Back-End Design of Application-Specific Integrated Circuits (ASICs) *

Damilola Caleb Awosuru
Electronics Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
damilola-caleb.awosuru@stud.hshl.de

***Abstract*—ASICs (Application Specific Integrated Circuits) are integrated circuits that are manufactured for specific applications. The steps involved (design flow) in manufacturing ASICs are specification extractions, design planning, front-end (i.e. logic) design, and back-end design (i.e. the physical implementation) [1]. The goal of this paper is to dive deep into the physical implementation of ASICs, the necessary steps and best practices involved. A poorly designed back-end (physical implementation) may lead to failure, poor performance, or errors within the IC (integrated circuit); congestion, routing issues, clock errors and design rule check fails are common errors when designing ICs that can be prevented with the proper back end design.**

***Index Terms*—ASIC, IC, back end, physical implementation**

I. Introduction

ASIC design is generally broken down into two stages; the **front-end design** and the **back-end** design. The steps involved in the front-end design of ASICs are real-time logic design, real-time logic verification, test, and scan insertion, equivalence checking, and pre-layout STA (static time analysis) [1]. The RTL (real-time logic) design involves the use of a hardware description language to specify the function of the ASIC, in this step the architecture of the ASIC is designed and implemented. Immediately after The RTL design is the RTL verification, software tools like Vivado or Modelsim are used to simulate the functionality of the design, and if errors are found changes are made to the RTL design (back and forth) until the primary objective of the design has been verified. Once the functionality of the design has been verified the RTL code is then synthesized to the respective gate level, the Xilinx EDA tool Vivado, and the Synopsys Design Compiler are popular synthesis tools that are used in this stage of ASIC design. The test and scan insertion are usually for designs that contain memory elements [2], the goal of this stage is to catch manufacturing defects [1]. To maintain the design intent an equivalence checking is carried out to check the logic equivalence with the RTL architectural design. The final and important step in the front-end design stage is to check the timing, thus the term ”pre-layout” static timing analysis is coined, the goal of this stage is to find timing violations. At the end of the front-end design the netlist is generated, the netlist describes the interconnection of blocks (e.g CPU, RAM, ROM ALU), the logic cells that implement the blocks, and the connectors. The back-end design flow of an ASIC includes floor planning, power planning, clock tree synthesis, place and route, layout versus schematic (LVS), design rule check (DSRC), static timing analysis (STA), and generic/geometric Data Structure Information Interchange (GDSII), all of this steps will be discussed in detail in this paper, figure 1 is a visual representation of the steps.

The gate-level netlist gotten from the **front end design** flow, the chip constraints, required libraries, pre-defined floor plan, the timing, and design constraints are the inputs (i.e requirements) needed to start the back-end design flow [4].

As shown in figure 1, the steps involved in the back-end design are iterative [3], and the steps are repeated to get the best possible output and meet the specifications set by the customer or physical designer.

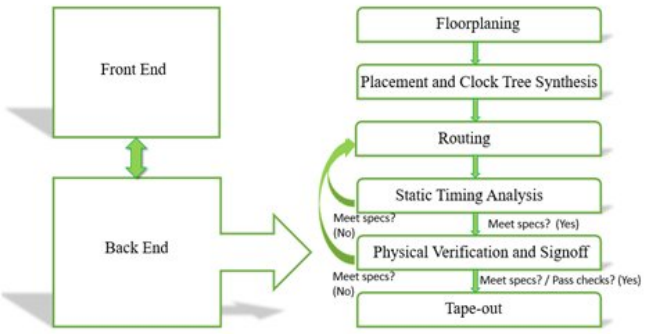

Fig. 1. Back end design flow [4].

II. Floor planning

An ASIC design with higher performance and optimum area can be achieved with a well-thought-out floor plan [3].

The floor planning is the mapping of the logical description of the ASIC gotten from the netlist to the physical description

[4]. In floor planing the logic blocks are arranged, the I/O pads (the logic cell connectors) are positioned, and the number, type, and distribution of power pads and clock are decided [3] [4].

A. Power and I/O

For an ASIC to function properly the power constraints and I/O pads must be selected and placed properly [5]. The main goal of the pad placement phase is to have the three pads: the power, ground, and signal be placed properly and have sufficient power and grid connection to avoid excessive electrical current in the direction of the electron flow, to prevent the molecular transfer of metal from one area to another, leading to ASIC design failure (i.e. Electromigration Damage) shown in figure 2 [3].

