iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XII **Month of publication:** December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76324

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

An Optimal Synthesis Approach for Achieving High Frequency with Leakage Optimization

Amisha Jha[1], Mukul Anand[2]
MDG-GPM, STMicroelectronics, Greater Noida, India

Abstract: ***The traditional purpose of synthesis aims at timing closure, but in modern design flows, tools often require multiple iterative adjustments to achieve optimal results due to the complexity of designs with multimillions of placeable objects. Logic synthesis, the process of generating optimized logic-level representations from high-level descriptions, plays a crucial role in high-performance microprocessors and microcontrollers design methodologies, especially with the rapid advances in integrated circuit technology and increasing design complexity, with a focus on achieving timing convergence.***

This paper discusses the challenges and improvements (methodologies) associated with synthesis and placement in the context of modern chip design. The experimental results presented in this paper suggest an approach that creates an efficient design flow, eliminating placement and synthesis iterations, leading to timing improvements.

This paper discusses the different approaches that have been used to achieve an operating frequency of 1.2Ghz with minimal power & area for a CPU Sub-system. Logical Synthesis has been carried out using Synopsys Design Compiler (DC) while the PnR implementation tool is Cadence Innovus. Using different strategies and automations, the complete synthesis to placement cycle has been reduced leading to a more robust flow for best PPA. This approach includes Register cloning, Data path optimization, Pipelining and Placement aware synthesis approach.

Keywords: Microcontroller, synthesis, strategy, DC

I. INTRODUCTION

This paper discusses the challenges and complexities associated with logical and physical synthesis in the design of microcontrollers/microprocessors. Physical synthesis is characterized as a runtime-intensive and iterative process that needs the collaboration of various strategies and algorithms. The main challenge is apprehended by the increase in design sizes from millions to potentially tens of millions of placeable objects.

This paper discusses the various ideas that help in the direction of timing closure that are accountable for both the run time and the quality of results (QoR).

The ideas (strategies) presented in this paper are oriented to achieve faster timing convergence and improved quality of results.

The presented logical and physical synthesis compile strategies focus on the effectiveness of the approach in achieving improved **timing** performance.

A. Overview of Strategies/Methodologies

Synthesis operates simultaneously in logic, timing, placement, and routing domains, replacing iterative processes for manufacturable designs. In the context of synthesis, the traditional goal is to achieve timing closure by creating a placed design that meets specified timing requirements, while satisfying various constraints. However, the synthesis tools often require multiple iterations by the design team to improve solutions, impacting design productivity.

The integrated logic/physical synthesis flow begins with traditional logic synthesis steps like logic restructuring and technology mapping. The technology-mapped design then undergoes global placement, aiming to minimize total wire length while addressing objectives such as congestion mitigation. Following this, physical location optimization focuses on timing-critical regions, assuming the presence of ideal buffer trees.

The paper presented discusses the approach taken to enhance the efficiency of the synthesis process in microprocessors and microcontroller design. However, the paper also emphasizes the need for estimating and optimizing physical design effects at earlier stages in the design flow at earlier. This suggests a recognition of evolving challenges and the importance of continuous exploration of more advanced strategies to enhance the overall efficiency and quality of microprocessors and microcontroller designs.

B. Reason

There has always been a prerequisite to achieve timing closure with Zero Wire-Load model (ZWL). The ZWL model serves as the starting point in the design process, meeting the timing specifications at this stage becomes significantly challenging and if the timing requirements are not achieved at this stage, it becomes unlikely to achieve timing at the (PnR) physical design implementation stage. This ZWL model is extremely optimistic because of the significant increase in the wire delays in modern chip designs, so before proceeding to perform physical synthesis, the designers are highly recommended to follow a more pessimistic approach which assures that meeting the timing requirements is robust, accounting for significant delays that are ignored in optimistic approach.

Design Details have been Captured as below.

