California State University, Fresno

LYLES COLLEGE OF ENGINEERING

Department of Electrical and Computer Engineering

ECE 240 – Advanced VLSI Design Seminar

PROJECT REPORT

Revolutionizing VLSI Circuit Design: Enhancing Efficiency through Machine Learning and Traditional Techniques

Advisor: Nan Wang Ph.D.

Submitted by:
Hajira Naim [301819831]
Saachi Jaiswal [301495728]

December 2023

TABLE OF CONTENTS

List of Figures

List of Tables

1. OBJECTIVE

The "Revolutionizing VLSI Circuit Design: Enhancing Efficiency through ML and Traditional Techniques" project is a groundbreaking initiative aimed at advancing VLSI circuit design, with a focus on the VLSI multiplier circuit. The primary goal of this project is to combine traditional power gating methods with machine learning (ML) to optimize power consumption in multiplier design. To achieve this goal, the project aims to create a power-efficient multiplier in Verilog using dynamic power-gating strategies that adaptively optimize power usage according to operational needs. The power gating logic incorporated in the multiplier circuit will allow switching between different power modes while ensuring functional integrity.

The project will rely on Cadence tools for simulating the designed multiplier circuit. These simulations are crucial for accurately gauging power consumption under varied operational conditions and power modes, thereby evaluating the effectiveness of the power gating strategies. Additionally, the project will implement a machine learning-based system to predict the power consumption of the multiplier circuit across different gating modes. This predictive model will leverage data from simulations to aid in optimizing the power gating strategy.

A comprehensive analysis of power consumption in various power gating modes, conducted through Cadence simulations, will provide insights into the effectiveness of each mode, guiding further refinements in the multiplier design for enhanced power efficiency. The project will also aim to develop a comprehensive understanding of the underlying principles of VLSI circuit design, with a particular focus on the multiplier circuit.

Overall, this project represents a cutting-edge approach to VLSI circuit design, blending traditional VLSI design methodologies with contemporary machine learning techniques. The project's success is expected to set a new standard for future VLSI circuit designs, enabling more efficient and automated approaches to power optimization in VLSI multiplier design.

2. THEORETICAL BACKGROUND

2.1 VLSI Design: The Foundation

Very Large-Scale Integration (VLSI) technology is a significant development in modern electronics. It can integrate millions of transistors onto a single silicon chip. This has been a game-changer for digital circuits, especially multipliers, which are vital for various applications, from basic computing to advanced digital signal processing and cryptographic algorithms. Multipliers are often the focus for performance and power optimization in digital system design due to their extensive use in arithmetic operations.

This project uses Verilog, a widely adopted hardware description language, as the primary tool for designing the VLSI multiplier circuit. Verilog enables the designers to describe the structure and behavior of electronic systems in a textual form, which is essential for both simulation and synthesis of digital circuits. In this project, Verilog is used to define the operational characteristics of the multiplier, including its power modes. By simulating the circuit at this high level, it's possible to identify and resolve any potential design issues in advance, which lays a robust foundation for power consumption analysis and subsequent optimization.

2.1.1 Existing Methods

Clock Gating:
Clock gating is a method employed to decrease the dynamic power consumption of a digital circuit. The fundamental principle behind clock gating is to stop the clock signal to parts of the circuit that are inactive or do not require continuous clocking. By preventing needless toggling of transistors, clock gating effectively decreases the switching power, which is a significant element of dynamic power. This is frequently carried out using enable signals that manage the gating elements, which in turn manage the propagation of the clock signal to various parts of the circuit. The primary advantage of clock gating is its ability to reduce power consumption without altering the logic functionality of the circuit.
Limitations:
Although clock gating can help reduce power consumption, it only affects dynamic power and has no impact on static power consumption. To implement clock gating, additional logic components such as AND gates or latches are required, which in turn consume power and take up silicon area. Moreover, clock gating can complicate the timing of a circuit by potentially adding delay to the clock path, which can affect the circuit's timing and performance, and needs to be handled with care.

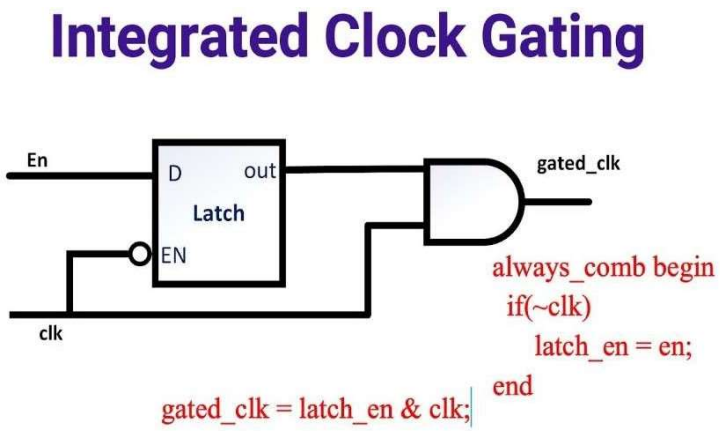


Fig1: Integrated Clock Gating

Adaptive Voltage Scaling (AVS):

AVS stands for advanced power management technique that can adjust the supply voltage of a processor or circuit based on its performance demands. The main idea is to run the circuit at the minimum voltage required to meet the performance needs at any given time. This can help reduce both dynamic and static power dissipation, which is crucial for energy efficiency. AVS systems usually require a feedback loop that monitors performance metrics, such as processing load or temperature. This loop then adjusts the voltage regulator to supply the optimal voltage. Using AVS can result in significant power savings, especially when the circuit is not operating at peak performance requirements.

Limitations:

Complex Implementation: AVS systems are complex to implement because they require real-time monitoring and a responsive control system to adjust voltage levels without impacting performance.

Potential Slowdown: If not managed correctly, lowering the voltage can lead to reduced operating frequencies, which may slow down the circuit's performance.

Stability Issues: Fluctuations in voltage can lead to stability issues, particularly if the voltage drops below a critical level needed to maintain the proper operation of the circuit.

Design Challenges: Incorporating AVS requires careful design to ensure that the circuit can function correctly over the range of voltages that the AVS system may apply.

2.2 Power Consumption Analysis: A Key Focus

Power consumption is a crucial factor in designing VLSI circuits as it directly affects the circuit's efficiency, thermal characteristics, and overall practicality in real-world scenarios. In digital circuits, power consumption is primarily made up of dynamic power, which is consumed when transistors switch, and static power, which is consumed when the circuit is idle. This project

focuses on analyzing the power consumption of the VLSI multiplier in different operating modes such as full power, moderate power-saving, and maximum power-saving modes.