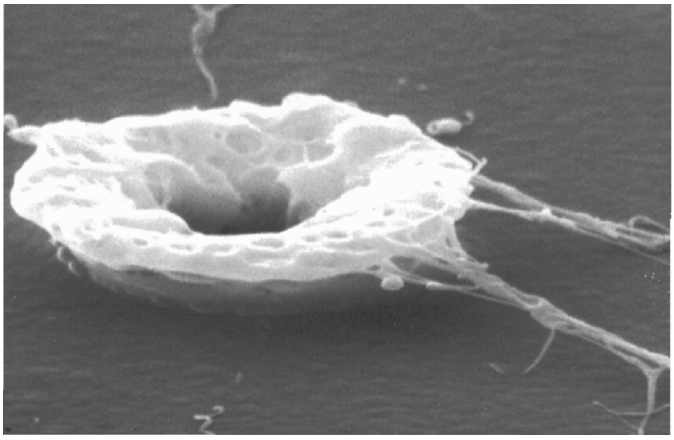

Fig. 2. Electromigration Damage [3].

The power and ground busses of the I/O pads are built into the pads [5], the sum of the static and dynamic currents (I_{total}) in amperes divided by the maximum electromigration current (I_{max}) in amperes per ground pad gives the minimum number of ground pads required to prevent electromigration damage [3].

$$N_{gnd} = \frac{I_{total}}{I_{max}} \tag{1}$$

There is no need to compute the number of power pads because it is the same as the number of ground pads.

As shown in figure 3, a set of wires are each connected to the ground and power pads to provide ground and power in the ASIC.

B. Macro-Placement

After the definition of the ground and power bus structures and the I/O placement, the macro placement takes place [3]. The quality of the ASIC is greatly impacted by the proper placement of macros. One major criterion for proper macro placement is to ensure that there is enough area between the macros for interconnection [3], this can be achieved either manually or automatically by floorplan tools like Vivado. Macro placement is repeated until optimization is achieved;

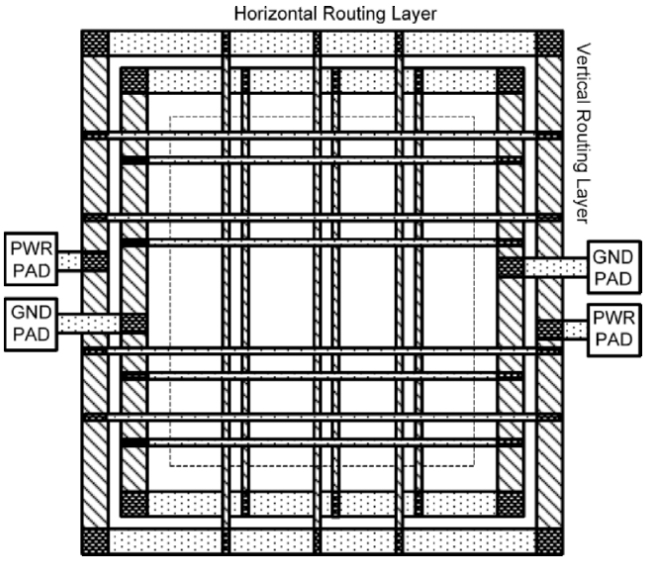

Fig. 3. Internal power strips [3].

placement that involves few macros is carried out manually, however, when dealing with a large number of macros an automatic placement is carried out by a tool so that the macros can be initially placed, then manual placement can be carried out to the satisfaction of the physical designer. A visual representation is shown in figure 4 where the macros have been automatically placed by the tool, this automatic placement is then updated manually by the physical designer as shown in figure 5.

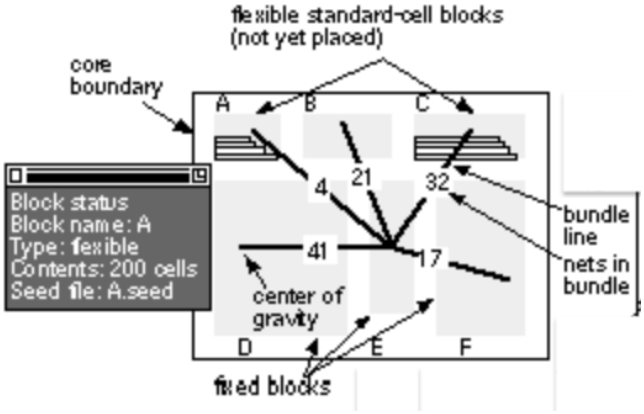

Fig. 4. Automatic Macro-Placement [5].