Design	Details
Config	ARM Cortex CPU
Process	TSMC 12FFC
Libraries/Memories	Arm IPTSMC 12 FFC SVT/LVT/ULVT
PVT Corners	Setup: RC Max: ssgnp/0.81v/125c Setup & Power: TT/0.99v/125c Hold: RC Min: ffgnp/0.99v/n40c
Target Operating frequency	1.5 GHz

II. DIFFERENT COMPILE STRATEGIES

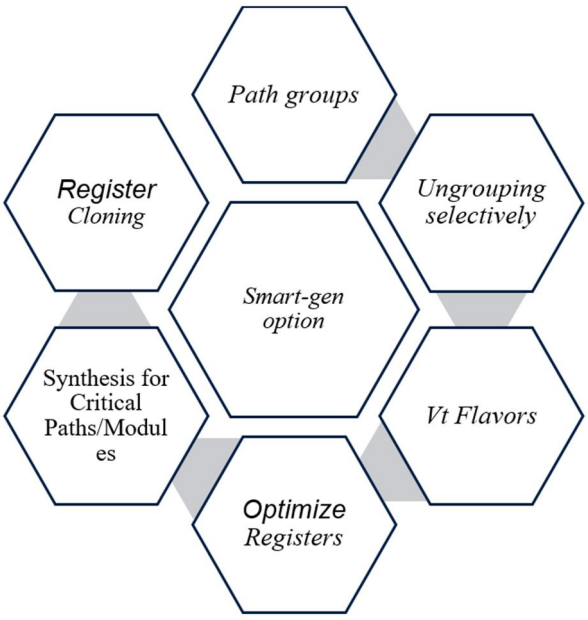


Fig.1. Different Compile Strategies

A. Path Groups

Path groups play a crucial role in the synthesis process within the VLSI design flow. Path groups are used to identify and analyze specific paths or groups of paths within a design, mainly focusing on the critical paths for timing analysis and optimization. The critical path is the longest path in the circuit, and it's timing directly affects the overall performance of the design. Path groups significantly identifies and analyzes critical paths in the design. Path groups help in efficiently managing and optimizing the timing characteristics of specific paths, ensuring that the design meets the required timing constraints. Path groups definition in the synthesis process offers a targeted approach to critical path analysis, timing closure, incremental optimization and debugging. They play a crucial role in ensuring that the synthesized design meets the desired performance.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

```
create_auto_path_groups -mode mapped -exclude IO
proc_auto_weights -wns -nvp
```

B. Smart-gen Option

The physical design tools offer various options and settings to control the synthesis process, optimize the design for performance, area, and power, and ensure that the final netlist meets the specified design constraints. This command helps in the Datapath synthesis of the design, controls the strategies used when generating the data path cell for arithmetic and shift operators. The timing path in design consists of a lot of adders/multipliers and other arithmetic operators needed for data manipulation. The logical depth in such designs can be a nightmare for achieving desired frequency. To reduce logic depth in critical paths we used, below commands:

-optimize for speed - Supplies a specific optimization goal for the specified cells. This option overrides the default optimization goals for the specified cell or cells.

When the smart generation options are specified without an object list, the specified options are automatically applied to the entire design through all levels of hierarchy regardless of the use of -hierarchy option.

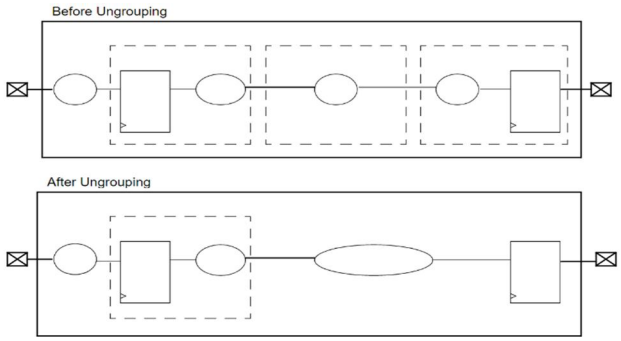
```
set_dp_smartgen_options -optimize_for speed
compile_ultra -no_autoungroup -
no_seq_output_inversion -gate_clock -scan
```

C. Ungrouping

Ungrouping refers to the process of breaking apart or separating previously grouped elements or structures within the design.

Ungrouping is performed when certain analysis and optimizations need to be applied selectively to specific portions of the design hierarchy. Ungrouping allows the synthesis tool to apply optimizations more precisely to individual elements or small groups, likely resulting in better performance.

Designers may select to ungroup elements strategically to achieve desired results in terms of (PPA) power, performance, and area efficiency.