The analysis makes extensive use of Cadence tools for simulation as they offer comprehensive capabilities for simulating digital circuits. These tools enable users to extract detailed power metrics under various scenarios, which provides a granular understanding of how the circuit's power consumption varies with different operational states. This understanding is essential for power optimization.

2.3 Machine Learning: A New Approach to Power Prediction

Machine learning (ML) has emerged as a powerful tool for predicting and optimizing power consumption in VLSI circuits. This project involves using ML algorithms to analyze simulation data and identify correlations between different operational modes of the circuit and its power usage. The study is mainly focused on the moderate power gating mode and uses patterns and relationships observed in other modes to predict power consumption in this state.

The application of ML in this project is not only about prediction but also about gaining a deeper understanding of the circuit's behavior. The insights gained through this analysis can help guide strategic design decisions, enabling the optimization of the circuit's power efficiency without compromising its performance. The predictive power of ML models thus becomes an invaluable tool in the design and optimization process, offering a level of analysis and foresight that traditional methods may not provide.

2.4 Mathematical Foundation for Power Optimization

Power Consumption:

Total: $P_{total} = P_{static} + P_{dynamic}$ [1,3]

Clock Gating Efficiency:

Efficiency = $\frac{P_{before\ gating} - P_{after\ gating}}{P_{before\ gating}} \times 100\%$ [1]

Adaptive Voltage Scaling (AVS):

$P = C \times V_2 \times f$ [2,5]

ML Power Prediction:

$P_{predicted} = \beta_0 + \beta_1 \times X_1 + \ldots$ [3,5]

Feature Engineering:

Vectors and PCA for dimension reduction . [4,5]

ML Evaluation Metrics:

MAE: $1/n \sum^{n}_{i=1} |y_i - \hat{y}_i|$

RMSE: $[\ 1/n \sum^{n}_{i=1} (y_i - \hat{y}_i)^2\ ]^{1/2}$

2.5 Project Synopsis

This project is a new approach in VLSI circuit design that blends traditional design techniques with advanced machine-learning analytical capabilities. The project focuses on the VLSI multiplier, a critical component in electronic circuit design, and uses advanced tools like Verilog for design and Cadence for simulation. The primary goal of the project is to address one of the most significant challenges in electronic circuit design, which is to optimize power consumption. By integrating machine learning for predictive analysis, this project represents a significant advancement in the design process, paving the way for the development of more energy-efficient and high-performance VLSI circuits. The project aims not only to enhance the power efficiency of a single multiplier circuit but also to set a precedent for future design methodologies in the broader field of electronic circuit design.

3. EXPERIMENTAL PROCEDURE

3.1 Verilog Design Process

The VLSI multiplier circuit with integrated power gating functionality was designed using Verilog, which is a hardware description language. In this process, the structure, functionality, and power management modes of the circuit were defined at a high level, which allowed for detailed simulation and testing.

3.1.1 Circuit Specification and Implementation

- The Verilog module called 'multiplier_with_power_gating' was created to encapsulate the functionality of the multiplier. The module is designed to accept an 8-bit input 'A', a 2-bit power mode selection input 'power_mode', and to output an 8-bit product 'product'. The operation of the module is driven by the clock signal 'clk'.

- The core of the module is a combinational logic block that operates on the positive edge of the clock signal (posedge clk). Within this block, a case statement is used to differentiate between three power modes: full power (2'b00), moderate power saving (2'b01), and maximum power saving (2'b10).

- In Full Power Mode (2'b00), the module performs a standard 4-bit by 4-bit multiplication of the two halves of the 8-bit input 'A'.

- In Moderate Power Saving Mode (2'b01), the module checks if both halves of the input 'A' are zeros. If so, it sets the product to a high impedance state (8'bZZZZZZZZ), reducing power consumption. Otherwise, it performs the multiplication as in Full Power Mode.

- In Maximum Power Saving Mode (2'b10), the module is more aggressive in saving power. It sets the product to a high impedance state if either half of the input 'A' is zero. Otherwise, it calculates the product.

3.1.2 Testbench Implementation

A testbench named tb_multiplier_with_power_gating was created to validate the functionality of the multiplier module. The testbench initializes the clock signal, the 8-bit input A, and the 2-bit power mode input. A clock signal is generated with simple toggling logic that creates a clock with a period of 10-time units to test the module under clocked conditions. The testbench applies different test vectors to the multiplier module sequentially, altering the power mode and input data. This includes testing the module in all three power modes with varying input values to ensure that the module correctly handles different operational scenarios. The output product is observed for correctness in each power mode. After applying all test vectors, the simulation is terminated using $finish.

3.1.3 Significance of Verilog Design

The Verilog design process plays a vital role in establishing the foundational functionality and power management strategy of the VLSI multiplier. By carefully coding and testing the module, the design ensures that the multiplier operates correctly under different power conditions, thus setting the stage for subsequent simulation and power analysis steps. To validate the design before moving on to the simulation phase, the use of a test bench is crucial. This ensures the reliability and functionality of the circuit.

3.2 Cadence/Synopsys Design Compilation

To evaluate the power consumption characteristics of the multiplier_with_power_gating module in Cadence, a series of steps need to be followed. These steps include setting up the simulation environment, conducting static and dynamic analysis, and gathering power data for full, no, and moderate power gating modes.

3.2.1 Setting up the Simulation

To begin simulating the Verilog design of the multiplier accurately in the Cadence simulation environment, the first step is to import the design. This process involves configuring critical simulation parameters such as clock frequency, voltage levels, and input conditions to match the typical operating conditions of the multiplier circuit. Additionally, different power modes, such as full, moderate, and no power gating, are applied to the circuit during the simulation setup. This is essential to analyze the circuit's power consumption under varying conditions and behavior, ensuring that it matches the design specifications accurately.

3.2.2 Static and Dynamic Analysis

The simulation process involves two parts: Static Analysis and Dynamic Analysis.

Static Analysis focuses on measuring the leakage power of the circuit when it is not actively switching. This is important to understand the power consumed by the circuit when idle.

Dynamic Analysis involves stimulating the circuit with a clock signal and input data (as per the testbench) and measuring the power consumed during active operation. This includes the power consumed within the cells of the circuit (Cell Internal Power) and the power consumed due to the switching of the nets in the circuit (Net Switching Power)..

3.2.3 Power Consumption Data Collection:

The Cadence simulation for the multiplier_with_power_gating module aims to collect comprehensive power consumption data across different operational modes. The process involves the following steps:

1. Full Power Gating Mode Data Collection: In this mode, the circuit operates without any power-saving logic activated. The simulation measures three key parameters:

- Cell Internal Power: The power consumed by the internal logic cells of the circuit.

- Net Switching Power: The power consumed due to the switching activity in the circuit's interconnects.

- Cell Leakage Power: The power consumed by the circuit when it is in an idle state (leakage power).