C. Clock Planning

The clock planning stage aims to decide how to provide clocks in a **symmetrically structured manner** to all clocked elements in the design [3]. In order the minimize the clock

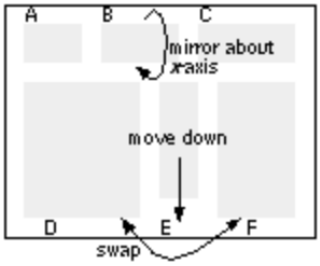

Fig. 5. Manually updated Macro-Placement [5].

skew between communicating elements due to their line resistance and capacitance and due to the fact the CTS (clock tree synthesis) may be insufficient for very high performance and synchronized ASICs [3]. Figure 6 shows a clock plan implementation example.

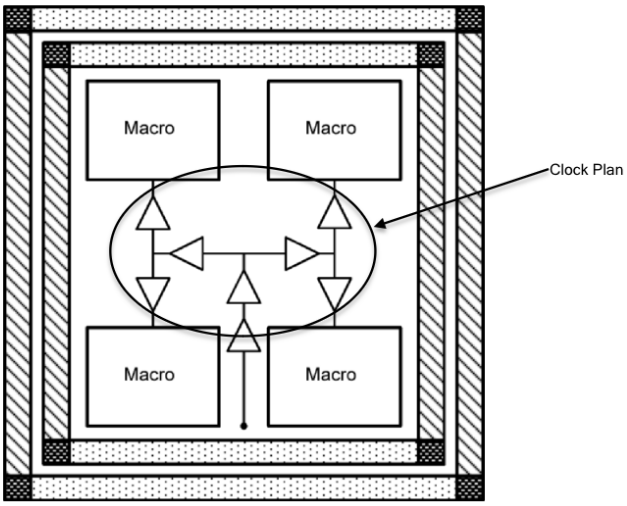

Fig. 6. Clock Plan [3].

III. Placement and Routing

Tools like Innovus and ICC2 are used for placement and routing. When you first create a floor plan, the cells are in a ”floating” state. This means they are just randomly placed in the ASIC core, without any specific location assigned. This can make routing the ASIC more difficult and may not meet timing requirements [4].

A. Global and Detailed Placement and Routing

1) Global Placement: To solve this issue one can partition the standard cell area and assign a group of cells to these partitions, or simply group a set of standard cells. The tools used in Global Placement are the “cluster” and “region” options [3]. In cluster, standard cells (i.e. components or blocks) are grouped and placed near each other, the cells that are usually clustered together are the time-critical cells. The only difference between cluster and region is that in region the location of the standard cell placement is predefined. In cluster, there is no specified location for where the cells will be placed. Regions can be soft or hard. A soft region is a physical constraint where a logical module is assigned to a location in the core and boundary of the region. This region can change during standard cell placement [3]. A hard region is more rigorous than a soft region and defines a physical partition for modular design. It has ”hard” boundaries that prevent standard cell crossing during placement. Hard regions are used primarily for timing-related issues, such as grouping clock, voltage, or threshold voltage domains [3].

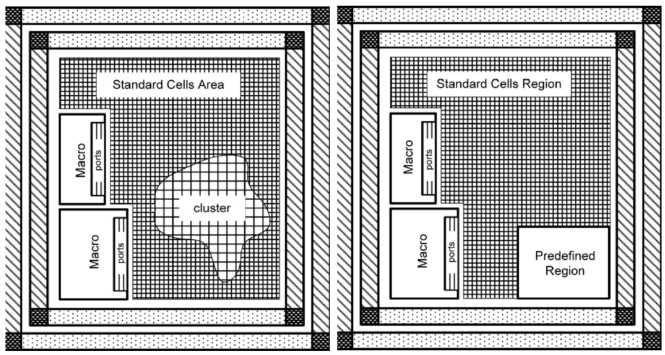

Fig. 7. Cluster and Region [3].

After clusters and regions have been defined, the global placement algorithm distributes standard cell instances uniformly across the available ASIC core and uses a method of estimation to minimize the number of wire lengths [3].