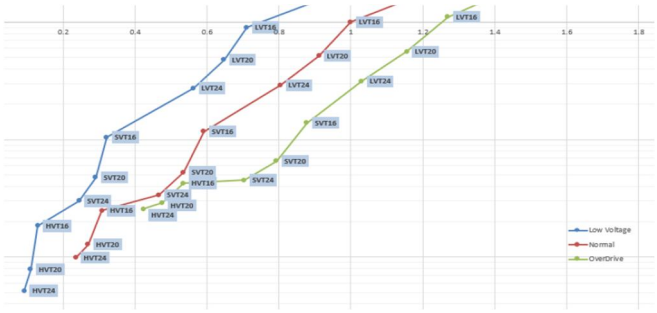
D. Vt flavors

The selection and utilization of different Vt flavors is one of the most important strategies used in the optimization process in the synthesis stage. With the usage of different Vt flavors, designers can balance power consumption and performance based on the timing requirements of different sections of the design.

LVT cells (High-performance Vt flavor) that are often used in the critical paths where speed/performance is essential.

HVT cells (Low-power Vt flavor) or SVT cells (Medium-performance Vt flavor) may be used in the non-critical paths to optimize power without affecting the overall design performance.

Different Vt flavors also help in optimizing the physical area occupied by the design. Different VT flavors were used based on Low Leakage and High Performance. Below Chart show the Leakage Vs the Performance for the different VT Flavors and Cells chosen was based on this :



E. Optimize Registers

Optimize registers option used in the synthesis process typically refers to the optimization techniques applied to registers within the design to improve (PPA)timing, power, or area.

Synthesis tool performs various techniques such as logic restructuring and technology mapping, including register balancing and retiming, to achieve better timing closure.

The optimization of registers often helps in adjusting the clock-to-q delays and setup/hold times to meet timing requirements.

The register optimization option also uses various techniques such as clock gating or power gating to reduce dynamic power consumption. Below are the commands used:

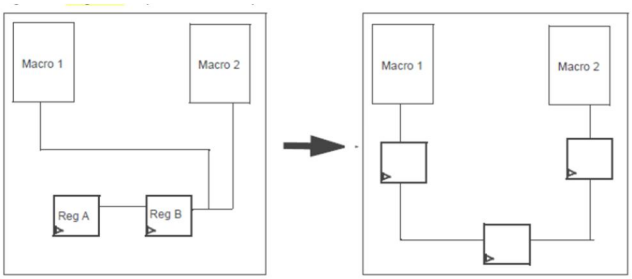
```
set_optimize_registers -check_design -verbose -
print_critical_loop -justification_effort high
compile_ultra -no_autoungroup -
no_seq_output_inversion -gate_clock -scan -retime
```

F. Synthesis for Critical Paths/Modules

There are potential challenges and limitations related to the critical path optimization in the microprocessor and microcontroller designs. Critical path optimization helps in identifying and isolating the poor-performing paths. It is acknowledged that despite the various efforts made in the direction of critical path optimization, there are chances of timing violations. These violations are related to specific paths present in the design that are not able to meet the tight timing constraints. In such cases, the best results can be obtained through physical synthesis, as it provides the designer with further scope for iterations and refinement, still there will be considerable number of paths that are not able to meet their timing targets.

G. Register Cloning

Register cloning creates additional copies or instances of registers present within the design follows an incremental approach which in turn helps in achieving specific optimization goals related to (PPA) requirements. Register cloning can be applied to critical paths that helps in the reduction of clock-to-q delays as it creates multiple instances of registers along a critical path, with this designer also has the flexibility to add pipeline stages, helps in improving the design performance. By the introduction of additional registers, the overall area of the design is also impacted.

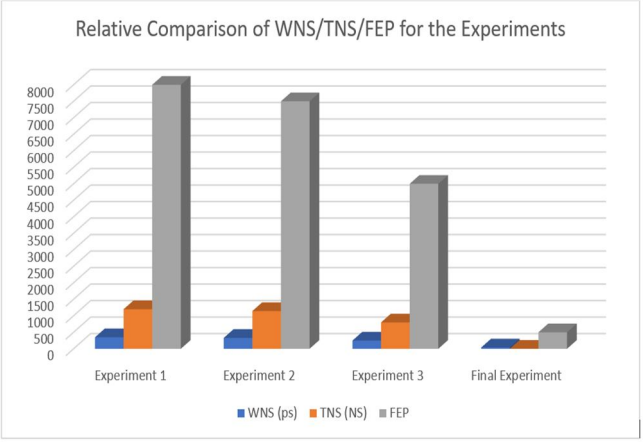



```
set_register_replication -max_fanout 6
-include_fanout_logic U123 {REG1 REG2 REG3
REG4}
```

