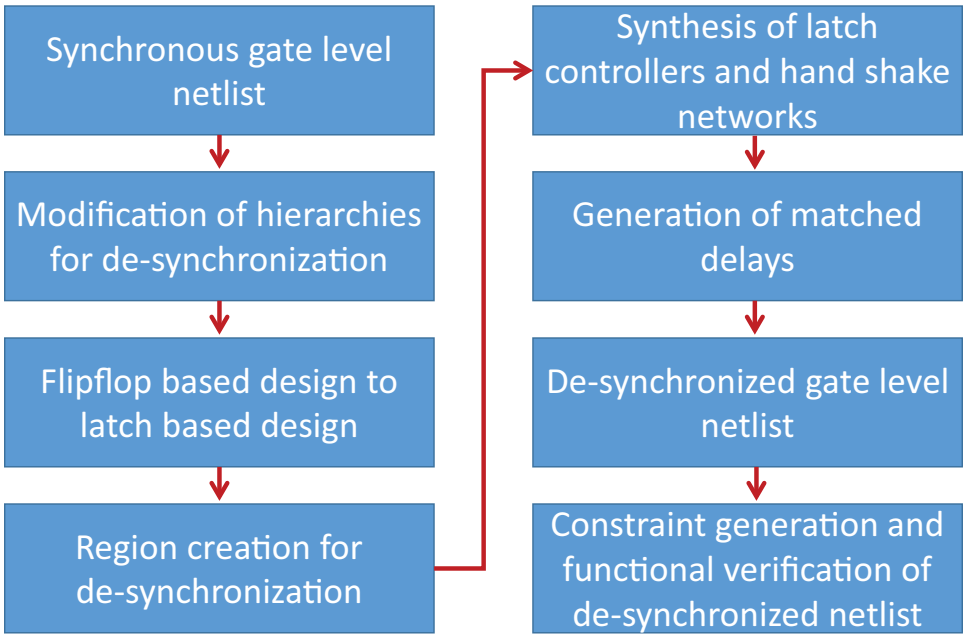

Figure 9. De-synchronization flow overview.

4. Synthesis of 2-phase latch controllers for implementing handshake protocols between de-synchronized regions.

5. Automated generation of matched delays. Delay chains are inserted in to the netlist which are greater than or equal to the corresponding combinational path delay as shown in Figure 10. The delay chains act as completion detection handshake signals. We implemented the delay chains with higher V_t cells as this ensures that the delay chain is always slower than the combinational path, even at lower V_{DD}.

6. Constraint generation: Data timing check points are extracted from the synchronous netlist and are used to generate the timing constraints for the de-synchronization flow which can aid optimization during the place and route stages.

3.3 2D Physical Design Flow

Current study is based on a 28-nm PDK. We take the design through standard design stages like floorplanning, placement , clock tree synthesis, routing and physical verification. The post-routed databases are used to perform parasitic extraction. The GDSII-level design

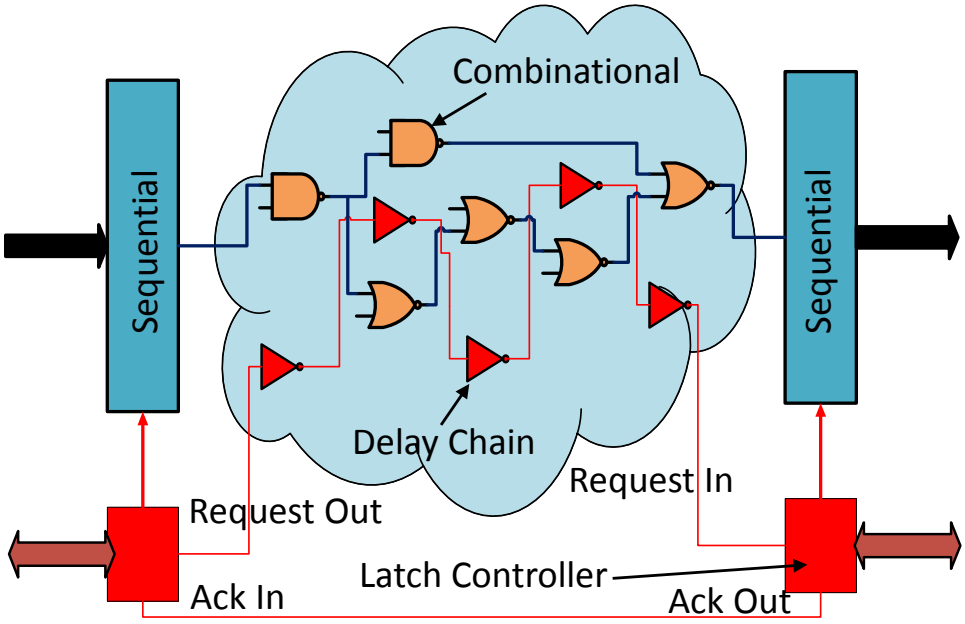

Figure 10. Synthesis of handshake controller and insertion of matched delays.