During Global Placement, the ASIC is divided into smaller pieces along the horizontal and vertical lines. This is done with help from mini-cut placement algorithms like **Quadrate**, **Slice and Bisection**, and **Bisection**. With Quadrate there is an equal number of instances on the ASIC core area, unlike slice and bisection which repeats the slicing and bisection procedure until all the standard cell instances have been assigned to rows, with the bisection algorithm the ASIC design is partitioned based on the standard cell rows as shown in figure 8.

2) Global Routing: Global routing is a way of telling the detailed routers which connections to make between different parts of the network. It tries to make sure that the network as a whole is organized in the most efficient way possible, and that the detailed routers can finish their work as quickly as possible [5].

3) Detailed Placement: After Global Placement, cell locations do not align with power rails, they do not have discreet coordinates and often overlap [7]. Detailed placement procedures such as; exchanging neighboring cells or making use of unused spaces improve the initial global placement, one of the main goals of detailed placement is to make the cells placed by Global placement easy to route. One of the best Detailed placement techniques/practices is the use of Tetris. The Tetris technique involves the sorting of cells by their x-coordinates and placing cells at the closest legal location

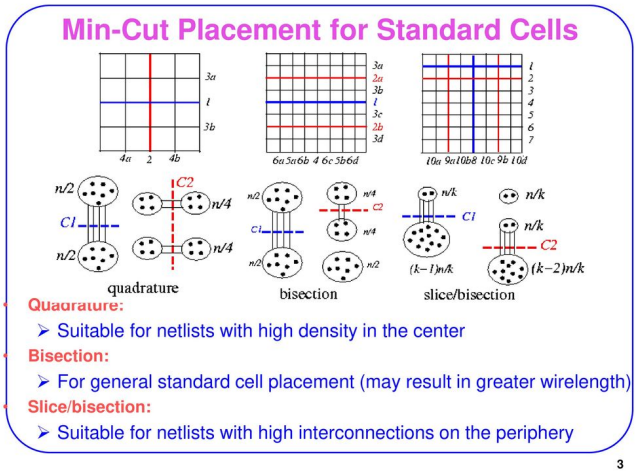

Fig. 8. Cluster and Region [6].

without exceeding the row capacity [7]. Sometimes, cells that are not adjacent can be rearranged so that their wires are shorter or three cells that are not connected by a net can be cycled [7]. Placement software tools like ECO-System carry out both Global and Detailed Placement during the Global Placement, software tools like FastPlace-DP on the other hand are only efficient for Detailed placement (i.e. after the Global Placement has been carried out).

4) Detailed Routing: The global routing step decides how many channels will be used for each interconnect, this information is used to decide the specific location and layers for each interconnects by the detailed layout [5].

B. Clock Tree Synthesis

The concept of clock tree synthesis (CTS) is the automatic insertion of buffers/inverters along the clock paths of an ASIC design to balance the clock delay to all clock inputs [3]. This is done after placement since it requires the precise physical location of cells and modules for clock propagation. This has an impact on dealing with accurate delay and operating frequency, and the clock is propagated before routing since clock routes are given higher priority than signal routes [3].

C. Power Analysis

After the clock tree synthesis, a detailed analysis of the power grids of the ASIC design must be performed to reduce the voltage (IR) drop risk in power, ground bounce on the ground nets, and electromigration (discussed in section II). To figure out how much power an ASIC will use we carry out Static timing analysis [3]. Ideally, power analysis involves knowing the operating frequency of each net and the corresponding load capacitance in the ASIC design to determine the dynamic power. However, this is almost impossible to do and involves a lot of computation [3]. A solution to this problem is to determine the static timing analysis instead, because "the static power analysis approach approximates the effect of dynamic switching on the power network using the fact that the decoupling capacitance between power and ground smooths out the maximum voltage drops or ground rises, and localized dynamic effects are not included." - - Golshan, K. (2007).

