DigWise Technology Corporation, LTD.

GRO User Guide

MLD
2025/2/24

Table of Contents

1. Introduction

The semiconductor industry is facing unprecedented challenges, shifting from traditional yield improvement to comprehensive productivity optimization. To stand out in a highly competitive market, chip design must simultaneously focus on efficiency enhancement, cost reduction, and quality assurance. The current emphasis is no longer limited to yield alone but extends to cost-effectiveness, performance, yield enhancement, and overall market competitiveness.

Design-Technology Co-Optimization (DTCO) has been widely adopted in semiconductor physical design processes to improve chip production efficiency and competitiveness.

For ultra-efficient designs, during the design phase, process and standard cell library analysis are conducted to optimize the drive strength, area, and power consumption of key components within the chip architecture, ensuring overall reliability. Customized designs for critical components and the use of on-chip sensors allow the implementation of defensive strategies to effectively mitigate process variations and dynamic IR-drop, ensuring design robustness.

In the production phase, machine learning techniques are used to analyze actual process variations, enabling the formulation of timing re-characterization (Re-K) and signoff strategies. This approach provides in-depth insights into process uniformity and on-chip variation (OCV), achieving optimal design margins.

Figure 1 illustrates the implementation details of DTCO, covering pre-silicon preparation and post-silicon analysis & feedback, with a process optimization framework driven by data and machine learning.

- Pre-silicon preparation includes process analysis, SPICE and standard cell library analysis, feature extraction and modeling, critical path performance optimization, as well as area and power optimization.
- Post-silicon analysis encompasses the design and integration of on-chip monitoring circuits, testing, data correlation analysis, machine learning platform applications, automated binning strategies, and optimization feedback.

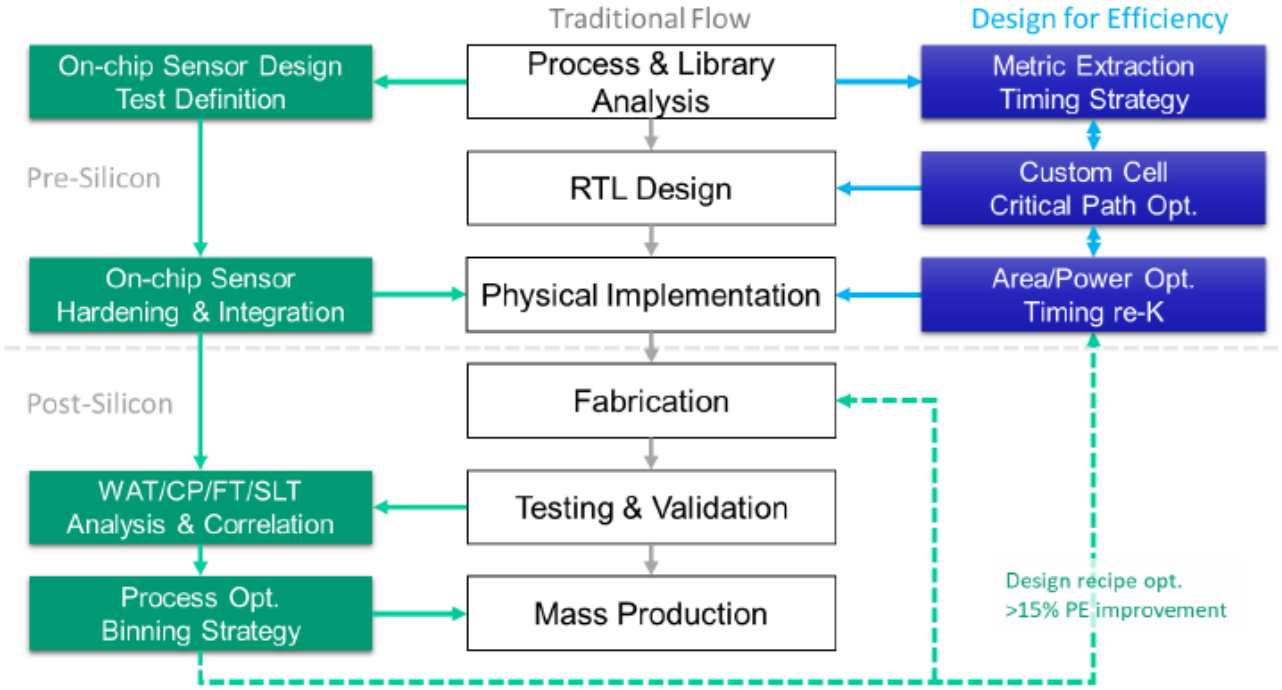

Fig. 1 DTCO Physical Design Flow

2. Why we need to Ring Oscillator

In the implementation of the DTCO process, embedding on-chip monitors is an essential step, with Ring Oscillators (RO) being one of the key components. The following section provides a detailed introduction to the applications of RO.

2.1. Goal-Oriented RO Design

In the physical design process, multiple stages are typically involved, such as front-end design constraints and F/V Shmoo evaluations, as well as back-end stages including pre-CTS, post-CTS, and post-route. By analyzing cell usage reports, we can observe that 80% of the contributions to area and power consumption usually come from just 20% of the critical components. These key components are often primary targets for S2S optimization. For example, in the SHA-3 algorithm, there is extensive use of XNR and XOR cells, along with cells utilized in the clock-tree structure.

Additionally, if N/P balance needs to be considered, specific structured components such as P-MOS stacking NOR and N-MOS stacking NAND can be introduced. As shown in Figure 2, the selection and optimization of these components can effectively meet performance requirements while reducing area and power overhead, thereby enhancing overall design efficiency.

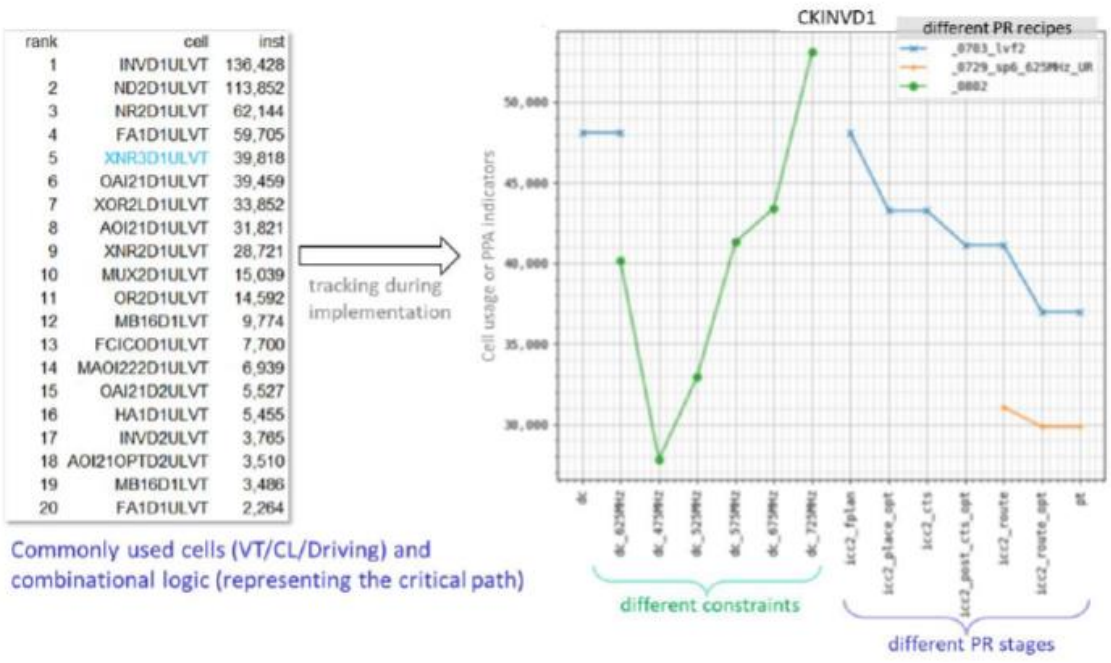

Fig. 2 Cell Usage and PPA Evaluation

2.2. SPICE-Silicon Correlation Analysis

Post-silicon measurement results can be accurately compared with SPICE simulation targets (SS, TT, FF) to determine the actual distribution under current process conditions and its deviation from SPICE targets, with precision reaching the picosecond level. As shown in Figure 3, by incorporating Lot ID or time-axis parameters, it is also possible to track the impact of process parameter adjustments on chip performance distribution and assess process control capability and stability.

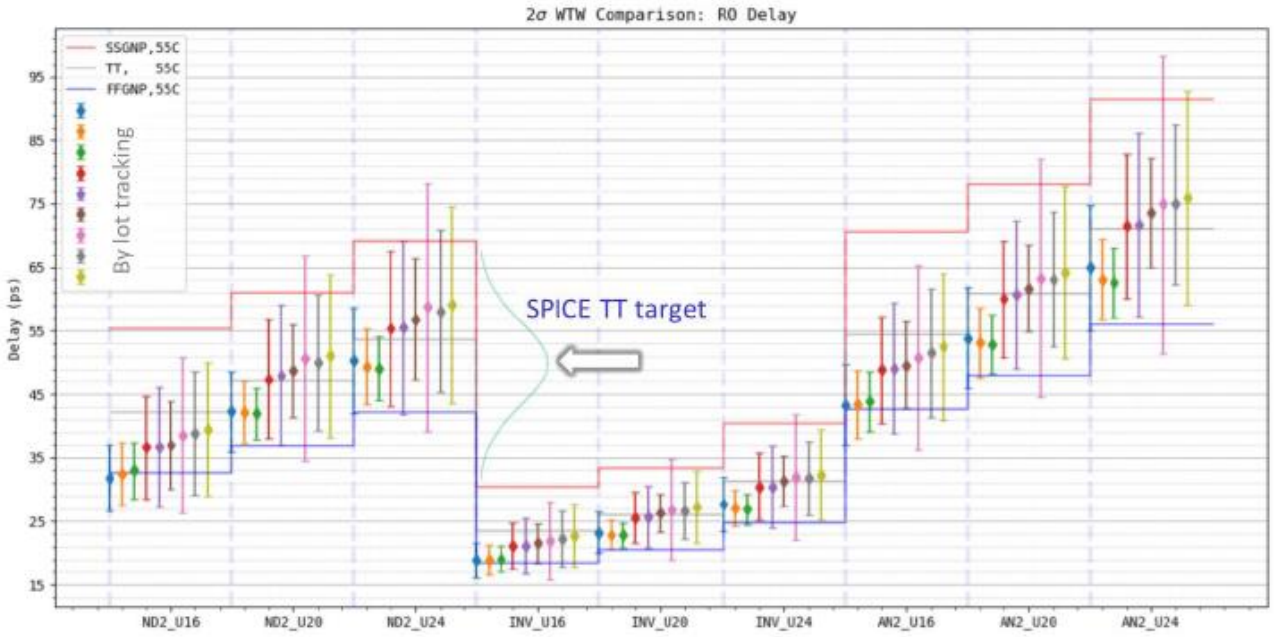

Fig. 3 RO SPICE-Silicon Correlation

In general, post-silicon chip measurement environments have limited flexibility due to cost constraints. Additionally, the inherent IR drop in the test environment may be difficult to fully control. At the same time, SPICE simulation targets cannot cover all possible scenarios. As shown in Figure 4, in such cases, a regression model built with a small or sparse set of simulation data can be used to predict conditions at non-simulated points. This approach helps align post-silicon measurement data with simulation results, enhancing both analysis accuracy and efficiency.

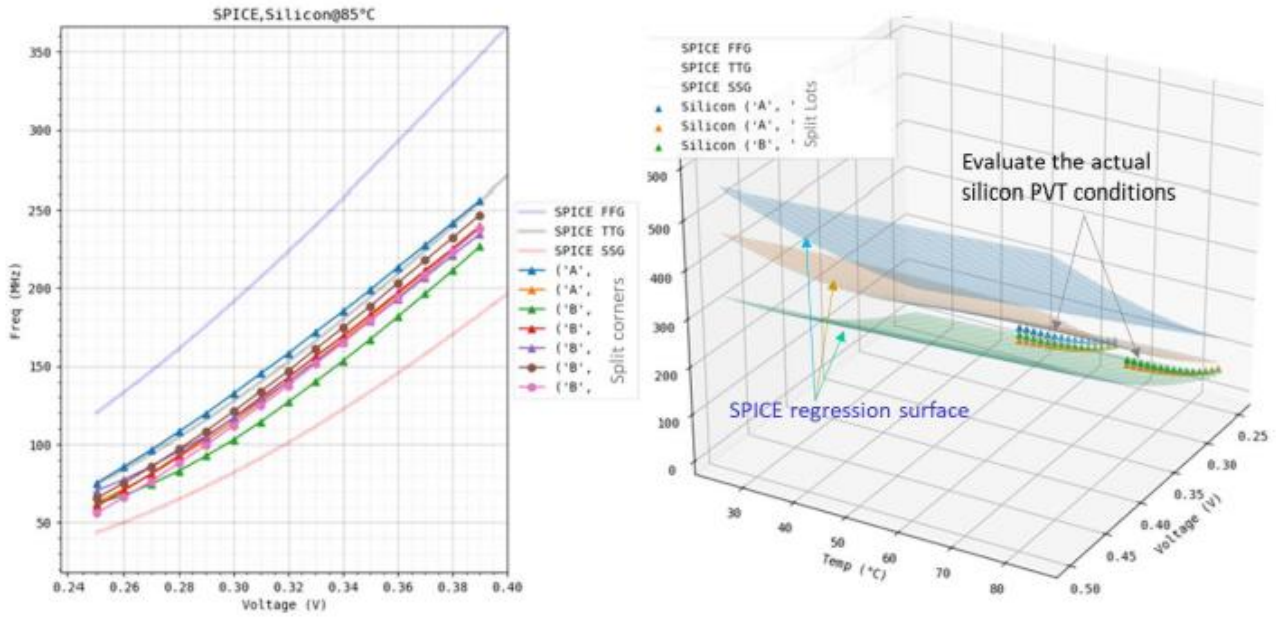

Fig. 4 V-F Shmoo and SPICE Correlation

2.3. Process Tracking and Adjustment

Process tracking and adjustment are critical to ensuring chip performance stability and yield improvement. By utilizing on-chip Ring Oscillators (ROs), real-time monitoring of process parameter variations, such as threshold voltage shifts (Vth shift) and leakage current (SIDD) changes, can be achieved. This enables a better understanding of how process variability impacts circuit performance, as illustrated in Figure 5.

Additionally, the high-sensitivity timing data provided by ROs effectively reflects process uniformity and local variations. Based on these monitoring data, engineers can adopt the following adjustment strategies:

- Process Offset Correction: Optimize key process parameters based on monitoring results to minimize the impact of process deviations.
- Functional Compensation: Leverage machine learning algorithms to dynamically adjust internal chip parameters to compensate for process inconsistencies, such as Dynamic Voltage and Frequency Scaling (DVFS).
- Long-term Trend Analysis: Accumulate RO data from different chip batches to identify long-term process variation trends, providing data-driven insights for process improvement and new designs.