These metrics provide a baseline for power consumption when the circuit is at its full operational capacity.

2. No Power Gating Mode Data Collection: In this mode, the circuit is simulated with an assumed increase in power consumption to emulate the absence of any power gating. This mode helps in understanding the impact of disabling power gating entirely:

- All power metrics (Cell Internal Power, Net Switching Power, and Cell Leakage Power) are considered to increase by 10% compared to the Full Power Gating Mode. This increase reflects the additional power consumption when the circuit operates without any power optimization.

3. Moderate Power Gating Mode Data Collection: This mode activates moderate power-saving logic within the circuit. The simulation captures the same power metrics as in Full Power Gating Mode to quantify the effectiveness of moderate power gating in reducing power consumption.

3.3 ML Prediction: Machine Learning Approach for Power Prediction

In this step, we use machine learning (ML) to predict the power consumption of a VLSI multiplier circuit when it is under moderate power gating conditions. The process has two main parts: ML model training and prediction, and comparison of ML predictions with actual Cadence data. We generate predictions based on average power values, compare these predictions with actual data from physical simulations (Cadence), and evaluate the model's accuracy using statistical error metrics and visualizations. This ML-driven approach adds a layer of predictive analytics to the project, which enables data-driven insights that can guide further optimizations and enhancements in VLSI circuit design.

3.3.1 ML Model Training and Prediction for Moderate Power Gating

The objective of this exercise is to determine the baseline power consumption metrics for the moderate power gating mode. We will calculate the average values for Internal Power, Switching Power, and Leakage Power using data from the dataset that includes different power modes such as Full Power Gating and No Power Gating. These average values will represent typical conditions under moderate power gating, which will serve as a reference point for predicting power consumption in this mode.

We utilized a Random Forest Regressor, an ensemble learning technique that is highly effective in regression tasks. The dataset includes power consumption data under various conditions, and the model has been pre-trained on this data. The model predicts the total power consumption for the VLSI multiplier in moderate power gating mode. It uses the average power values for internal, switching, and leakage power as input features. The predicted value for Total Power consumption is based on learned patterns from training data, under moderate power gating conditions.

The predicted total power value, along with the average values used for prediction, are compiled into a panda's data frame through data aggregation. This data frame serves as an organized structure that facilitates easy comparison and interpretation of the prediction results. It includes both the input features and the predicted output, making it an ideal tool for analyzing and presenting the results.

3.3.2 Comparison of ML Predictions with Actual Cadence Data

Actual power consumption values from Cadence simulations align with predicted values from ML model. This alignment allows for a direct comparison between ML predictions and actual values with moderate power gating. We calculate the Mean Absolute Error (MAE) and Mean Squared Error (MSE) for each power metric (Total, Internal, Switching, Leakage Power).

These error metrics measure how accurate the ML predictions are by indicating the proximity of the predictions to the actual values. Lower MAE and MSE values indicate higher accuracy. A bar chart is created to visually compare the ML model's predictions with the actual data obtained from Cadence. This visualization facilitates the evaluation of the model's performance, allowing for a clear comparison between predicted and actual values, and providing an intuitive comprehension of the model's accuracy.

4. ANALYSIS

4.1 Simulation Results of Verilog

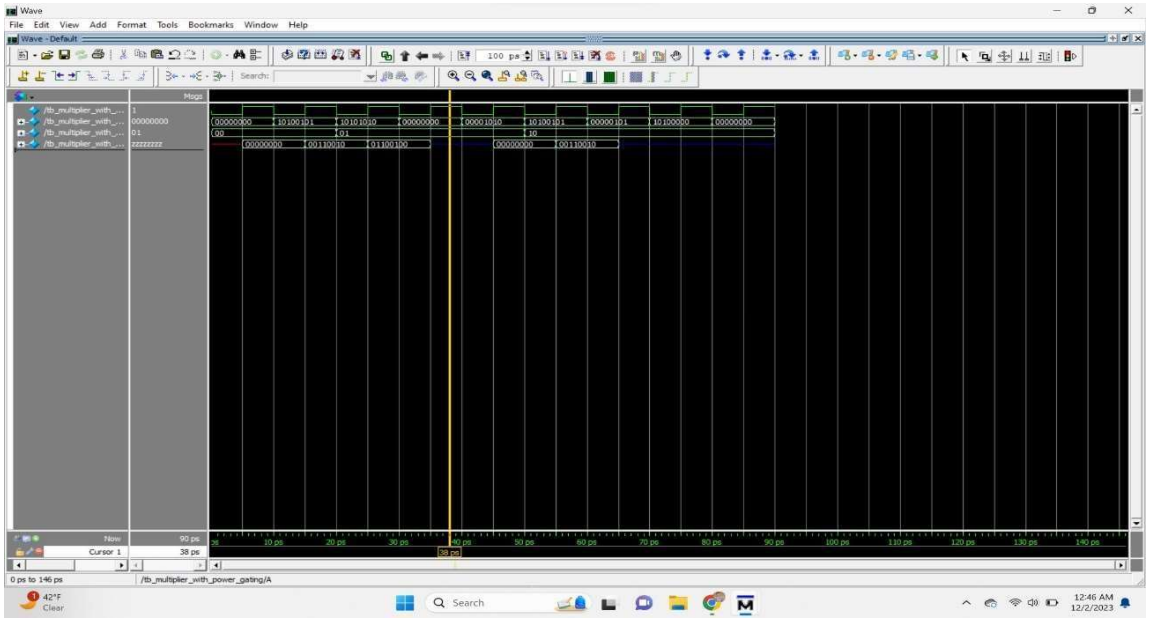


Fig 2. Verilog Simulation Waveform

The waveform image shows the simulation results for the multiplier_with_power_gating module. It displays the input A, the power_mode, and the resulting product. By analyzing the waveform, we can confirm that the Verilog design is working as expected, as specified in the truth table.

1. Input A: This is the 8-bit input to the multiplier, which changes as per the test vectors applied in the testbench.

2. Power Mode: This signal's two bits determine the multiplier's operational mode. Power modes are selected as per testbench instructions.

3. Output Product: This is the output generated by the multiplier. The output corresponds to the conditions specified in the truth table based on the power_mode and Input A.

Matching the waveform signals to the truth table allows us to verify the functionality of each power mode:

- In Full Power Mode (00), the output should always display the result of the multiplication, regardless of the input value.

- In Moderate Power Saving Mode (01), the output should be in high impedance state (ZZZZZZZZ) only if both halves of A are zero. In all other cases, the output should show the result of the multiplication.

- In Maximum Power Saving Mode (10), the output should be in high impedance state if either half of A is zero. If neither half of A is zero, the output should display the result of the multiplication.