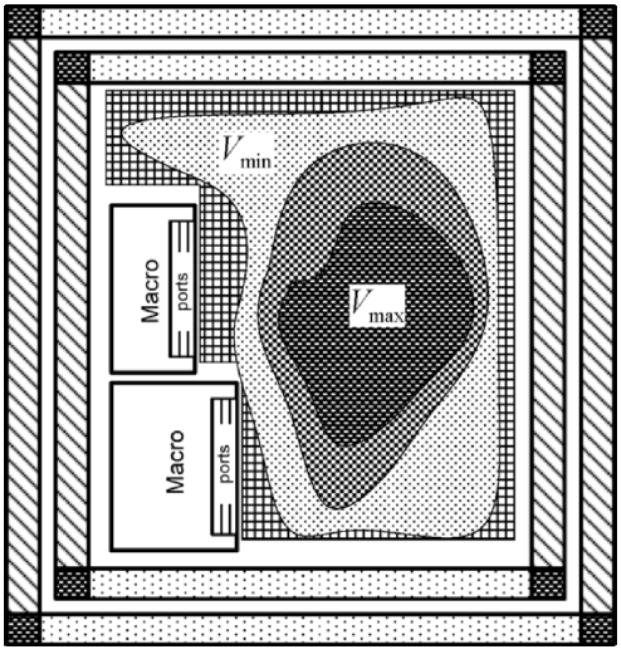

Fig. 9. Power Analysis Image View [3].

As shown in figure 9, For an ASIC to work correctly, the difference between the highest voltage drop (Vmax) and the lowest voltage drop (Vmin) plus external power supply voltage and the ground rise voltage must be smaller than the voltage that is defined in the standard cell library, the presence of inappropriate power or ground routing might make the voltage drop to be below the safe limit [3]. This can make it difficult to continue designing the ASIC, so it is important to fix the problem first.

IV. Verification

The verification stage of ASIC design is very important because at this stage we check to see if the ASIC can be manufactured and if it functions properly. The three major objectives of the verification stage are to check **the functionality** of the ASIC, **the timing** of the ASIC and the **the physical verification** (i.e. manufacturability) of the ASIC .

A. Functional Verification

We normally go with simulation-based verification for functional verification. But there are limits to simulation functional verification, such as the impracticability of simulation verification of complex ASICs and the long execution time [3]. A solution to this issue is to go with rule-based functional

verification. Two widely accepted techniques used for rule-based functional verification are assertion-based and formula-based functional verification. In assertion-based functional verification, conditional statements (assertion macros) are used to check the properties of the ASIC, if a condition statement is satisfied there is a checkmark that everything is fine else an error message is output with diagnostic information about the error [3]. Formal-based functional verification however checks the Real-Time-logic(RTL) against the gate level netlist [3].

B. Timing Verification

In timing verification, the ASIC is checked to see if it meets all the timing requirements. The timing of an ASIC is checked either by **dynamic Simulation** or **static timing analysis** [3].

1) Dynamic Simulation: For better accuracy dynamic simulation is carried out on the transistor level. However one should note that large-scale ASICs involve a lot of transistors (several million or billions), and it's impossible to do a dynamic simulation of them on a transistor level. But on a gate level, that's possible [3]. This is because it is impossible to provide vector sets that cover all functional behavior of several million or billions of transistors.

2) Static Timing Analysis: Static timing analysis is the preferred method for timing verification because of the elimination of the need for a vector set. And therefore can provide accurate timing analysis irrespective of the ASIC scale (i.e. the number of transistors).

C. Physical Verification

In this stage tools like the Cadence Physical Verification System (PVS) is used to check the Layout Versus Schematic (LVS), Design Rule Check (DRC), and Electrical Rule Check (ERC) of the ASIC against the semiconductor foundry's requirements to see if the ASIC can be manufactured and work properly.

V. Tape-out

The term "tapeout" is used in the context of ASIC design to refer to the process of preparing the final production data for the ASIC (i.e. the GDSII binary file) and sending it to the foundry for fabrication. The term "tapeout" derives from the practice of sending the production data on magnetic tape to the foundry, although today the data is typically sent electronically.

At this stage, the ASIC design is complete and binary files that contain information about the ASIC in hierarchical form (i.e. the Generic/Geometric Data Structure Information Interchange II (GDSII) file) with help of tools like the Quantum Extraction Solution can be extracted and sent to the semiconductor foundry for manufacturing.

GDSII is a file format used for storing geometric data for integrated circuits [9]. It was developed by Calma, a software company that was acquired by Cadence Design Systems in 1990.

GDSII files contain a hierarchical representation of the integrated circuit, it's geometric layout, layers and other features. Each layer is composed of a series of geometric shapes, such as lines, rectangles, and polygons, which are used to define the layout of the IC. The layout is stored in a set of coordinates relative to a reference point called the "database unit" [8].

GDSII files are commonly used to pattern layers of material onto a substrate during the fabrication process of ICs.