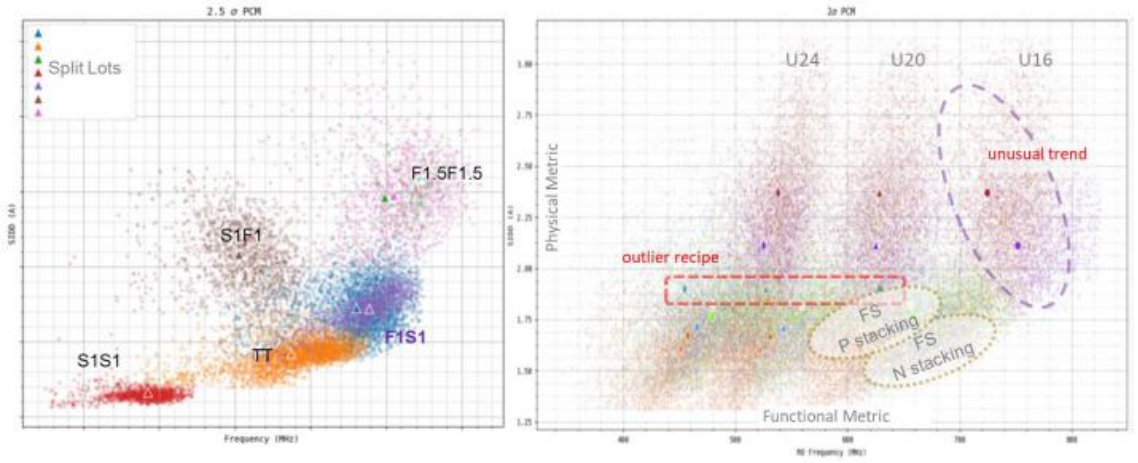

Fig. 5 Process Tuning

2.4. On-Chip Local Voltage Distribution Monitoring

If area overhead allows, denser ROs can be evenly distributed across the chip, as shown in Figure 6. By adjusting different voltage levels and measuring RO frequency (V-F sweep), it can be observed that within most operating ranges, voltage and RO frequency exhibit a linear relationship. The impact of process variation on the slope is minimal, with the primary difference reflected in the intercept. When the process is faster, the intercept is positioned higher; conversely, when the process is slower, the intercept is lower. With this well-defined V-F characteristic, differences in RO frequency can be translated into differences in effective voltage, accurately reflecting the relative effective voltage between ROs. However, it is important to note that this method can precisely determine relative voltage differences but cannot directly obtain absolute voltage values.

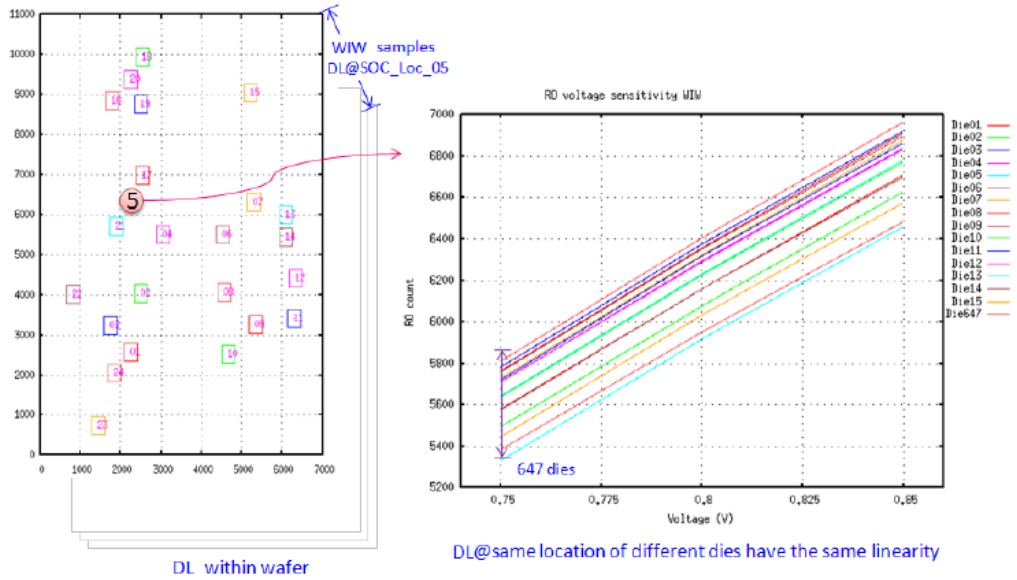

Fig. 6 On-chip Ring Oscillators (ROs) and V-F Curve

An interesting dataset reveals that when we back-calculate the relative voltage difference for each RO on individual chips, the RO frequency shows almost no correlation with the chip's coordinates. As illustrated in the figure, each color represents a different chip and the frequency distribution of its 24 internal ROs. However, when we average the data of approximately 700 chips from the same wafer based on their coordinates, we observe a strong correlation between the average value and coordinate position, as indicated by the bold black line in Figure 7, even surpassing the influence of process variation. This suggests that regardless of whether the chips originate from SS, TT, or FF wafers, their results tend to converge.

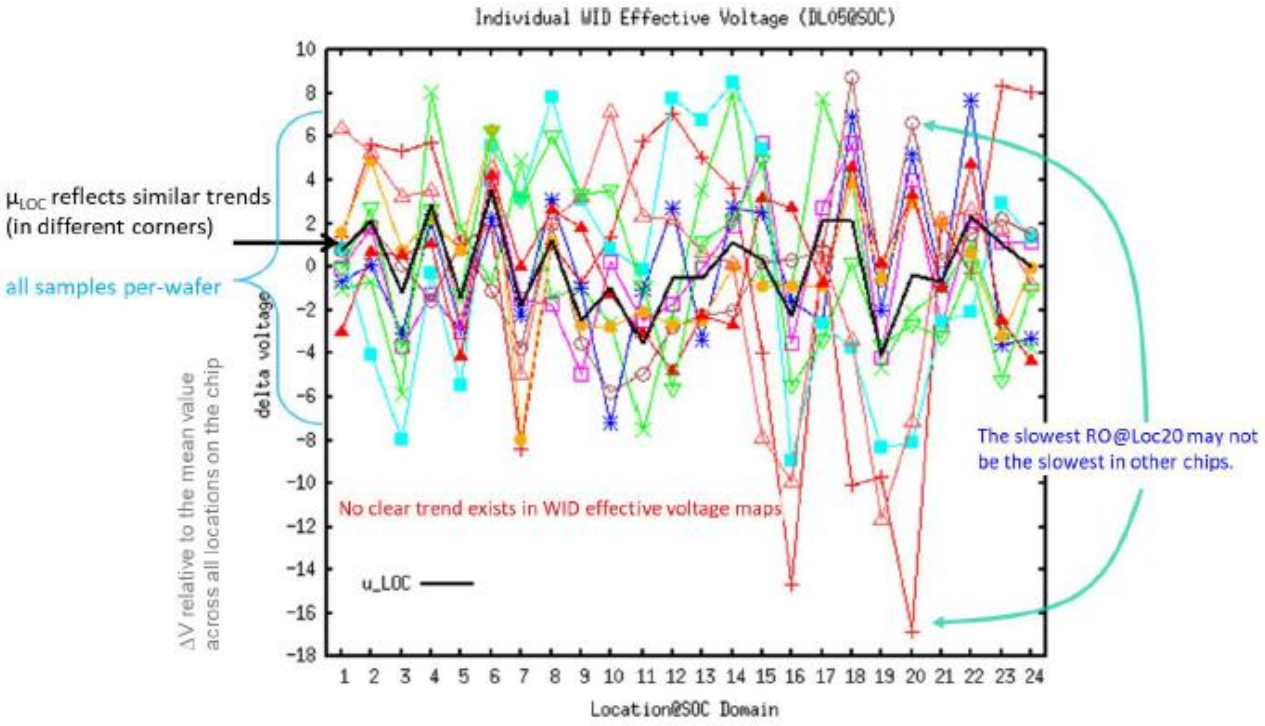

Fig. 7 On-chip RO Effective Voltage Analysis

By leveraging different RO data, we can derive the equivalent voltage at each detection point. As shown in Figure 8, measurement results reveal an interesting characteristic: when analyzing individual chips separately, the non-uniformity of equivalent voltage across different regions appears random, and the fluctuation pattern varies from chip to chip. However, through extensive experimental data analysis, we discovered that the mean value of ROs at the same locations within a chip follows a clear trend, without randomness. Moreover, even across wafers from extreme process corners, such as SS, TT, or FF, the observed trend remains consistent.