III. RESULTS

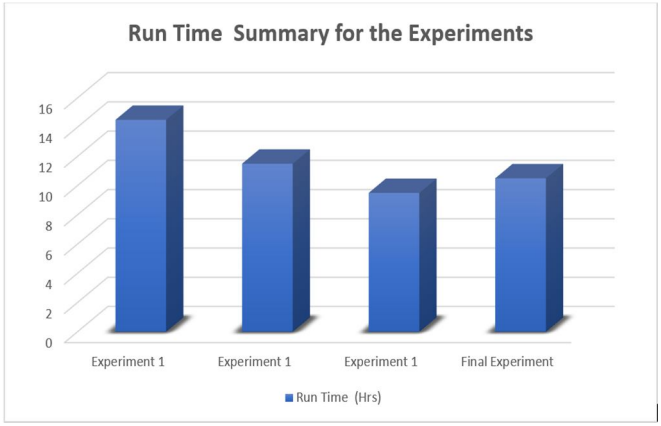
Multiple Experiments were Carried out to achieve the best PPA, out which Best progressive 4 experiments with options used are presented as below:

	Experiment 1	Experiment 2	Experiment 3	Final Experiment
Auto Path groups	YES	YES	YES	YES
Register Cloning	NO	NO	NO	YES
Selective Ungrouping	NO	YES	YES	YES
Selecting Best VT Flavors	YES	YES	YES	YES
Optimize Registers	NO	NO	NO	YES
Smart Gen Options Enabled	YES	YES	YES	YES
Synthesis for the Critical Paths	NO	NO	NO	YES



Leakage Recovery has been a part of all the experiments, and it was targeted to be achieved in final iteration.

Run Time Summary is bit impacted, yet not to higher to achieve the desired PPA:



IV. CONCLUSION

The strategies and approach discussed in this paper aim to minimize the number of iterations, resulting in significant timing improvements. The paper highlights the extensive utilization of different logical and physical synthesis techniques, emphasizing the significance of integrating synthesis within the design process. The methodologies presented are recognized as a crucial factor in achieving optimized results for high-performance microprocessors and microcontroller designs. The recognition of this need suggests an acknowledgment of evolving challenges in microcontroller design and the importance of proactive strategies to address issues related to physical design considerations.

REFERENCES

[1] C. J. Alpert et al., "Techniques for Fast Physical Synthesis," in Proceedings of the IEEE, vol. 95, no. 3, pp. 573-599, March 2007, doi: 10.1109/JPROC.2006.890096.

[2] V. Filaretov, K. Gorshkov and A. Mikheenko, "A circuit synthesis technique based on network determinant expansion," 2012 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Seville, Spain, 2012, pp. 293-296, doi: 10.1109/SMACD.2012.6339397.

[3] L. Trevillyan, D. Kung, R. Puri, L. N. Reddy and M. A. Kazda, "An integrated environment for technology closure of deep-submicron IC designs," in IEEE Design & Test of Computers, vol. 21, no. 1, pp. 14-22, Jan.-Feb. 2004, doi: 10.1109/MDT.2004.1261846.

[4] B. Halpin, R. Chen and N. Sehgal, "Timing driven placement using physical net constraints," Proceedings of the 38th Design Automation Conference (IEEE Cat. No.01CH37232), Las Vegas, NV, USA, 2001, pp. 780-783, doi: 10.1145/378239.379065.