GDSII has been widely adopted in the microelectronics industry and is supported by many different design tools and fabrication facilities. It is a standard file format for storing and exchanging layout data in the semiconductor industry and is often used as the final output format for ASIC design data.

The standard binary file format for GDSII is the Calma GDSII stream format [8].

Once the GDSII file has been generated it is sent to a semiconductor foundry, where it is typically used to create photomasks, with the aid of a tool the photomask is used to pattern layers of material onto a round thin slice of a semiconductor material as shown in figure 10.

The thin slice of semiconductor material (i.e. silicon) is known as a substrate or wafer, the resulting patterns creates a cluster of the actual rectangular physical device, known as a die or integrated circuit as shown in figure 11.

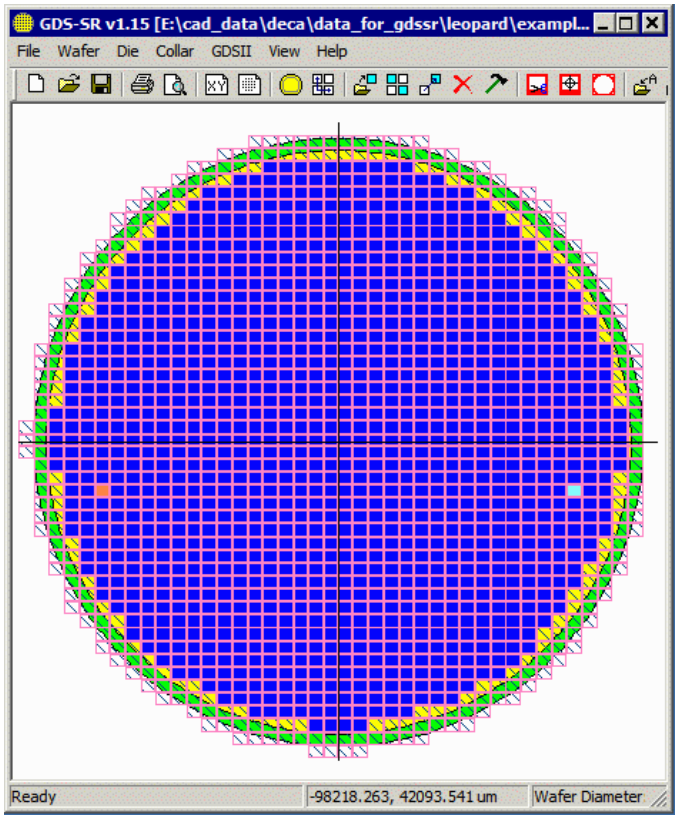

Fig. 10. Power Analysis Image View [10].

There are many different tools that can be used to create photomasks from GDSII files. Some examples of these tools include:

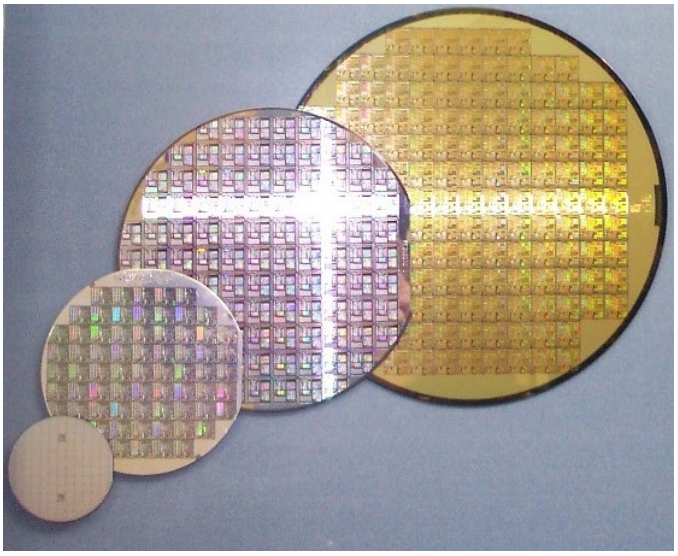

Fig. 11. Power Analysis Image View [10].

Calibre: This is a suite of software tools developed by Mentor Graphics that is used to design, verify, and manufacture microelectronic devices. Calibre includes tools for creating photomasks from GDSII files, as well as tools for simulating and analyzing the performance of devices.

IC Mask Design: This is a software tool developed by Cadence Design Systems that is used to design and create photomasks for microelectronic devices. It includes support for importing and manipulating GDSII files, as well as tools for layout editing, verification, and mask making.