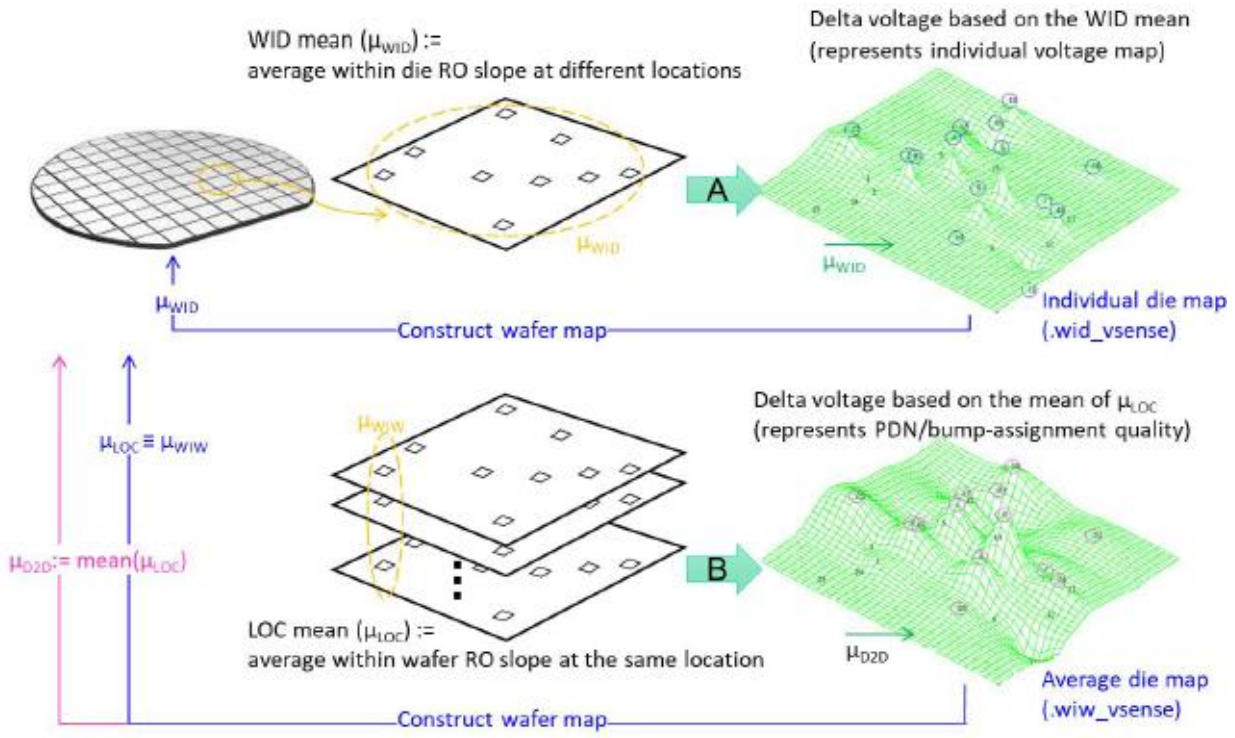

Fig. 8 On-chip Effective Voltage Regression

This reflects a non-random systemic issue and structural defect, involving factors such as metal density, layout effects, and PDN/bump distribution. These factors collectively contribute to overall structural non-uniformity, which must be considered during the design process. By applying a 34th-order surface regression to the discrete data points, we can generate a heatmap of effective voltage contours, as shown in Figure 9. Through analysis and diagnostics, we found that voltage differences are highly correlated with power bump distribution density, with an inherent voltage variation ranging from +6mV to -12mV (measured when the chip is powered on but not operating). Using a simple third-party ERA analysis tool, we can evenly overlay TAP current to detect potential issues early. In ERA, hotspots can be identified, which may later develop into low-voltage regions on the chip. This static effect is independent of the manufacturing process and exhibits no randomness, making it a critical factor that warrants greater attention in yield analysis. However, these issues cannot be effectively identified through the more complex dynamic IR analysis.

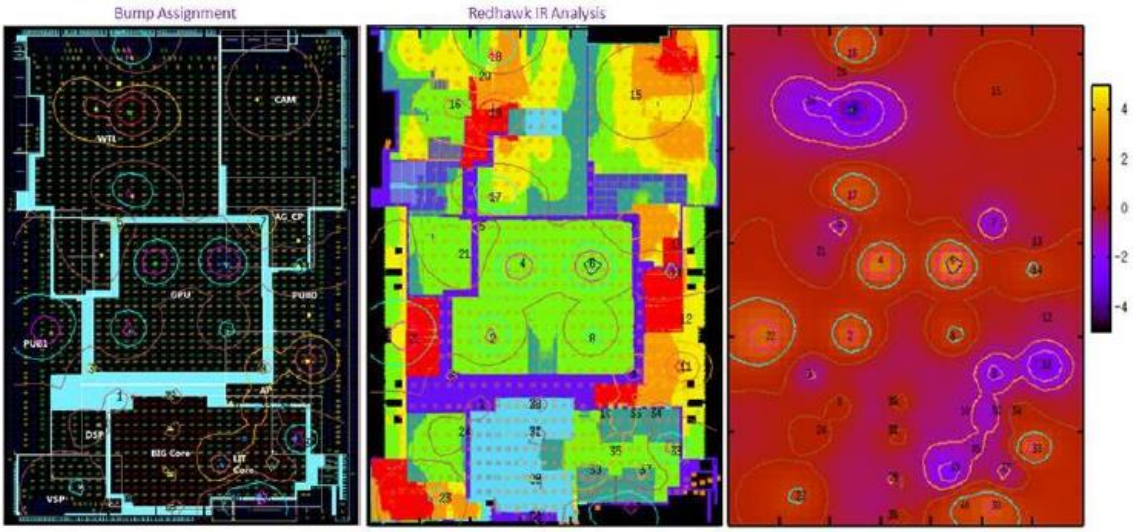

Fig. 9 Effective Voltage and ERA Correlation

2.5. Compensation Strategy Development

During the mass production process of chips, unexpected results often occur due to various factors. These may include inherent errors in SPICE models, incomplete physical design processes, overly aggressive or conservative timing signoff strategies, and poor process control such as parameter drift in mass production processes and insufficient wafer uniformity. As shown in Figure 10, if a significant adjustment to process parameters is made for compensation—such as shifting the target from TT to FF discrete points—it may meet speed specifications but could also result in an exponential increase in power consumption, making it difficult to control. Therefore, voltage compensation is a more predictable and cost-effective approach. Whether static or dynamic compensation is used, it can more effectively balance performance and power consumption.

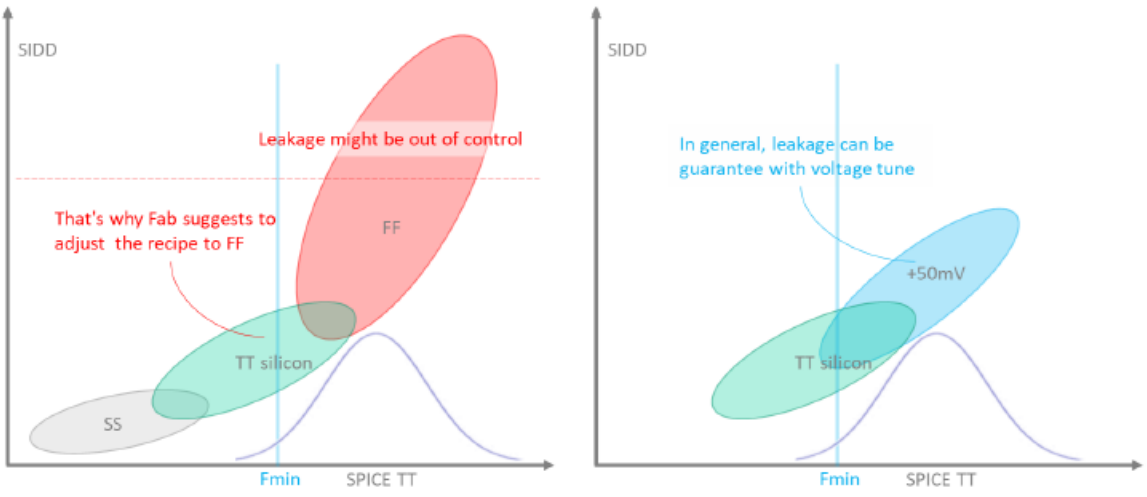

Fig. 10 Process and Voltage Tuning

However, real-world scenarios are often not as intuitive as we might expect. In reality, process parameter deviations are neither random nor Gaussian-distributed; instead, they are often accompanied by distinct mechanical harmonics (harmonic spin), which become a major obstacle to design margin accuracy. These non-random effects, when compounded with process and test environment deviations, ultimately introduce an element of randomness. This inherent non-randomness renders seemingly robust defensive strategies unreliable in practice. Figure 11 illustrates various physical characteristics of a wafer, such as RO frequency or SIDD, and their trends as voltage or temperature changes. It can be observed that local non-uniformities cannot be fully resolved through voltage compensation alone, and the performance gradient across the chip does not disappear with voltage compensation.

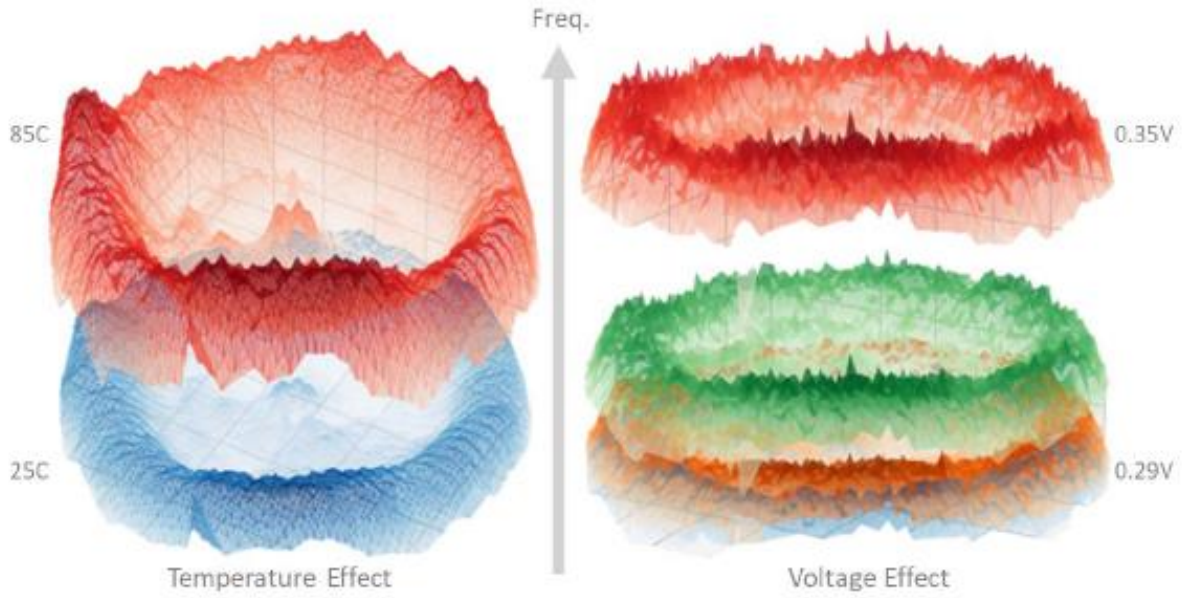

Fig. 11 RO Frequency vs. Temperature and Voltage Effects

In advanced chip design, integrating on-chip monitors (OCM) and machine learning-based compensation mechanisms has become a key strategy for enhancing performance and reliability. On-chip monitors can continuously collect real-time operating parameters, such as voltage, frequency, and temperature, providing high-precision status information as the foundation for static or dynamic compensation. As shown in Figure 12, a multi-level compensation strategy is recommended, including: Process parameter optimization, Chip binning strategy, Voltage-frequency configuration and Real-time dynamic adjustments of voltage and frequency. This comprehensive approach improves chip performance and ensures system stability.