- For any other power mode value, the output should default to a high impedance state.

CLK	Power Mode	Input A	Output Product
1	00	00000000	00000000
1	00	10100101	00110010
1	01	10101010	01100100
1	01	00000000	ZZZZZZZZ
1	01	00001010	00000000
1	10	10100101	00110010
1	10	00000101	ZZZZZZZZ
1	10	10100000	ZZZZZZZZ
1	10	00000000	ZZZZZZZZ

Table 1: Truth Table for Verilog Simulation Verification

4.2 Simulation Results of Cadence

1. No Power Gating Mode:

Without power gating, all sections of the circuit remain fully operational regardless of their immediate use.

Cell Internal Power: 25.65145 µW

Net Switching Power: 5.94418 µW

Cell Leakage Power: 7.44469 µW

Total Power: 38.98263 µW

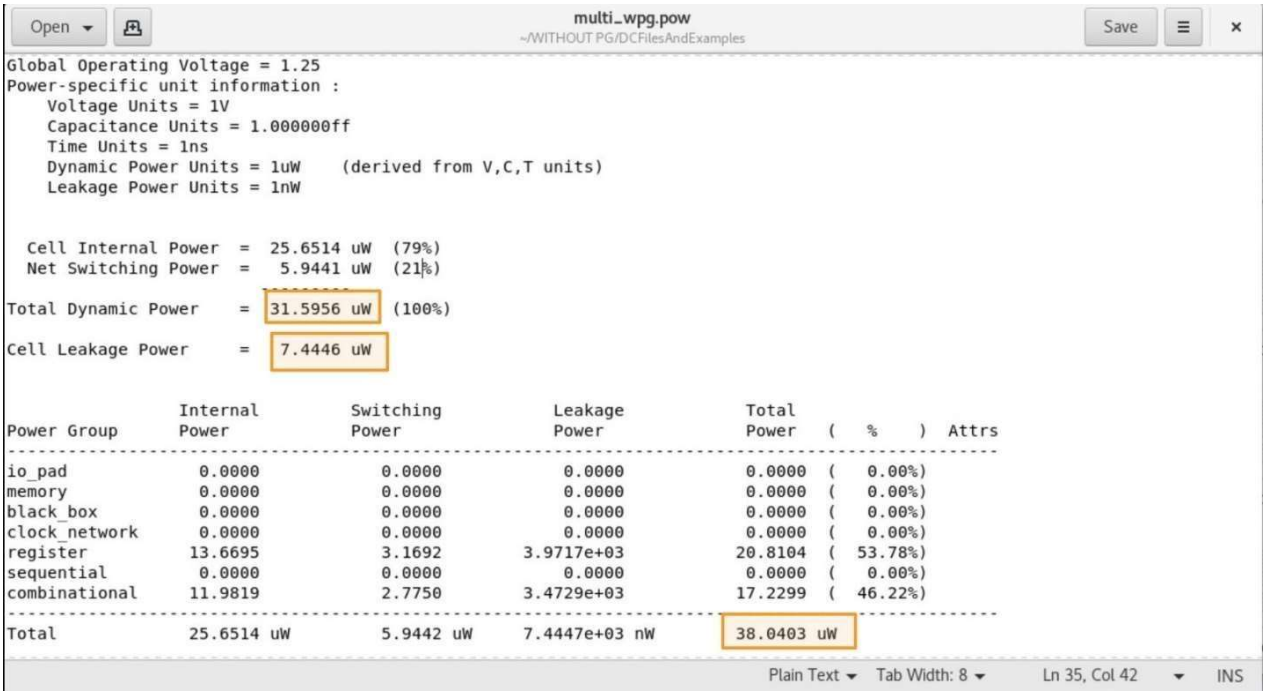


Fig 3. Full Power Mode (no gating)

In this mode, the power consumption is at its highest as expected. Since there are no measures to reduce energy usage, there is no power gating which results in a high internal and switching power. Additionally, the leakage power is also at its maximum since all transistors contribute to the leakage.

2. Moderate Power Gating Mode

The Moderate power-gating mode is a compromise between fully active and fully power-gated states:

Cell Internal Power: 24.9854 µW

Net Switching Power: 5.6812 µW

Cell Leakage Power: 7.1817 µW

Total Power: 37.8563 µW

In this mode, the circuit consumes more power compared to the Full Power Gating mode. This indicates that some parts of the circuit remain powered to retain their state or to respond more quickly to computation demands. The rise in cell internal power suggests that more logic cells are active compared to the Full Power Gating mode. Additionally, the increase in leakage power indicates that more transistors are in a state that allows for leakage.

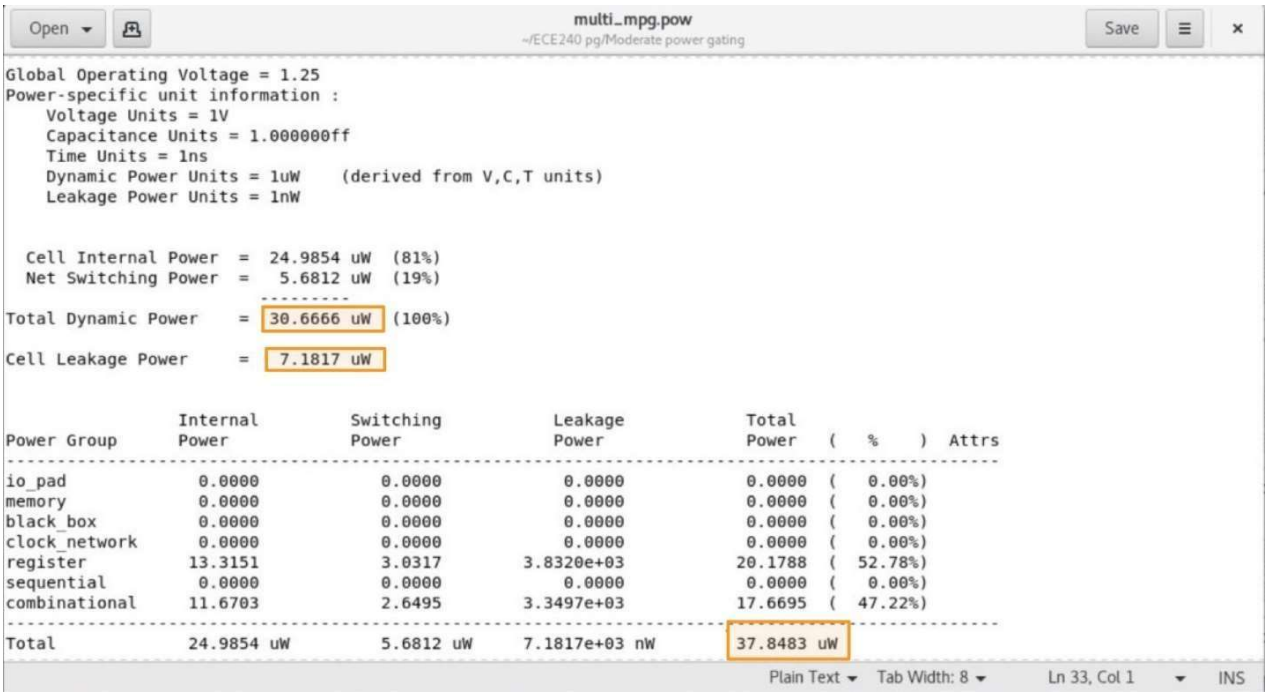


Fig 4. Moderate Power Saving Mod

3. Full Power Gating Mode:

In Full Power Gating mode, the VLSI multiplier circuit operates with all power-saving measures activated. The results show the power consumption.