Laker: This is a software tool developed by Silicon Labs that is used to design and create photomasks for microelectronic devices. It includes support for importing and manipulating GDSII files, as well as tools for layout editing, verification, and mask making.

Photomask Design: This is a software tool developed by Synopsys that is used to design and create photomasks for microelectronic devices. It includes support for importing and manipulating GDSII files, as well as tools for layout editing, verification, and mask making.

Conclusion

As ASICs get more complex, it's important to have the right tools for designing and implementing the backend for the ASIC.

There are a variety of ASIC (Application Specific Integrated Circuit) back-end design tools available, each with its own set of features and capabilities. Some popular ASIC back-end design tools include:

A. Synopsys Tools

1) Synopsys Design Compiler: This is a popular physical synthesis tool that is used to optimize the design of an ASIC for performance, area, and power.

2) Synopsys PrimeTime: This is a static timing analysis tool that is used to analyze the timing performance of an ASIC design. It can help to identify timing violations and suggest design changes to resolve them.

B. Cadence Tools

1) Cadence Innovus: This is a physical implementation tool that is used to create the physical layout of an ASIC, including the placement and routing of the various components of the design.

2) Cadence Tempus: This is a signoff-quality static timing analysis tool that is used to analyze the timing performance of an ASIC design. It can help to identify timing violations and suggest design changes to resolve them.

C. Siemens Tool

1) Mentor Graphics Calibre: This is a verification and signoff tool by that is used to verify the functionality and performance of an ASIC design. It includes a variety of verification and analysis capabilities, including static timing analysis, power analysis, and electromigration analysis.

Fig. 12. Back-end design tool. [14]

The most widely used tool is Synopsys.

References

[1] TARAATE, V.A.I.B.B.H.A.V. (2021) ASIC design and synthesis: RTL design using Verilog. S.l.: SPRINGER.

[2] Taraate, V. (2019) Advanced HDL synthesis and soc prototyping: RTL Design using Verilog. Singapore: Springer.

[3] Golshan, K. (2007) Physical Design Essentials an ASIC design implementation perspective. Boston, MA: Springer Science+Business Media, LLC.

[4] VLSI Physical Design Methodology for ASIC development with a flavor of IP hardening (no date) Design And Reuse. Available at: https://www.design-reuse.com/articles/48852/vlsi-physical-design-methodology-for-asic-development-with-a-flavor-of-ip-hardening.html (Accessed: December 1, 2022).

[5] Sebastian, S.M.J. (1997) Application-specific integrated circuits by Michael John Sebastian Smith. London: Addison Wesley.

[6] Allison, L. (no date) Min-cut placement. - ppt download, SlidePlayer. Available at: https://slideplayer.com/slide/13134218/ (Accessed: December 14, 2022).
[7] Kahng, A.B., Lienig, J., Markov, I.L., Hu, J. (2011). Global and Detailed Placement. In: VLSI Physical Design: From Graph Partitioning to Timing Closure. Springer, Dordrecht. https://doi.org/10.1007/978-90-481-9591-64
[8] Singla, Aayush Lippmann, Bernhard Graeb, Helmut. (2019). Verification of Physical Chip Layouts Using GDSII Design Data. 55-60. 10.1109/IVSW.2019.8854432.
[9] Calma :: GDS II stream format manual 6.0 feb87 : Free Download, borrow, and streaming (1987) Internet Archive. Available at: https://archive.org/details/bitsaverscalmaGDSII.0Feb871367585 (Accessed: December 22, 2022).
[10] DiBartolomeo, S. (no date) GDS-SR - GDSII STEP and repeat for wafer level masks. Available at: https://www.artwork.com/gdsii/gdssr/index.htm (Accessed: December 12, 2022).
[11] Wafer (electronics) (no date) Wikiwand. Available at: https://www.wikiwand.com/en/Wafer(electronics) (Accessed: December 12, 2022).
[12] EDA tools, semiconductor IP and Application Security Solutions (no date) Synopsys. Available at: https://www.synopsys.com/ (Accessed: December 22, 2022).
[13] Cadence Design Systems (no date). Available at: https://www.cadence.com/enUS/home.html (Accessed: December 22, 2022).
[14] Calibre Design Solutions (no date) Siemens Digital Industries Software. Available at: https://eda.sw.siemens.com/en-US/ic/calibre-design/ (Accessed: December 22, 2022).