data is then analysed using industry standard tools like PrimeTime. While handling de-synchronized designs, it is ensured that the delay chains are placed near the respective combinational logic to track variations more closely. We also break any timing loops caused by the handshake controllers manually, as the synchronous 2D tool is not capable of recognizing them. Next a pseudo clock-tree-synthesis is performed to distribute the latch triggering pulses from the handshake controller to the latches. Finally, after the routing stage of de-synchronized design, a recalibration step is done to fine tune the delay chains to account for variations in delays due to the placement and routing steps.

3.4 3D Integration Choice: TSV vs Monolithic

Initial preference for this work was to use a TSV based integration. In a TSV based integration the netlist is partitioned in to two tiers using a simple min-cut algorithm. The min-cut strategy strives for an area balance between both the tiers while minimizing the cut-size which is equivalent to the TSV count. As shown in the Figure 11 the De-synchronized netlist has several regions where each region only communicates with a limited number of adjacent regions. Partitioning regions on to multiple tiers would lead to half the regions

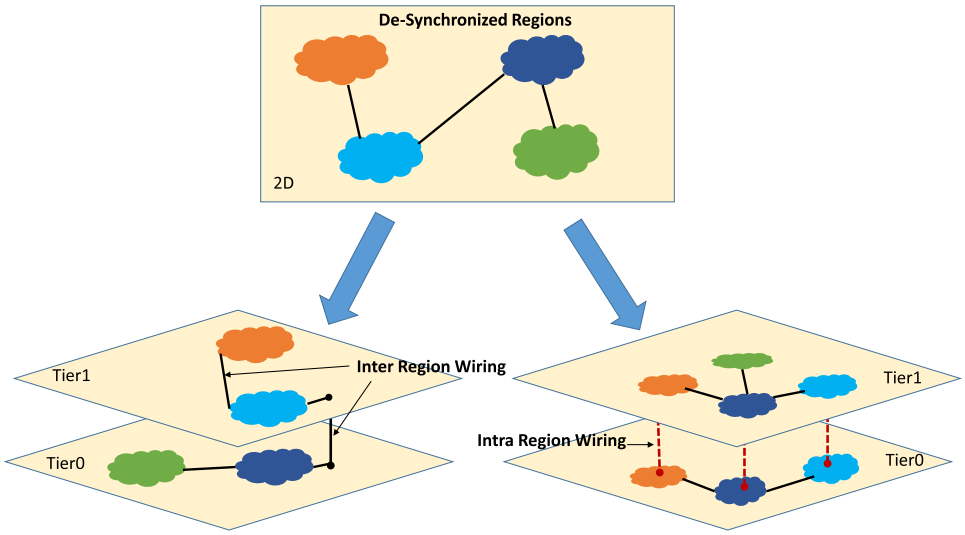

Figure 11. Partitioning of De-Synchronized regions

split on to one die while other half on to another die. In this strategy TSVs are used for inter-region wiring. Since each region is still effectively a 2D design in itself, not much benefit is obtained from this strategy. Min-cut experiments with this style of partitioning lead a 2 tier design with 230 TSVs to achieve the required area balance on both the tiers. A second type of folding strategy is shown in the same picture. In this style of partitioning each region is folded on to multiple tiers with TSVs used for intra-region wiring. The advantage of this strategy is each region is now split on to multiple-tiers there by enabling effective optimization of intra-region interconnects. However using TSVs for this folding scheme has its own drawbacks. Since a typical TSV size is about 5um x 7um, their count has to be limited to about 15-20% of the total die area. This would put a limitation on the number of intra-region wires crossing the tiers thus limiting the optimal solution. Secondly the area overhead due to the TSVs as they take up considerable silicon area adds to the burden from de-synchronization. Hence what ever area and performance benefit achieved by 3D integration would be offset by this over head. Finally, TSV parasitics play a significant role in timing as they add a considerable amount of capacitance depending on various factors. This will indirectly impact the timing and performance of the de-synchronized design. Taking all the above factors into consideration a decision was taken to use monolithic