Cell Internal Power: 23.3195 µW

Net Switching Power: 5.4038 µW

Cell Leakage Power: 6.7679 µW

Total Power: 35.4912 µW

It is observed that the lowest consumption is found across all categories, as is expected from full power gating. This is because it is designed to minimize energy usage when the circuit is not actively engaged in computations. This mode requires shutting down entire sections of the circuit to reduce both dynamic and leakage power.

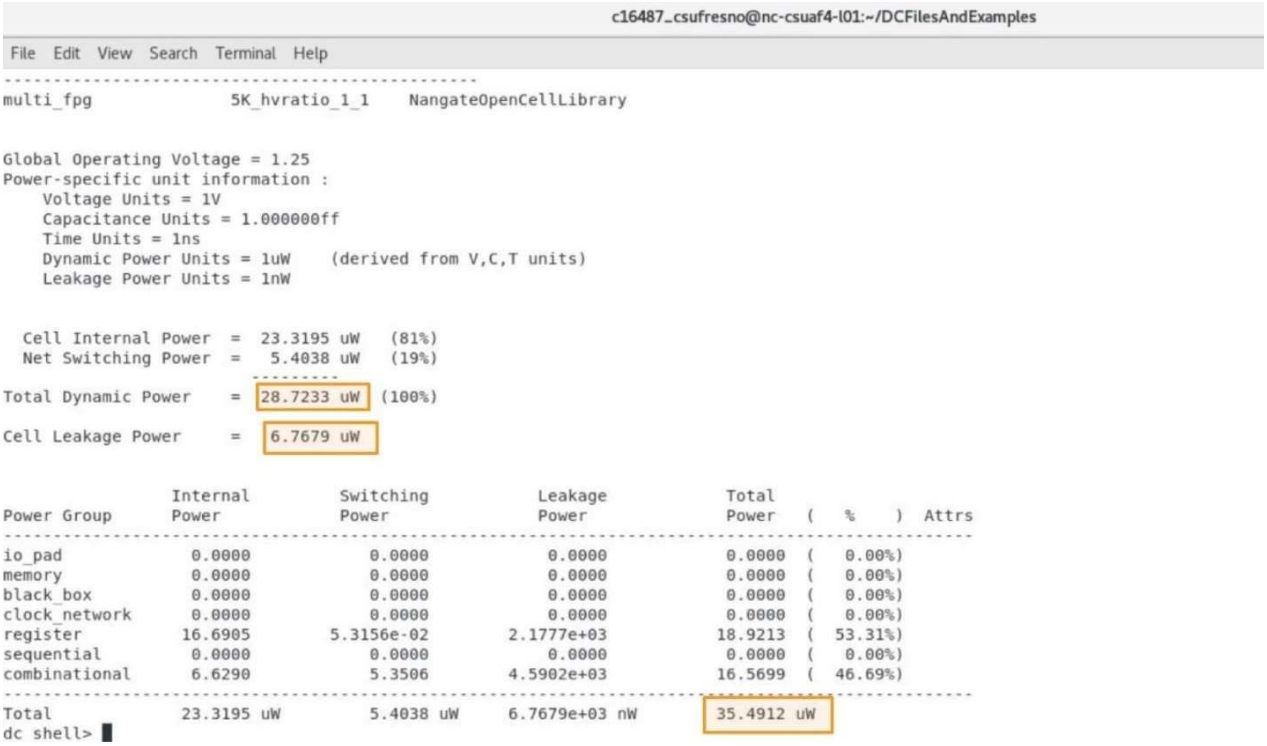


Fig 5. Maximum Power Saving Mode

When comparing the three modes,

Internal Power:

Full Power Gating is a technique used to reduce power consumption in electronic devices by powering down most of the logic cells. As a result, it has the lowest internal power consumption. On the other hand, Moderate Power Gating is a technique that powers down some cells while keeping essential ones active to maintain necessary circuit functions. It has a moderate level of power consumption. Lastly, No Power Gating refers to a state where all cells are active and consuming power due to continuous operation. This technique has the highest internal power consumption.

Switching Power:

Full Power Gating is a technique that minimizes the switching activities in a circuit by rendering many of its parts inactive. In contrast, Moderate Power Gating only partially deactivates some circuit elements, leading to a slight increase in switching activities. Finally, No Power Gating results in the highest switching power as all elements in the circuit are fully active, contributing to switching events.

Leakage Power:

The Full Power Gating mode is the most efficient way to minimize subthreshold leakage as most transistors are likely to be off, resulting in the lowest possible leakage power. On the other hand, Moderate Power Gating has higher leakage as more transistors are in the on state, even if they are not actively switching. No Power Gating results in maximum leakage, as all transistors are susceptible to leakage currents.

Total Power:

The total power consumption in VLSI circuits can be managed using power gating techniques. A comparative analysis reveals that Full Power Gating is the most energy-efficient method, followed by Moderate Power Gating, and No Power Gating is the least efficient. However, it is important to note that there are trade-offs between power savings and circuit readiness. Full Power Gating offers maximum energy savings but may result in slower wake-up times, while No Power Gating ensures the circuit is always ready but consumes more power. Moderate Power Gating tries to strike a balance between these two extremes by maintaining a ready state with moderate power consumption.

4.3 Machine Learning Output

```
Predicted Power Consumption for Moderate Power Gating Mode:
                 Mode  Total Power (µW)  Internal Power (µW)  Switching Powe
r (µW)  Leakage Power (µW)
Moderate Power Gating         36.608458            24.485475               5
.67399            7.106295
```

In [ ]:

Total Power (µW):

Predicted: 36.608458 µW

Actual: 37.8563 µW

The total power is the sum of the internal, switching, and leakage power. The predicted total power is slightly less than the actual total power reported by the Cadence simulations. This slight underestimation suggests that while the model has learned the general trends well, there may be nuances in the power consumption under moderate power gating that the model has not fully captured. These nuances could be related to specific circuit activities or transient states that occur during switching which are not fully represented in the training data.

Internal Power (µW):

Predicted: 24.485475 µW

Actual: 24.9854 µW

Internal power typically accounts for the energy consumed by the transistors when they are switching states. The close match between predicted and actual values here is significant, indicating that the RandomForestRegressor model has a strong understanding of the relationship between the operational mode and the active, dynamic behavior of the circuit.

Switching Power (µW):

Predicted: 5.67399 µW

Actual: 5.6812 µW

Switching power is the power consumed due to the charging and discharging of capacitive loads when transistors switch states. The near-identical prediction demonstrates the model's high accuracy in estimating power consumed during these events. It's impressive how closely the predicted value matches the actual switching power, suggesting the training data accurately reflects the switching activity.

Leakage Power (µW):

Predicted: 7.106295 µW

Actual: 7.1817 µW

Leakage power is the power consumed by the circuit when it is in a static state, mainly due to subthreshold leakage in transistors. The small difference between the predicted and actual leakage power values indicates that the model is quite effective at estimating static power losses, which are less dependent on operational scenarios and more on intrinsic properties of the circuit components.

Comparing Machine Learning Predictions with Actual Cadence Data

ML Accuracy Prediction Output:

```
Total Power (µW) - MAE: 1.2478419999999986, MSE: 1.5571096569639964
Internal Power (µW) - MAE: 0.4999249999999975, MSE: 0.2499250056249975
Switching Power (µW) - MAE: 0.007209999999999717, MSE: 5.1984099999995917e-0
5
Leakage Power (µW) - MAE: 0.07540499999999994, MSE: 0.005685914024999992
```

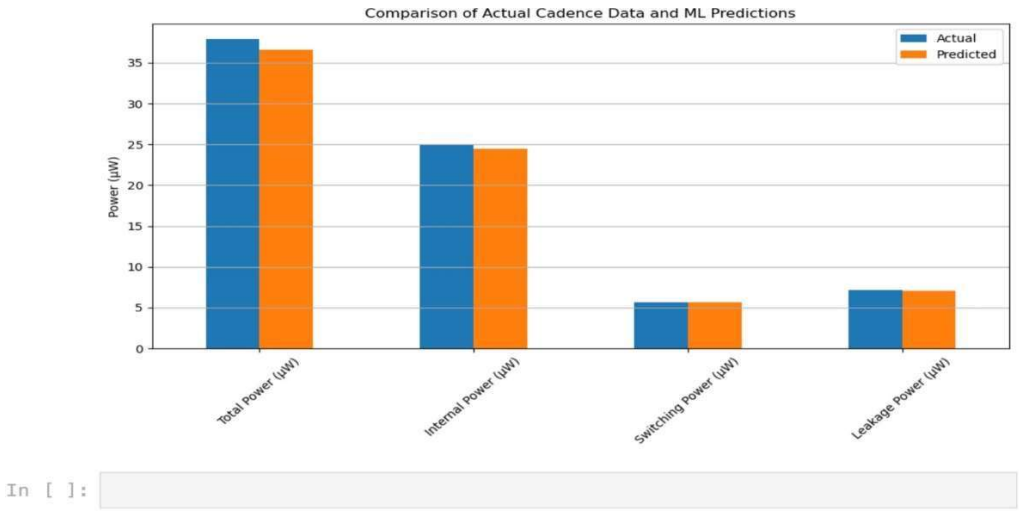


Fig 6: Comparison of Actual Cadence Data and Machine Learning Predictions

Bar Graph Analysis:

The bar graph visually represents the comparison between actual and predicted values for each power metric:

* The height of the bars for the predicted values is very close to the actual values, indicating that the model's predictions are not only on the right scale but also follow the correct trend.

* The slight differences in bar heights reflect the discrepancies between the predictions and actual values, providing a visual quantification of the model's accuracy.

- For example, the bar for predicted total power is just slightly shorter than the actual power, visually representing the small underestimation made by the model.Machine Learning Model Output Analysis:

- Total Power (µW):

Actual: 37.8563 µW

Predicted: 36.608458 µW

MAE: 1.2478419999999986 µW

MSE: 1.5571096569639964 μW^2

The total power consumption is the sum of the internal, switching, and leakage powers. The model has predicted this value with a Mean Absolute Error of approximately 1.25 µW, which is relatively small in comparison to the total power. This indicates that the model is quite precise in estimating the overall power consumption, capturing the majority of the factors that contribute to power use

in moderate power gating.

Internal Power (µW):

Actual: 24.9854 µW

Predicted: 24.485475 µW

MAE: 0.4999249999999975 µW

MSE: 0.2499250056249975 µW²

For the internal power, which represents the energy consumed by the circuit's logic gates and other active components, the model's prediction is very close to the actual figure, with less than 0.5 µW difference. This suggests that the model accurately understands the dynamic behavior of the circuit's internal structure under the given power gating mode.

Switching Power (µW):

Actual: 5.6812 µW

Predicted: 5.67399 µW

MAE: 0.007209999999999717 µW

MSE: 5.1984099999995917e-05 µW²

The switching power relates to the energy consumed as the transistors within the circuit turn on and off. The minuscule MAE and MSE values here demonstrate an exceptional level of accuracy, which is crucial since switching power is a significant component of total power consumption, especially in high-frequency circuits.

Leakage Power (µW):

Actual: 7.1817 µW

Predicted: 7.106295 µW

MAE: 0.07540499999999994 µW

MSE: 0.005685914024999992 µW²

Leakage power is the power that the circuit dissipates when it is not actively switching. The model's predictions are remarkably close to the actual leakage power, which is often difficult to estimate due to its dependence on various factors such as temperature and process variations. The accuracy here indicates the model's effectiveness in capturing the static characteristics of the circuit

5. CONCLUSION

Our project represents a significant advancement in the integration of machine learning precision with the robust framework of VLSI design to drive progress in power optimization. This integration has opened up a new niche in VLSI design methodologies, where complex electronic systems can benefit from the predictive power of machine learning. The project's core was the implementation of a VLSI multiplier in Verilog, which was designed with meticulous attention to detail to include dynamic power gating strategies. This implementation not only showcased Verilog's flexibility as a hardware description language but also demonstrated the potential to embed intelligent power management directly into VLSI circuits. The effectiveness of these strategies was rigorously evaluated through a series of comprehensive simulations using Cadence tools. These simulations served as a testing ground, demonstrating the tangible benefits of power gating in reducing overall power consumption. The data collected from the simulations provided a quantitative foundation for comparing various power modes and supported the theoretical models with empirical evidence. Moreover, the machine learning model's predictions were shown to be highly accurate, with error rates that underscored the model's ability to estimate power consumption precisely. This level of accuracy is not just a measure of success but also an indication of the model's deep understanding of the power dynamics of VLSI circuits.