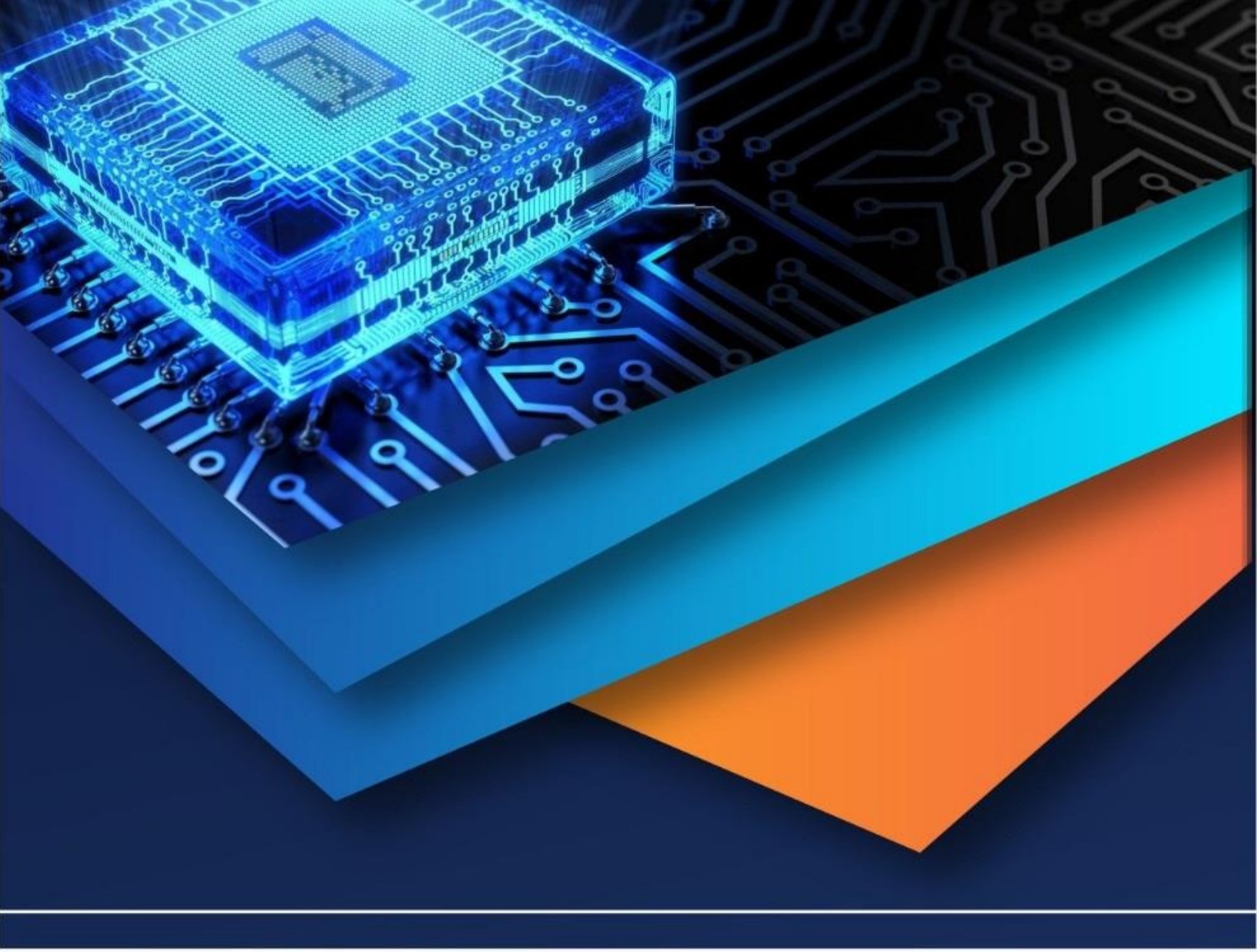
[5] N. Farahini, A. Hemani, H. Sohofi and S. Li, "Physical design aware system level synthesis of hardware," 2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), Samos, Greece, 2015, pp. 141-148, doi: 10.1109/SAMOS.2015.7363669.

[6] Yiu-Hing Chan, P. Kudva, L. Lacey, G. Northrop and T. Rosser, "Physical synthesis methodology for high performance microprocessors," Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451), Anaheim, CA, USA, 2003, pp. 696-701, doi: 10.1109/DAC.2003.1219108.

[7] M. Anwar, S. Saha, M. M. Ziegler and L. Reddy, "Early Scenario Pruning for Efficient Design Space Exploration in Physical Synthesis," 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Kolkata, India, 2016, pp. 116-121, doi: 10.1109/VLSID.2016.94.

[8] S. Maestre, A. Gumera, J. Hora and M. J. de Guzman, "Improving Digital Design PPA (Performance, Power, Area) using iSpatial Physical Restructuring," 2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Phuket, Thailand, 2022, pp. 647-651, doi: 10.1109/ITC-CSCC55581.2022.9895022.

[9] S. H. Loh, I. M. Tan and J. J. 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), Penang, Malaysia, 2021, pp. 55-60, doi: 10.1109/ICCSCE52189.2021.9530746.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY