International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

VLSI Design Analysis with Commercial EDA Tools - From Logic Synthesis to Physical Design

Pranoti Wanjari

Physical Design Engineer, Cyient, Hyderabad, India
Email:*psalankar3[at]gmail.com, shaktiwanjari. pranoti[at]cyient. com*

Abstract: ***The VLSI (Very Large Scale Integration) industry has been growing constantly obeying Moore's law to an extent where multiple processors can be implemented on a single chip. The VLSI trend is getting complex day by day, especially the complexity of IC technology. It is very important to have better design approach. Physical design is the process of converting a circuit description at Register Transfer Level into the physical layout. This paper focuses on analyzing the process flow from logic synthesis to physical design by Using commercial electronic design automation (EDA) tools.***

Keywords: VLSI, CAD, EDA, IC design, Logic synthesis, Physical design

1. Introduction

Very large-scale integration (VLSI) is a process which allows millions of transistors to be embedded on a single silicon semiconductor chip. The VLSI industry has now entered into nanometer technology by shrinking the size of transistors which enables to pack more and more number of transistors in the same die area. To create a chip, its physical layout is required. The process to acquire that layout is known as VLSI design. The designers may fully or partially utilize the EDA software, also known as computer-aided design (CAD) tools throughout the VLSI design flow. Basically, the idea of a system is first recorded along with its specifications such as speed, power and area in a formal language, then the register-transfer level (RTL) function is written using Hardware Description Language (HDL). Furthermore, the abstract RTL description is interpreted into gate-level or logic level through logic design. Then, the optimized gate-level net list can be implemented at transistor or circuit level through circuit design, normally the process is fully automated by the EDA. At this point, the implementation is at the lowest level of abstraction. Finally, the physical layout which concerning the chip's geometry aspect can be generated through physical design. The optimized gate-level netlist can also be converted into layout in physical domain which is known as module layout through physical design. The file format for the layout can be recorded as Graphical Database System (GDS) II Format. It is a binary file consisting of references of cells and geometry.

2. Physical Design Flow

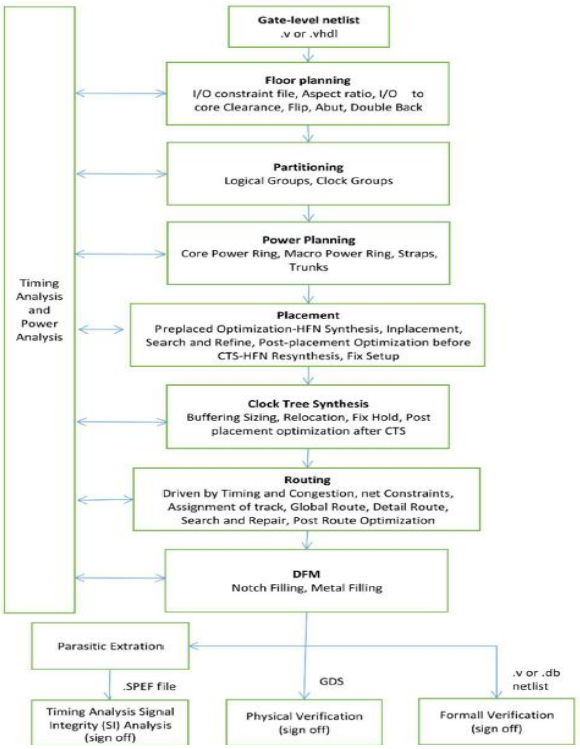

Figure 1: Physical design flow

A. Logic Synthesis

Synthesis is the implementation of hardware to produce desired output from the given input. Logic synthesis aims to meet the constraints on power, area, and speed. Therefore, optimization challenge is to minimize logic gates' power dissipation, the overall design chip area, as well as delay on the critical path. Additionally, logic synthesis also aimed to produce designs with a good degree of testability for verification of the fabricated chip.

B. Physical Design

The design process to transform the gate-level netlist obtained from logic synthesis into geometrical layout is known as physical design. The physical layout includes information such as standard cells, clock trees, power nets, as well as placement and routing. Naturally, the files used in logic synthesis such as constraint file and standard cell library files are required for the physical design implementation. The flow of physical design can be seen in Fig.1, adapted from a VLSI blog. In physical design, the design components such as macro cells, standard cells, logic gates and transistors are instantiated with the corresponding geometrical representations. For instance, the components with fixed shapes and sizes per fabrication layer are assigned with spatial locations and their connections are completely routed in the metal layers. It is known that physical design has direct impacts on circuit's performance, reliability, power, area, as well as manufacturing yield.

I. Floor Planning

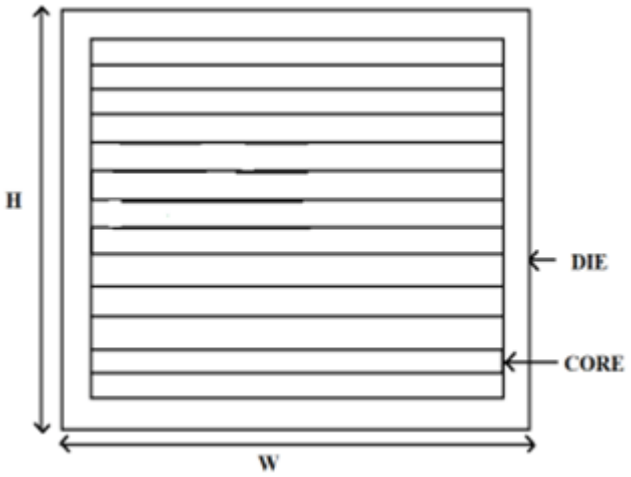

Figure 2: Height and Width of Core and Die

The floor planning stage is the most important flow in physical design. The floor planning involves various steps. Define height and width of Core and Die: The height and width of Core and Die mainly depends on utilization factor and aspect ratio as shown in Fig.1.

The utilization factor of the design defines congestion and cell density of functional block.

Utilization Factor = Area Occupied by Netlist/Total Area of Core and
Aspect Ratio = Height/Width.

Define location of Preplaced Cells: The preplaced cell can be defined as functionality of netlist and cell that are encapsulated in a block. The preplaced cell can be ICG, macros and IP's etc.

Surround preplaced cell with decoupling capacitor: The decoupling capacitor acts as a power supply, whenever the preplaced cells need instant supply. The preplaced cell or the sensitive cell are surrounded by the decoupling capacitor to drive the needs of supply. This stage is optional, this depends on the block of current design and power rails distance to those preplaced cell or sensitive cell.

- Power Planning: The mesh kind of power supply is being implemented instead of single source of power supply. This kind of approaches will avoid voltage droop and ground bounce.
- Pin/Port Placement: Any circuit will have input and output pins in order to interact with different modules or blocks. The static timing analysis is dependent on the pins that are to be located close to the logic cell or module in which it has a functional connectivity in order to meet the timing requirement of pin placement.
- Logical Cell Placement Blockage: The blockage can be placement blockage and routing blockage. The placement blockage which is placed in between core and die which ensures that there is no logic cells accidently being placed in that area. This area is dedicated to ports or the pin which is functionally connected to the block.

II. Floor plan Aware Synthesis

The synthesis mainly involves translation, optimization and mapping of the functional block as per specifications extracted from top level module. The logic gates have some shape that represents their functionality, but in practical those shape does not exist, all the cell will be in square shape box of same standard cell height. The logical connection binds the net list with physical cell.

III. Placement

The block of module need to be placed in the optimized way such that the wire length should be of minimum length is as shown in Fig.3. The module consists of combinational, sequential and preplaced block. The port/pin present are of both data and clock. In this stage, the tool estimates wire length and signal degradation is analyzed, and based on this degradation a repeater is inserted. The repeaters are signal conditioners to improve the slew rate. For example, the signal is taken from port Din2, condition it, reconstruct the original signal and send it to next stage. This a scenario was port Din1 is close to flip flop FF1 (orange) the wire length is very less hence no need of repeaters as shown in Fig.4.

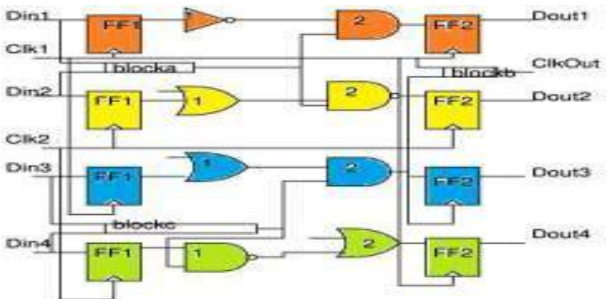

Figure 3: Functional Block

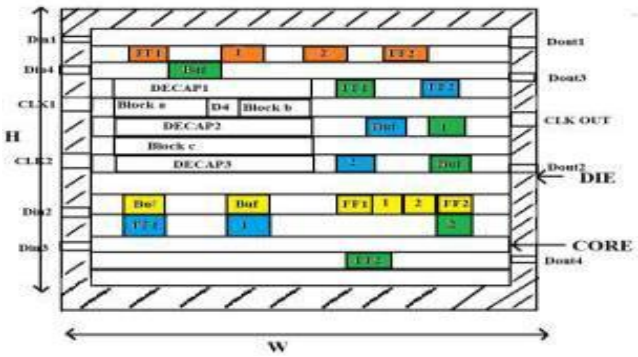

Figure 4: Placement

IV. Pre-timing Analysis

With the optimized placement from the previous stages, the ideal timing analysis is carried out and QoR report is generated. The idea of getting timing check at this stage is to ensure that floor planning of preplaced cells in previous stage are placed legally and to check whether the necessary timing is met. If not, floor plan can be changed in early stage, so that there is no surprise at the end. This process involves (1) Setup and Hold Check and (2) Data Slew Check

V. Clock Tree Synthesis

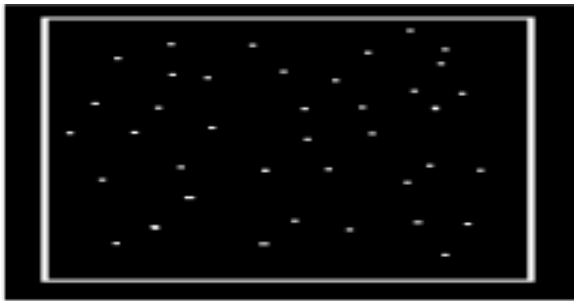

Figure 5: CTS Clock End Points

The clock is used on chip to plan various functionality of block. In a chip, the flip-flops are spread all over; the purpose of clock tree is to connect from the clock source to the clock end points as shown in Fig.5. The Clock Tree Synthesis motive is not only to drive the flip-flops by the clock but also need to meet few quality checks like (1) Skew, (2) Pulse Width, (3) Duty Cycle, (4) Latency, (5) Clock Tree Power and (6) Signal Integrity and Crosstalk.

VI. Timing Analysis with Real Clock

Timing analysis with real clock as the name indicates, here the clock paths will be present though the data paths nets are not created. Based on this path timing analysis is carried out.

VII. Route

This stage routes the design using actual wire, now it is no more estimated wire. The real wire is routed based on connectivity of netlist and maintain the existing timing scenario. The timing analysis in Clock Tree Synthesis stage should not deteriorate as in line with router engine working.

VIII. Design Rule Check (DRC)

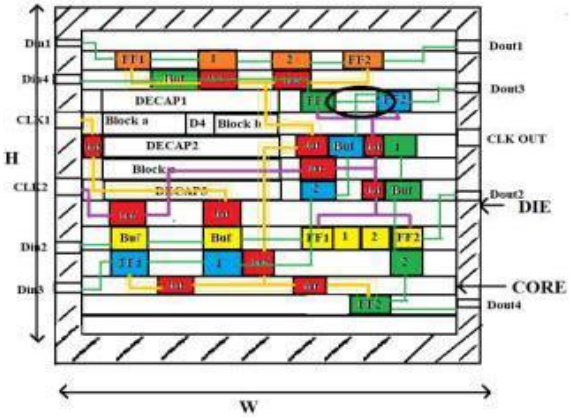

Figure 6: Design Rule Check

Routing at higher density might cause some spacing and shorting issues which is encircled as shown in Fig.6. These two routes are much closer to each other, which may lead to coupling or might not be able to fabricate by a foundry. Finally, the design goes to foundry for fabrication and they have their own guidelines set.

For example, the minimum spacing between two wires should be 10-micro meters but practically if the minimum spacing between two wires is 5-micron meters then the flag is raised as DRC error.

IX. Chip Finishing

Before VLSI chip tapes out, there is an important preparation step known as chip finishing. One of the tasks in chip finishing is filler cell insertion. In reality, it is impossible to fill up the standard cell rows with regular cells. For the purpose of fabrication, filler cell insertion is needed to add in filler cells to fill in the empty rows between the standard cell. This process ensures the N-well and P-well continuity as well as the power buses continuity for the fabrication process. For the reason to meet the density requirements specified by the foundry, extra metals and polys fill layers need to be presented to the areas that is not routed.

X. Parasitic Extraction

The parasitic extraction explores the exact resistance and capacitance value there is no estimation of resistance and capacitance. The Standard Parasitic Exchange Format (SPEF) specifies the SI units of resistance, capacitance, power and various other standard parameters called as SPEF header. The IC design has to go through physical verification to verify its manufacturability and electrical connectivity. Physical verification can be separated into three several steps, namely electrical rule check (ERC), design rule check (DRC), antenna check, layout versus schematic (LVS) check, and parasitic extraction. Before the final layout is sent for fabrication, it must be checked properly through these verification steps.

3. Results and Discussion

The EDA tools used for logic synthesis is Synopsys Design Compiler (DC). The EDA tools used for physical design is Synopsys IC Compiler (ICC). The details of the practical

implementation in this paper will be discussed in this section.

A. Logic Synthesis with Synopsys Design Compiler

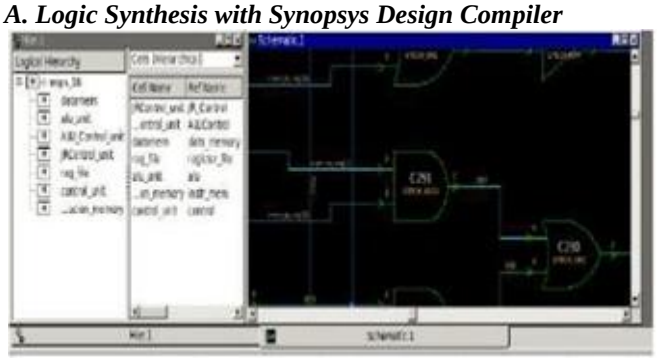

Figure 7: Part of the schematic in technology-independent design (GTECH) format

In the DC environment, the command 'read_file-format-rtl {$file_list}' is used to read multiple design files into the memory of DC. The RTL design files are written in Verilog HDL. Then, 'analyze-format verilog {$file_list}' is used to check the design for errors. Next, 'elaborate' translates the design into a technology-independent design (GTECH) from the intermediate files produced during analysis. Also, the design references are resolved by performing 'link' command. If there's any unresolved references in the netlist, it means the design data for the module is not detected and cannot be used in processing. To resolve the reference, it has to be made sure that the design name can be found in the DC memory, or that the library cell used is from the link library.

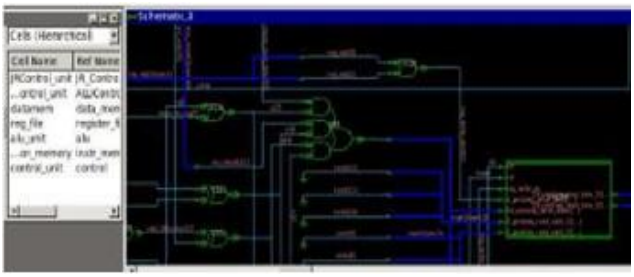

Figure 8: Part of the schematic after mapping and optimization

```
module control ( opcode, reset, reg_dst, mem_to_reg, alu_op, jump, branch,
       mem_read, mem_write, alu_src, reg_write, sign_or_zero );
  input [2:0] opcode;
  output [1:0] reg_dst;
  output [1:0] mem_to_reg;
  output [1:0] alu_op;
  input reset;
  output jump, branch, mem_read, mem_write, alu_src, reg_write, sign_or_zero;
  wire   \mem_to_reg[0] , \alu_op[1] , n7, n8, n9, n10, n11, n12, n13, n1, n2,
         \reg_dst[1] , n4, n6;
  assign mem_read = \mem_to_reg[0] ;
  assign mem_to_reg[0] = \mem_to_reg[0] ;
  assign alu_src = \alu_op[1] ;
  assign alu_op[1] = \alu_op[1] ;
  assign mem_to_reg[1] = \reg_dst[1] ;
  assign reg_dst[1] = \reg_dst[1] ;

  or03d1 U4 ( .A1(opcode[2]), .A2(reset), .A3(opcode[1]), .Z(n8) );
  an02d1 U5 ( .A1(n2), .A2(n7), .Z(reg_dst[0]) );
  an02d1 U6 ( .A1(n6), .A2(n11), .Z(jump) );
  nd04d0 U14 ( .A1(n8), .A2(n9), .A3(n1), .A4(n10), .ZN(reg_write) );
  nd03d0 U15 ( .A1(opcode[0]), .A2(opcode[1]), .A3(alu_op[0]), .ZN(n9) );
  nr03d0 U16 ( .A1(opcode[2]), .A2(reset), .A3(n11), .ZN(n7) );
```

Figure 9: Portion of synthesized gate-level netlist after the design was constrained, mapped, and optimized.

B. Physical Design with Synopsys IC Compiler

Next, the mapping and optimization of the design is performed using *compile* command. The output of this command is a gate – level implementation or netlist. During mapping process, DC identify the logic of the design and assign the suitable logic cells available from the technology library to those logic. After mapping, there might be some design violations and therefore optimization was required. The optimization process honored the DRCs and optimization constraints provided by the designer and it attempted to create an optimized gate-level netlist in many iterations. By default, DC attempted to create the smallest possible design while meeting the timing specification.

DC will refer to the technology library to calculate the delay through each of the points as shown in the 'Incr' column in the timing report. The Static Timing Analysis (STA) algorithm is used to generate timing report, as shown in Fig.10. With all the delays, the data arrival time can be calculated and used for setup timing analysis. DC will also calculate the data required time for the signal to safely travel to the endpoint. In this case, the data required time is the clock period minus the clock network delay, clock uncertainty period, and the library setup time. Since the data arrival time did not exceed the data required time, it is said to have a positive slack. In case of negative slack, there will be a setup timing violation as the arriving signal might cause unexpected and undesired behaviors. Generally, hold time analysis is not the concern during synthesis process as it will normally appear only after CTS.

```
Startpoint: pc_current_reg[1]
            (rising edge-triggered flip-flop clocked by clk)
Endpoint: pc_current_reg[15]
            (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Des/Clust/Port      Wire Load Model       Library
------------------------------------------------
mips_16             70000                 cb13fs120_tsmc_max

Point                                     Incr       Path
------------------------------------------------------------
clock clk (rise edge)                     0.00       0.00
clock network delay (ideal)               0.00       0.00
pc_current_reg[1]/CP (dfcrq1)             0.00 #     0.00 r
pc_current_reg[1]/Q (dfcrq1)              0.38       0.38 r
U177/Z (an02d0)                           0.15       0.53 r
U175/Z (an02d0)                           0.19       0.72 r
U173/Z (an02d0)                           0.19       0.90 r
U171/Z (an02d0)                           0.19       1.09 r
U169/Z (an02d0)                           0.19       1.28 r
U167/Z (an02d0)                           0.19       1.46 r
U165/Z (an02d0)                           0.19       1.65 r
U163/Z (an02d0)                           0.19       1.83 r
U161/Z (an02d0)                           0.19       2.02 r
U159/Z (an02d0)                           0.19       2.20 r
U157/Z (an02d0)                           0.19       2.39 r
U155/Z (an02d0)                           0.19       2.57 r
U153/Z (an02d0)                           0.15       2.72 r
U152/Z (xr02d1)                           0.28       3.00 f
U103/ZN (wo122d1)                         0.29       3.29 r
U125/ZN (nd02d1)                          0.06       3.35 f
pc_current_reg[15]/D (dfcrq1)             0.00       3.35 f
data arrival time                                    3.35

clock clk (rise edge)                     5.00       5.00
clock network delay (ideal)               0.00       5.00
pc_current_reg[15]/CP (dfcrq1)            0.00       5.00 r
library setup time                       -0.07       4.93
data required time                                   4.93

data required time                                   4.93
data arrival time                                   -3.35
```

Figure 10: Timing report generated with STA algorithm in DC

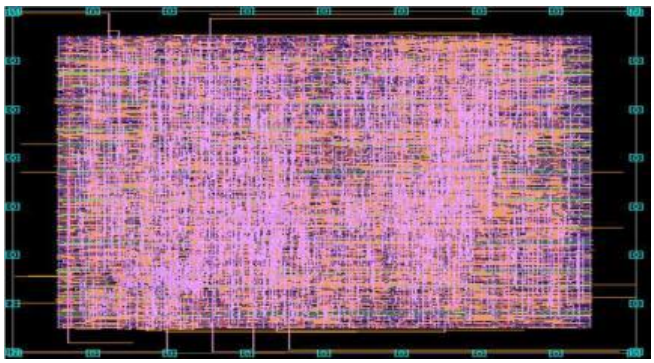

Figure 11: Layout generated after placement, CTS and routing

```
Cell Count
----------------------------------
Hierarchical Cell Count:          8
Hierarchical Port Count:        339
Leaf Cell Count:               7141
Buf/Inv Cell Count:             373
Buf Cell Count:                 301
Inv Cell Count:                  72
CT Buf/Inv Cell Count:           71
Combinational Cell Count:      2901
Sequential Cell Count:         4240
Macro Count:                      0
----------------------------------


Area
----------------------------------
Combinational Area:     7749.500000
Noncombinational Area: 30837.000000
Buf/Inv Area:            532.000000
Total Buffer Area:           477.25
Total Inverter Area:          54.75
Macro/Black Box Area:      0.000000
Net Area:              10673.329438
Net XLength        :      270495.47
Net YLength        :      440865.44
----------------------------------
Cell Area:             38586.500000
Design Area:           49259.829438
Net Length         :      711360.88
```

Figure 12: QoR report

The verilog netlist generated after logic synthesis is used in physical design using ICC. After the initial floor plan was generated, virtual flat placement was quickly done to check the congestion and timing. It is good to make sure there is no congestion and timing issue during floor planning so that at the later stage the design can be cleaner. Generally, the maximum IR drop across the core area should not be exceeding 10 % of the supply voltage. Before proceeding to the placement stage, the design was checked again to make sure there is no violation caused by the design planning stage. An actual placement of standard cells is then performed after checking. The placement process can also fix congestion issues. Prior to performing CTS, the clock target skew must be set to certain realistic value and the clock uncertainty has to be set as well to account for clock jitter. Basically, the hold-time violation after CTS is fixed by adding buffers which also slightly increased the design area. To minimize the design area, an area optimization was specified during an incremental clock optimization. This feature will perform down-sizing of the buffers which impacts the timing paths negatively. To prevent timing degradation during routing stage causing setup violations, a timing margin of 5 % of the clock period is normally adopted. Fig.11 shows a sample of the layout at the core area after placement, clock tree synthesis and routing.

After performing routing, the chip finishing task involves standard filler cells insertion in the core area to ensure P/G continuity and continuity on n-wells and p-wells in each row of standard cells. Lastly, metal fill is inserted so that the design meets metal density rules. This process is done last because it used up most of the remaining routing resources. In the final stage, the quality of the finished design was checked to make sure there is no degradation. Specifically, the design is checked so that it does not violate any design constraints (area, timing and power). Finally, the finished design has to pass ERC, DRC, antenna check, LVS before streaming out to GDSII file format. Fig.12 depicts the report generated for QoR (Quality of Results). As can be seen in the report, the total design area is 49259.8 μm2. Combinational cell count and sequential cell count are 2901 and 4240, respectively. Total dynamic power is about 12 mW.

4. Conclusion

The VLSI design flow adopted in the IC design industry by using commercial EDA tools. Front-end design included the technology mapping and optimization of the RTL to produce a gate-level netlist inverilog format. The verilog netlist was then used for back – end design to produce the physical layout in GDSII format. Backend design included floor planning, placement, clock tree synthesis, routing, and chip finishing. The design and verification were performed by the EDA tools. The design automation focused on optimizing the area, power, and timing of the design while honoring the design rules constraints. At every stage of design, the quality of the result, including timing, power consumption, and area were monitored.

References

[1] Siu Hong Loh*, Ing Ming Tan, Jia Jia Sim *VLSI Design Course with Commercial EDA Tools to Meet Industry Demand – From Logic Synthesis to Physical Design*, 2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2021), 27–28 August 2021, Penang, Malaysia

[2] VLSI Physical Design. “Physical Design Flow. ” vlsijunction. com http: //www.vlsijunction. com/2015/08/physicaldesign- flow. html (accessed April 21, 2021).2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2021), 27–28 August 2021, Penang, Malaysia 60

[3] Ganesh Prasad, Sudha R Karbari, SajeeshAmmikkallingal, Sukanya Kamath Bellal “*Analysis, Physical Design and Power Optimization of Design Block at Lower Technology Node”, 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018), MAY 18th & 19th 2018*

[4] S. M. Kang, Y. Leblebici, C. Kim, *CMOS Digital Integrated Circuits: Analysis & Design.4*th Ed., Singapore: McGraw-Hill, 2015.

[5] D. Das, *VLSI Design*. New Delhi: Oxford University Press, 2010.

[6] L. Xiu, *VLSI Circuit Design Methodology Demystified: A Conceptual Taxonomy*. Hoboken, New Jersey: John Wiley & Sons, 2007.

[7] P. Corsonello, S. Perri and G. Cororullo, "Area-time-power tradeoff in cellular arrays VLSI implementations, " in *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol.8, no.5, pp.614-624, Oct.2000.

Online:

[8] www.udemy. com/vlsi-academy-physical-design-flow/learn/v4/overview

[9] Synopsys Design Compiler® User Guide and IC Compiler II User Guide