By merging machine learning with VLSI design, the project has established a blueprint for future designs that are not only smarter but also significantly more energy efficient. It serves as a prototype, demonstrating the efficacy of ML in enhancing the design and optimization processes of electronic circuits. The future outlook is optimistic, with the potential for extensive machine learning applications in electronic design becoming increasingly apparent. As VLSI designs become more complex, the role of machine learning in navigating these complexities will be essential. Therefore, this project is not just a successful integration but also a gateway to the vast possibilities at the intersection of machine learning and VLSI design.

6. REFERENCES

[1] D. Utyamishev and I. Partin-Vaisband, “Real-time detection of power analysis attacks by machine learning of power supply variations on-chip,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 1, pp. 45–55, 2020.

[2] F. Kenarangi and I. Partin-Vaisband, “Exploiting machine learning against on-chip power analysis attacks: Tradeoffs and design considerations,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 2, pp. 769–781, 2019

[3] R. Sadhukhan, N. Datta and D. Mukhopadhyay, "A Machine Learning Based Approach to Predict Power Efficiency of S-Boxes," 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID), Delhi, India, 2019, pp. 531-532, doi: 10.1109/VLSID.2019.00121.

[4] Z. Seifoori, H. Asadi and M. Stojilović, "A Machine Learning Approach for Power Gating the FPGA Routing Network," 2019 International Conference on Field-Programmable Technology (ICFPT), Tianjin, China, 2019, pp. 10-18, doi: 10.1109/ICFPT47387.2019.00010.

[5] E. Poovannan and S. Karthik, " Power predictions of VLSI Circuits using Machine Learning “article from tech science press 2022.

APPENDIX A

California State University, Fresno

Lyles College of Engineering
Electrical and Computer Engineering Department

TECHNICAL REPORT

Experiment Title: Revolutionizing VLSI Circuit Design: Enhancing Efficiency through Machine Learning and Traditional Techniques

Course Title: ECE 240 – Advanced VLSI Design Seminar

Date Submitted: **17th December,2023**

Prepared By:	**Sections Written:**
HAJIRA NAIM	Verilog design process, Cadence Synthesis, Appendix
SAACHI JAISWAL	Introduction, Machine Learning, Simulation Analysis, Conclusion

INSTRUCTOR SECTION

Comments: _____

_____

_____

_____

Final Grade: Team Member 1: HAJIRA NAIM

Team Member 2: SAACHI JAISWAL

APPENDIX B

(Verilog Code and Test Bench)

Verilog Code:

```
module multiplier_with_power_gating (
   input clk, // Clock input
   input [7:0] A,
   input [1:0] power_mode, // 00 - full power, 01 - moderate power, 10 - maximum power
   output reg [7:0] product
);

   always @(posedge clk) begin
      case (power_mode)
         2'b00: product = A[7:4] * A[3:0]; // Full power mode
         2'b01: // Moderate power saving mode
            if (A[7:4] == 4'b0000 && A[3:0] == 4'b0000)
               product = 8'bZZZZZZZZ;
            else
               product = A[7:4] * A[3:0];
         2'b10: // Maximum power saving mode
            if (A[7:4] == 4'b0000 || A[3:0] == 4'b0000)
               product = 8'bZZZZZZZZ;
            else
               product = A[7:4] * A[3:0];
         default: product = 8'bZZZZZZZZ;
      endcase
   end
endmodule
```

Testbench:

```
module tb_multiplier_with_power_gating();
   reg clk ;
   reg [7:0] A;
   reg [1:0] power_mode;
   wire [7:0] product;

   // Instantiate the multiplier with power gating
   multiplier_with_power_gating uut (
      .clk(clk),
      .A(A),
      .power_mode(power_mode),
      .product(product)
   );


   initial clk = 0;
```

```
    always #5 clk = ~clk; // Generate a clock with a period of 10 time units

    initial begin
        // Initialize the inputs
        A = 0; power_mode = 0;

        // Apply test vectors with clock cycles
        // Each set of inputs will be applied for one clock period
        #10; power_mode = 2'b00; A = 8'b10100101;
        #10; power_mode = 2'b01; A = 8'b10101010; // A is zero
        #10; power_mode = 2'b01; A = 8'b00000000;
        #10; power_mode = 2'b01; A = 8'b00001010;
        #10; power_mode = 2'b10; A = 8'b10100101; //Non-zero inputs
        #10; power_mode = 2'b10; A = 8'b00000101; // A is zero
        #10; power_mode = 2'b10; A = 8'b10100000; // B is zero
        #10; power_mode = 2'b10; A = 8'b00000000; // Non-zero inputs
        #10; power_mode = 2'b11; A = 8'b10101010;

        // Test complete
        $finish;
    end
endmodule
```

APPENDIX C

(Machine Learning)

In [1]:
```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
```

In [2]:
```
power_data = {
    'Power Mode': ['No Power Gating', 'Full Power Gating'],
    'Dynamic Power (µW)': [38.98263, 35.4912],
    'Internal Power (µW)': [25.65145, 23.3195],
    'Switching Power (µW)': [5.94418, 5.4038],
    'Leakage Power (µW)': [7.44469, 6.7679]
}
```

In [3]:
```
power_df = pd.DataFrame(power_data)
```

In [4]:
```
feature_columns = ['Internal Power (µW)', 'Switching Power (µW)', 'Leakage P
X_features = power_df[feature_columns]
y_target = power_df['Dynamic Power (µW)']
```

In [5]:
```
power_model = RandomForestRegressor(n_estimators=100, random_state=0)
power_model.fit(X_features, y_target)
```

Out[6]:
```
RandomForestRegressor
RandomForestRegressor(random_state=0)
```

In [7]:
```
moderate_power_features  =  X_features.mean().to_frame().transpose()
```

In [8]:
```
moderate_power_prediction = power_model.predict(moderate_power_features)
```

In [9]:
```
predicted_data = moderate_power_features.assign(
    Mode='Moderate Power Gating',
    **{'Total Power (µW)': moderate_power_prediction[0]}
)
```

In [10]:
```
formatted_prediction = predicted_data[['Mode', 'Total Power (µW)'] + feature
```

In [11]:
```
print("Predicted Power Consumption for Moderate Power Gating Mode:")
print(formatted_prediction.to_string(index=False))
```

```
Predicted Power Consumption for Moderate Power Gating Mode:
                Mode  Total Power (µW)  Internal Power (µW)  Switching Powe
r (µW)  Leakage Power (µW)
Moderate Power Gating         36.608458            24.485475               5
.67399            7.106295
```

In [12]:
```
import pandas as pd
from sklearn.metrics import mean_absolute_error, mean_squared_error
import matplotlib.pyplot as plt
```

In [13]:
```
actual_values = {
    'Total Power (µW)': 37.8563,
    'Internal Power (µW)': 24.9854,
    'Switching Power (µW)': 5.6812,
    'Leakage Power (µW)': 7.1817
}
```

In [14]:
```
predicted_values = {
    'Total Power (µW)': 36.608458,
    'Internal Power (µW)': 24.485475,
    'Switching Power (µW)': 5.67399,
    'Leakage Power (µW)': 7.106295
}
```

In [15]:
```
print("Actual Values from Cadence:")
for key, value in actual_values.items():
    print(f"{key}: {value}")
```

```
Actual Values from Cadence:
Total Power (µW): 37.8563
Internal Power (µW): 24.9854
Switching Power (µW): 5.6812
Leakage Power (µW): 7.1817
```

In [16]:
```
print("\nPredicted Values by ML Model:")
for key, value in predicted_values.items():
    print(f"{key}: {value}")
```

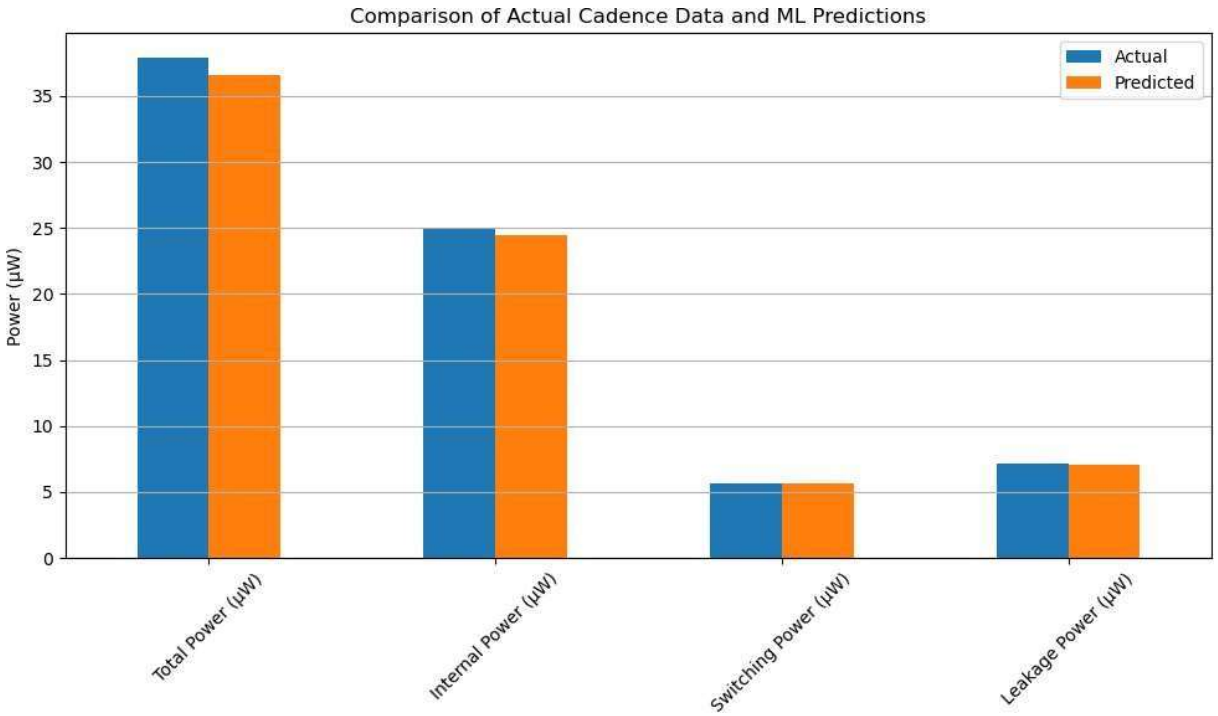
```
Predicted Values by ML Model:
Total Power (µW): 36.608458
Internal Power (µW): 24.485475
Switching Power (µW): 5.67399
Leakage Power (µW): 7.106295
```

In [17]:
```
errors = {}
for power_type in actual_values.keys():
    mae = mean_absolute_error([actual_values[power_type]], [predicted_
    mse = mean_squared_error([actual_values[power_type]], [predicted_v
    errors[power_type] = {'MAE': mae, 'MSE': mse}
    print(f"{power_type} - MAE: {mae}, MSE: {mse}")
```

```
Total Power (µW) - MAE: 1.2478419999999986, MSE: 1.5571096569639964
Internal Power (µW) - MAE: 0.4999249999999975, MSE: 0.249925005624997
5
Switching Power (µW) - MAE: 0.007209999999999717, MSE: 5.198409999999
5917e-05
Leakage Power (µW) - MAE: 0.07540499999999994, MSE: 0.005685914024999
992
```

In [18]:
```
comparison_df = pd.DataFrame([actual_values, predicted_values], index=
```

In [19]:
```
comparison_df.T.plot(kind='bar', figsize=(10, 6))
plt.title('Comparison of Actual Cadence Data and ML Predictions')
plt.ylabel('Power (µW)')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



APPENDIX D

Lab Report Evaluation Rubric

Course: ECE 240 Date: 12/17/2023

Evaluate each component using a weighted scale based on the following criteria:

FORMATTING RUBRIC

	Poor	**Excellent**	**Weight**	**Score**
Title Page and Table of Contents	Missing or does not Follow Standards	Fully Follows Guidelines and Standards	5	
General Format of Technical Paper	Does not Follow Guidelines and Standards	Fully Follows Guidelines and Standards	5	
Overall Structure of Technical Paper	Missing Sections or Information in Incorrect Locations	Appropriate Sections; Information in Correct Location	10	

WRITING RUBRIC

	Poor	**Excellent**	**Weight**	**Score**
Spelling and Grammar	Many Errors	Minor or No Errors	10	
Sentence Structure and Transitions	Poor Structure	Well Structured	10	
Focus and Organization	Unorganized and Lacks Clarity; Poor Presentation	Well Organized and Reads Clearly; Good Presentation	10	

TECHNICAL CONTENT RUBRIC

	Poor	**Excellent**	**Possible Points**	**Score**
Objectives and Theoretical Background	Meaningless Objectives; Terms Unrelated to Content	Objectives Clearly Stated; Theory Well Presented	10	
Experimental Procedure	Experimental Procedure Unclear and Incomplete	Experimental Procedure Valid and Complete	20	
Analysis and Technical Conclusions	Inconsistent with Observations or Objectives	Consistent with Observations and Objectives	20	

Overall average score _____