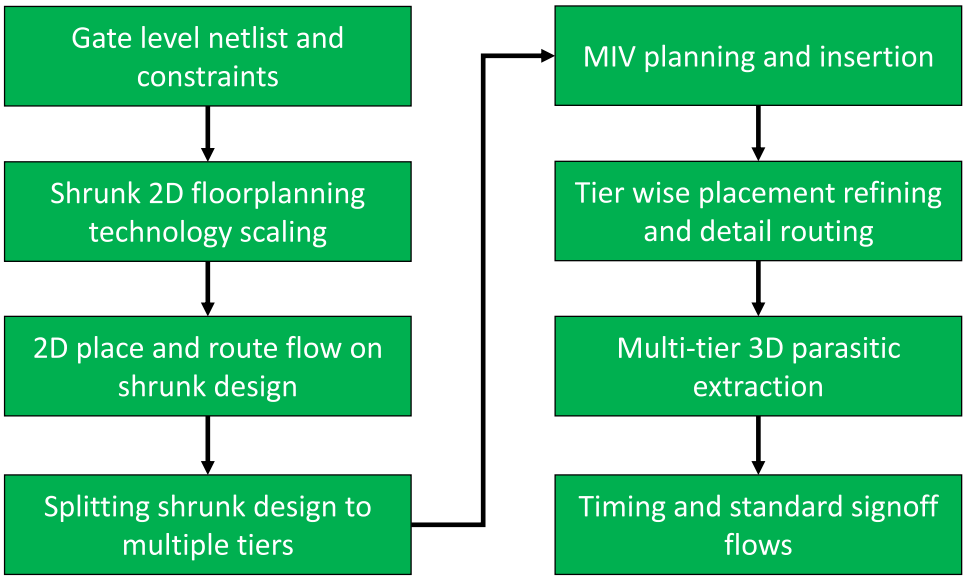

Figure 12. Monolithic 3D flow

integration approach for this work. Monolithic Inter-tier VIAs (MIV) have very small sizes compared to TSVs (in the order of 100nm) and present significantly low parasitics. This allows us to use a large number of MIVs with minimal impact to area or performance. Such an approach would be suitable for intra-region folding.

3.5 3D Physical Design Flow

This section presents the description of our in-house RTL-GDSII CAD flow for monolithic 3D ICs [13]. A mix of industry standard tools and custom in-house tools is used in this approach. In this work, we focus only on two tier designs. A block diagram of the flow steps is shown in Figure 12. Once we obtain a gate-level synthesized netlist, we make use of an industry standard tool (SoC Encounter) to place all the standard cells on to a shrunk footprint corresponding to that of a monolithic 3D IC. In order to do this, first the chip width and height are shrunk, as well as the width and height of all the standard cells by a factor of 0.707. Then the traditional 2D flow is run as described in the 2D physical design flow section to obtain a shrunk 2D design.

The next step is to split the shrunk 2D design into multiple tiers to obtain a DRC clean

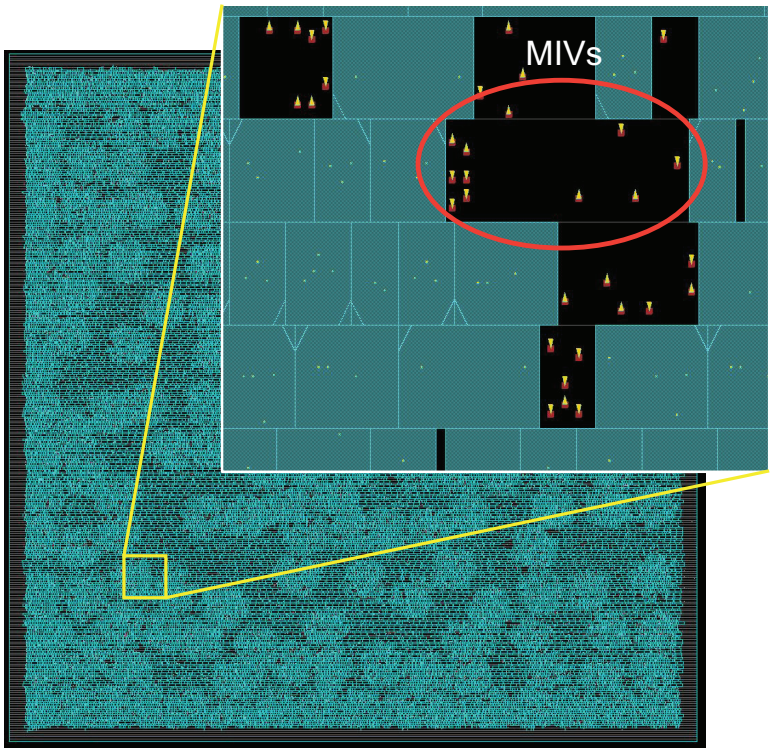

Figure 13. MIVs are inserted into the whitespace between the standard cells.

design with MIVs inserted into the whitespace between the standard cells (Figure 13). There are various sub-steps involved here. First, all the standard cells are expanded back to their original sizes, which will cause a lot of overlaps in their placement. Next, placement bins are created in a traditional fashion. A partitioner is then used to split the cells from each bin onto top and bottom tiers such that area balance is maintained within each placement bin. Once this step is completed, each tier is routed separately and a tier-level parasitic extraction is done. Then custom tools are used to create a 3D parasitics database by stitching all the individual tiers and MIV parasitics together. In the final stage, this information is used along with 3D netlists to perform timing and functional sign-off flows.

3.6 Partitioning of Delay Chains

In addition to monolithic 3D flow presented earlier, the delay chains must be partitioned in 3D de-synchronized designs. As we perform intra-region folding, several timing paths snake across tiers as shown in Figure 14. The delay chains must be partitioned in such a

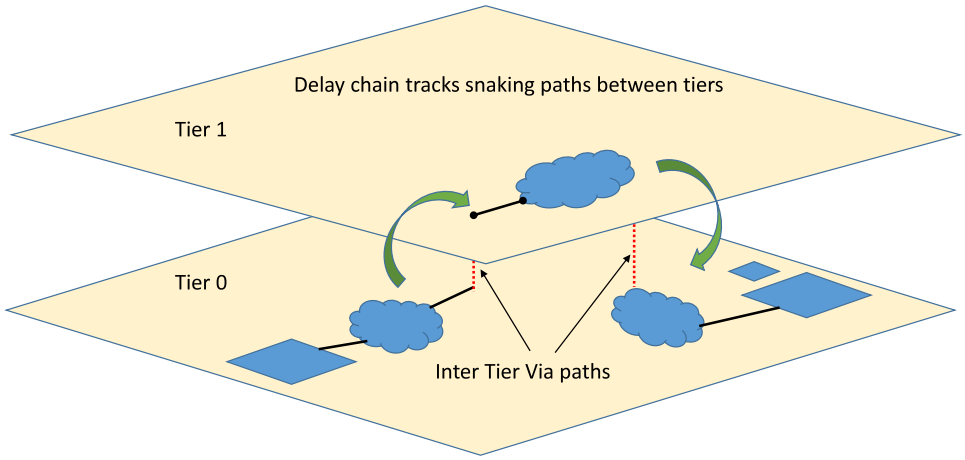

Figure 14. Snaking timing paths in 3D designs

way so they track the snaking paths across tiers. This way the delay chains can respond to tier-tier variations in timing paths. This can help make the 3D de-synchronized design more tolerant to tier-tier variations compared to its synchronous counterpart.This step is done after the netlist partitioning and MIV placement step. A custom script analyzes the number of tier-tier transitions in the combinational paths and corrects its corresponding delay chains to reflect the similar transitions across tiers.

CHAPTER 4

RESULTS AND ANALYSIS

4.1 Functional Verification and Power Simulations

PrimeTime-based timing analysis on all the designs is done using the extracted parasitics and the post-routed gate-level netlists. From this timing analysis, timing delays are extracted for each cell of the design into a standard delay format (SDF) file. This file is used to back-annotate timing delays in gate-level functional simulations. Both synchronous and de-synchronous designs are functionally verified with real time encryption work loads. Basic system level verification of the de-sycnronized design is shown in Figure 15 The advantage of de-synchronized design is its ease of interfacing with other synchronous designs. Input request and output acknowledge of the de-synchrnoized blocks can be driven by an external interface clock while ignoring their corresponding acknowledge and request signals respectively. As shown in Figure 16, in de-synchronized designs, an external clock is used to shift out the encrypted data from the final pipeline stage. Several pre-calculated encryption work loads are used to verify correctness of operation and generate a value change dump (VCD) file containing the switching activities of all the gates. We use this file for accurate real time power simulations.

4.2 Footprint and Wirelength Reduction

Both synchronous and de-synchronized designs are implemented in 2D and monolithic 3D. Various key metrics such as wirelength, footprint area, cell area and buffer count are presented in Table 1. This work primarily focuses on ISO-performance comparisons, and hence the critical path delays of all implementations have been optimized to be 0.25ns. This bound is decided because of the speed limitation from the 2D de-synchronous design.

From Table 1, we first observe that while the 2D footprint is forced to be the same between synchronous and de-synchronized designs, the cell area in the later goes up. This is

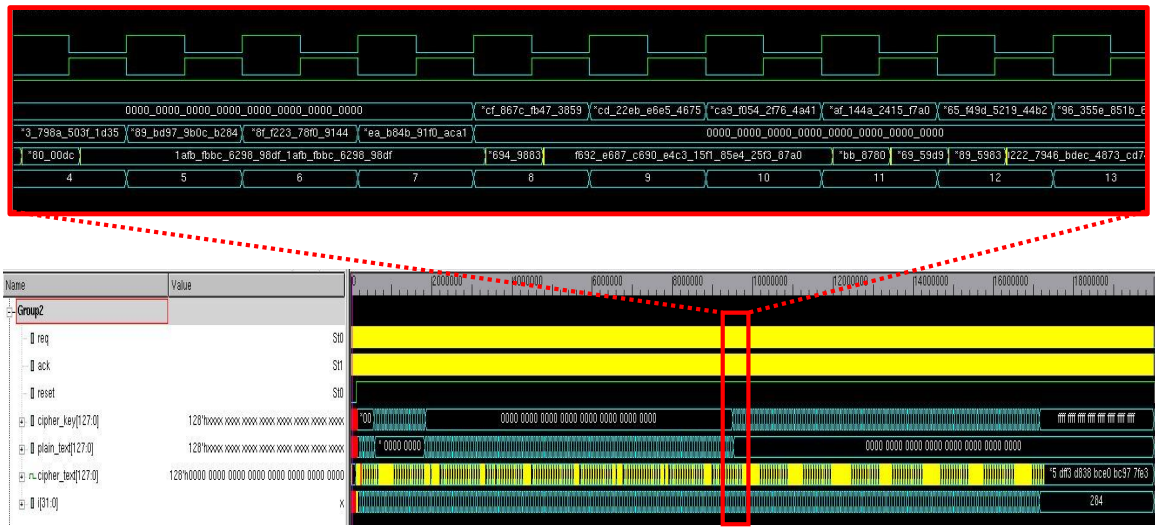

Figure 15. Functional Verification of the De-Synchrnonized design

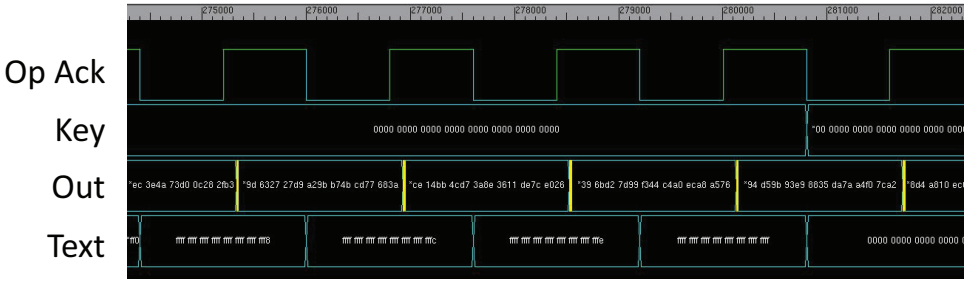

Figure 16. OpAck clock is used to shift out the data from last de-synchronized pipeline stage.

because de-synchronized designs can reach a slightly higher utilization than synchronous counterparts due to the absence of global interconnects. Each de-synchronized region only interacts with its neighboring region which facilitates a tighter packing. However, we observe that de-synchronized design has higher buffer count and total wirelength. This is due to the area and interconnect overhead from various handshaking controllers. This matches existing literature, where asynchronous designs have an area and wirelength penalty compared to their synchronous counterparts. This is one of the reasons asynchronous designs are are not widely used today. Note that the average wirelength is lower in de-synchronized designs due to the absence of long global connections.

To overcome these limitations in de-synchronized 2D, it is implemented in a monolithic 3D fashion. The footprints and routed die-level screenshots of all implementations are

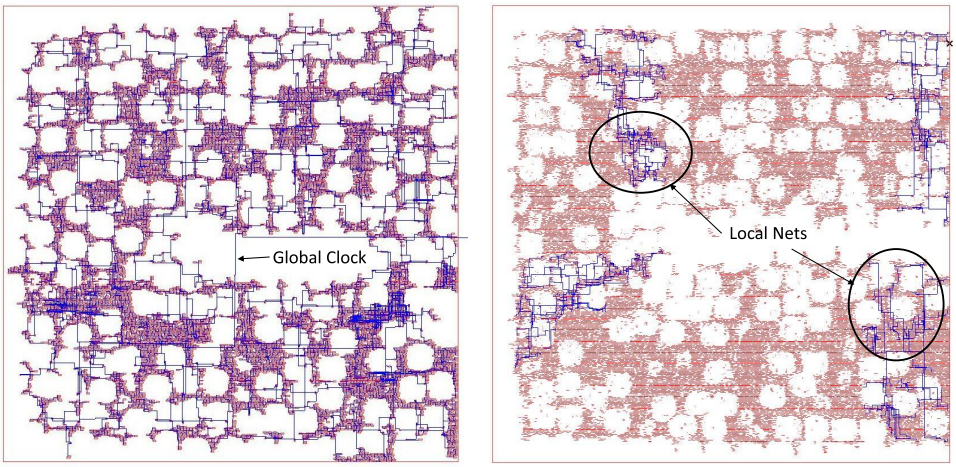

Figure 17. Localized interconnects in De-Synchronized designs

Table 1. Iso-performance (0.25ns) comparison for various implementation flavors. WL is wirelength

	Synchronous		Desynchronous	
	2D	3D	2D	3D
footprint (mm^2)	0.504	0.25 (-50.3%)	0.504	0.25 (-50.3%)
cell area (mm^2)	0.400	0.373 (-6.80%)	0.425	0.399 (-6.06%)
buffer count	31757	26440 (-16.7%)	34292	29834 (-13.0%)
total WL (m)	3.03	2.09 (-31.0%)	3.06	2.01 (-34.3%)
avg WL (um)	20.27	14.582 (-28.1%)	18.20	13.18 (-27.5%)

shown in Figure 18. From this figure and Table 1, we see that 3D offers a 50.3% footprint reduction over 2D. 3D ICs can operate faster than our target timing constraints, but since we are performing iso-performance comparisons, we can trade performance for power saving. Optimizing 3D ICs for a frequency less than what they are capable of will lead to significant buffer count and power savings.

As a result of the footprint reduction and close proximity of cells in 3D designs compared to 2D, we see significant reduction of wirelength in 3D designs. From de-synchronized 2D to de-synchronized 3D, we see about 34.3% reduction in total wirelength and 27.5% reduction in average wirelength. This leads to de-synchronized 3D having lower wirelength and using fewer gates overall than the 2D synchronous design. Therefore, monolithic 3D IC technology can overcome all the shortcomings of this asynchronous design style. Next section discusses how asynchronous operation helps monolithic 3D.

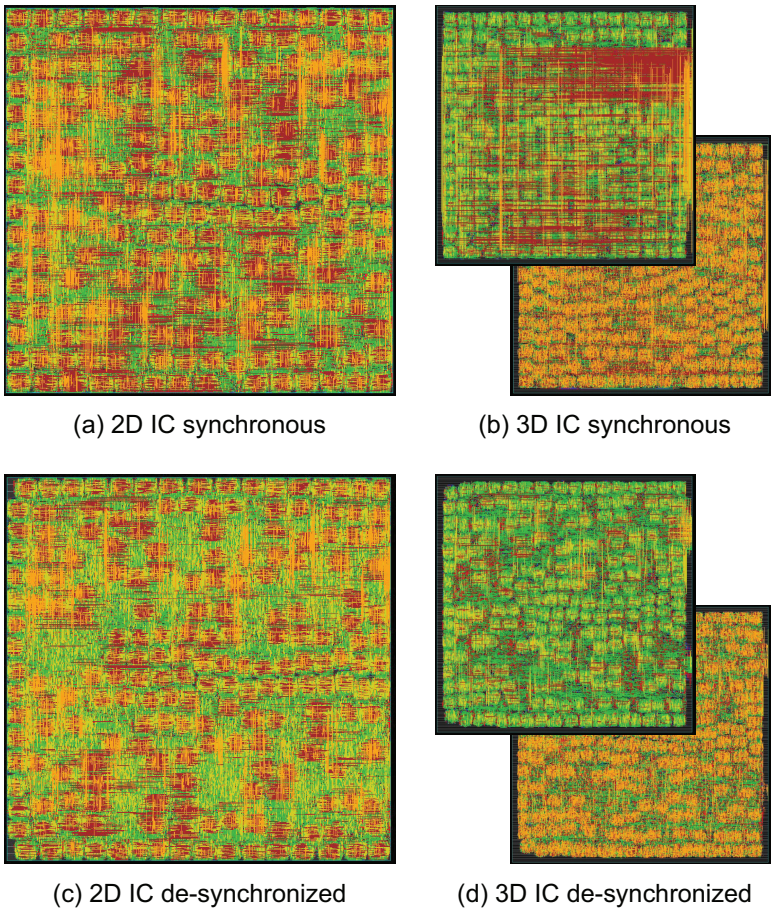

Figure 18. GDSII Layouts of 2D and 2-tier 3D synchronous and de-synchronized AES designs. 2D footprint is 710x710um, and 3D is 500x500um. We observe that de-synchronous has fewer global interconnects.

4.3 Power Reduction

The power results obtained from vector based power simulations are presented in Table 2. We observe that de-synchronized 2D consumes about 9.2% more power than its synchronous counterpart. This power overhead is due to the handshake controllers and splitting of flip-flops into master-slave latch pairs, and is in line with the results in the previous section. After analyzing final 3D and 2D designs with standard real time test vectors, we observed significant power savings in de-synchronized 3D of up to 25.7% total power reduction compared to de-synchronized 2D and 18.9% percent reduction compared to 2D synchronous.

As mentioned in the last section, 3D can meet the timing target more easily, and hence

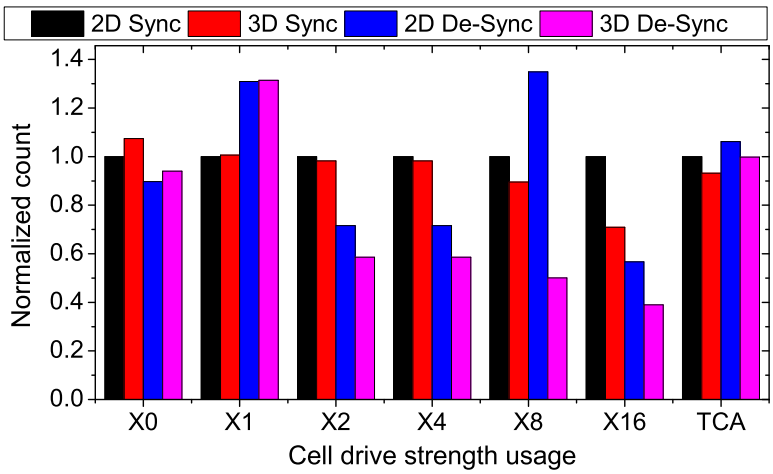

Figure 19. Comparison of cell usage of various drive strengths normalized to 2D-Sync (X0 being the smallest). TCA is Total Cell Area.

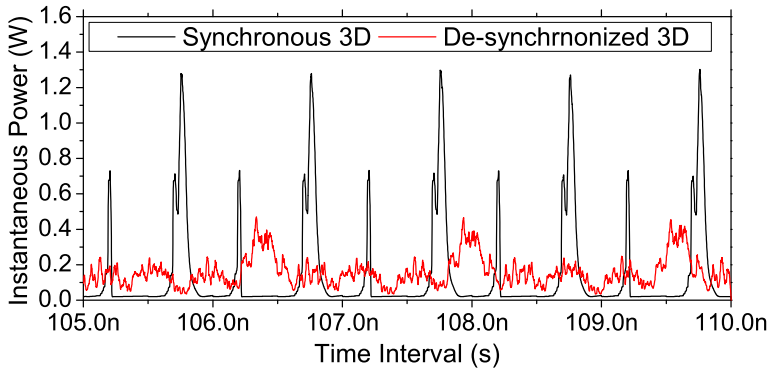

Figure 20. Transient power analysis of 3D Sync and 3D De-sync

uses fewer gates overall. This effect is quantified in Figure 19, where we plot the cell usage in each design grouped by size. We observe both fewer cells overall, as well as fewer larger cells. This also leads to a reduction in the total cell area as shown in this figure.

So far, we have discussed the benefits monolithic 3D brings to asynchronous. However, asynchronous operation also mitigates many potential issues in monolithic 3D ICs. Although there is a slight increase in average power from 3D synchronous to 3D de-synchronous, we see a huge reduction of 63.9% in terms of peak power (Table 3). Peak current is a primary concern in the design of power distribution networks especially for 3D ICs. Such peaks determine the maximum voltage drop and probability of failure due

Table 2. Power comparison of 2D and 3D designs in Watts

	Synchronous		De-Synchronous	
	2D	3D	2D	3D
Switching power	0.1171	0.0824 (-29.6%)	0.1361	0.0981 (-27.9%)
Cell power	0.0529	0.0423 (-20.0%)	0.0513	0.0372 (-27.4%)
Leakage power	0.0221	0.0198 (-10.4%)	0.0225	0.0205 (-8.88%)
Total Power	0.1921	0.1444 (-24.8%)	0.2098	0.1557 (-25.7%)

Table 3. Comparison between the designs for peak power consumption in Watts

2D Sync	2D De-sync	% change	3D Sync	3D De-sync	% change
1.39	0.602	-56.6	1.302	0.47	-63.9

to electro-migration. This may lead to performance gaurdbands in 3D ICs, which asynchronous operation helps gets rid off. Since 3D ICs have double the thermal density of 2D designs, it is critical to reduce thermal fluctuations. These fluctuations make the heat removal process more difficult and may penalize design metrics. We have characterized the power spectrum of 3D synchronous and 3D de-synchronous designs based on standard real time encryption workloads. As shown in Figure 20, 3D de-synchronous has the best power profile with almost negligible fluctuations compared to its synchronous counterpart.

4.4 Performance Benefit

All the previous results have assumed that asynchronous and synchronous have an identical worst case stage delay of 0.25ns. Our AES core has 41 such stages as it is pipelined for maximum throughput. In a synchronous system, the operating frequency is limited by slowest stage which naturally slows down the faster stages. However, in the de-synchronized design, since every stage is locally timed, the latency of the circuit is equal to the sum of delays in each pipeline stage. When a single packet of data is sent for encryption, we observe that the synchronous design has a total input to output latency of 10.25ns. In contrast, the de-synchronized design has a total latency of 6.33ns, which is a significant improvement.

We have also designed for the best performance that each implementation flavor can achieve. 2D synchronous can achieve a critical path delay of 0.24ns while 3D synchronous

is 20% faster with a critical path of 0.20ns. Similarly, 2D de-synchronous can achieve a critical path delay of 0.25ns while 3D de-synchronous is 16% faster with a critical path of 0.21ns. We still observe that 3D de-synchronous can operate 12.5% faster than 2D synchronous.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this thesis, for the first time, we studied the synergistic benefits of 3D IC and asynchronous circuits. It is demonstrated that the power, performance and area overhead in asynchronous designs can be reduced significantly by using 3D IC integration. At the same time, asynchronous can help 3D IC designs with better power supply integrity and thermal characteristics. By switching to 3D, we obtain significant footprint reduction of the AES core, which facilitates encryption capabilities into products of various form factors. At the same, time de-synchronization gives the 3D IC-based AES design modular capabilities and mitigates some of its negative effects. However to fully establish the fact that asynchrnous techniques are very robust with respect to the variations in 3D ICs a thorough study on variation analysis must be conducted. This can be an interesting work to pursue in the future.

REFERENCES

[1] A. Nikolaos, "A Fully-Automated Desynchronization Flow for Synchronous Circuits," 2006.

[2] "International Technology Roadmap for Semiconductors (2013 Update)," http://www.itrs.net.

[3] O. C. Akgun, J. Rodrigues, and J. Sparso, "Minimum-Energy Sub-threshold Self-Timed Circuits: Design Methodology and a Case Study," in *Proceedings of the International Symposium on Asynchronous Circuits and Systems*, 2010.

[4] M. Lotse, M. Ortmanns, and Y. Manoli, "A Study on self-timed asynchronous sub-threshold logic," in *Proc. IEEE Int. Conf. on Computer Design*, 2007.

[5] J. Cortadella, A. Kondratyev, L. Lavagno, and C. P. Sotiriou, "De-Synchronisation: Synthesis of Asynchronous Circuits from Synchronous Specifications," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2006.

[6] J. C. I. Blunno, A. Kondratyev, K. L. L. Lavagno, and C. Sotiriou, "Handshake protocols for de-synchronization," in *Proceedings International Symposium on Advanced Research in Asynchronous Circuits and Systems*, 2004.

[7] B. Black *et al.*, "Die Stacking (3D) Microarchitecture," in *Proc. Annual Int. Symp. Microarchitecture*, 2006.

[8] U. Kang *et al.*, "8 Gb 3-D DDR3 DRAM Using Through-Silicon-Via Technology," in *IEEE J. Solid-State Circuits*, 2010.

[9] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Design and CAD Methodologies for Low Power Gate-level Monolithic 3D ICs," in *Proc. Int. Symp. on Low Power Electronics and Design*, 2014.

[10] J. T. K. J. Tschanz, S. Narendra, D. A. R. Nair, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," in *IEEE J. Solid-State Circuits*, 2002.

[11] C. O. F. Akopyan, D. Fang, S. J. Jackson, and R. Manohar, "Variability in 3-D integrated circuits," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2008.

[12] B. Rajendran, R. S. Shenoy, D. J. Witte, N. S. Chokshi, R. L. DeLeon, G. S. Tompa, and R. F. W. Pease, "Low Thermal Budget Processing for Sequential 3-D IC Fabrication," in *IEEE Trans. on Electron Devices*, 2007.

[13] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Placement-Driven Partitioning for Congestion Mitigation in Monolithic 3D IC Designs," in *Proc. Int. Symp. on Physical Design*, 2014.