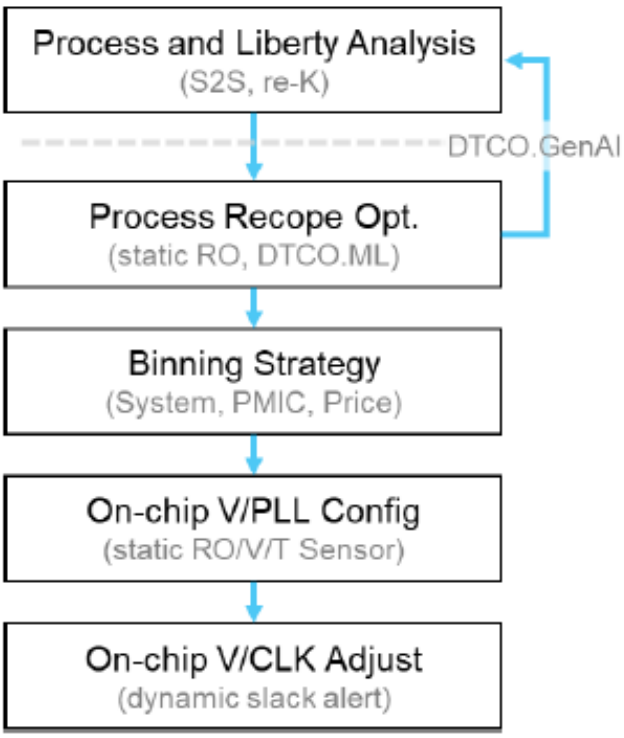

Fig. 12 Multi-level Compensation Strategy

2.6. OCV Analysis

Each chip requires only one RO, which, through machine learning, can be used to indirectly estimate wafer uniformity, on-chip equivalent voltage differences, and On-Chip Variation (OCV). During the CP testing process, the die-to-die RO differences can be calculated. As shown in Figure 13, the distance between ROs at the same coordinates on two different wafers corresponds to one die's distance. By gradually increasing the distance between the two wafers and computing the RO differences at the same coordinates, followed by regression analysis, the OCV at D0 (when there is no distance between on-chip components) can be estimated.

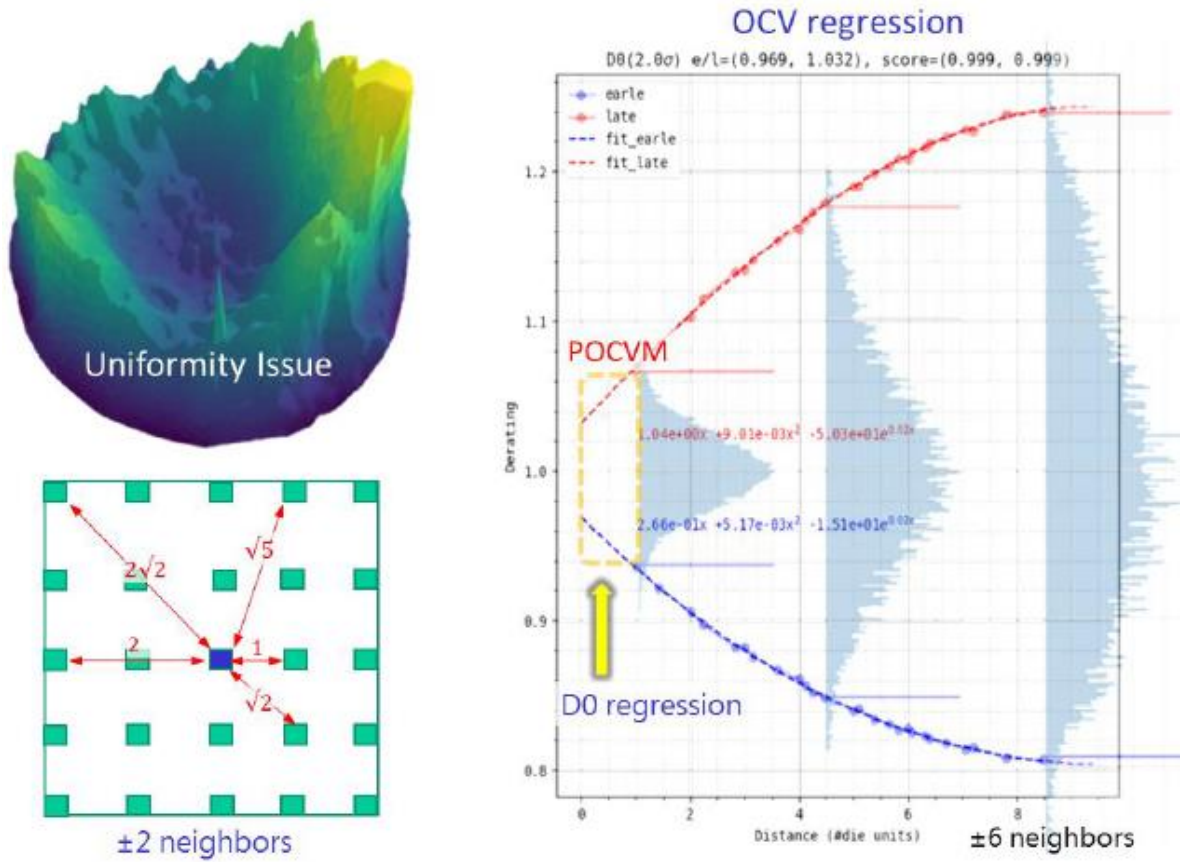

Fig. 13 On-chip Sensor and OCV Regression

3. GRO Automation Tool and Verification Process

In the core objectives of DTCO, on chip embedded monitoring circuits play a crucial role. With on-chip measurement data available, the design process illustrated in Figure 14 can be executed.

The purpose of the GRO tool is to provide the industry with a reliable and convenient solution by automatically generating essential baseline circuits. By integrating data analysis and machine learning with test results from eFuse, WAT/CP, and Aging, numerous scalable application scenarios can be explored in the future, such as binning strategies and voltage/frequency compensation, making them valuable areas for further planning and development.

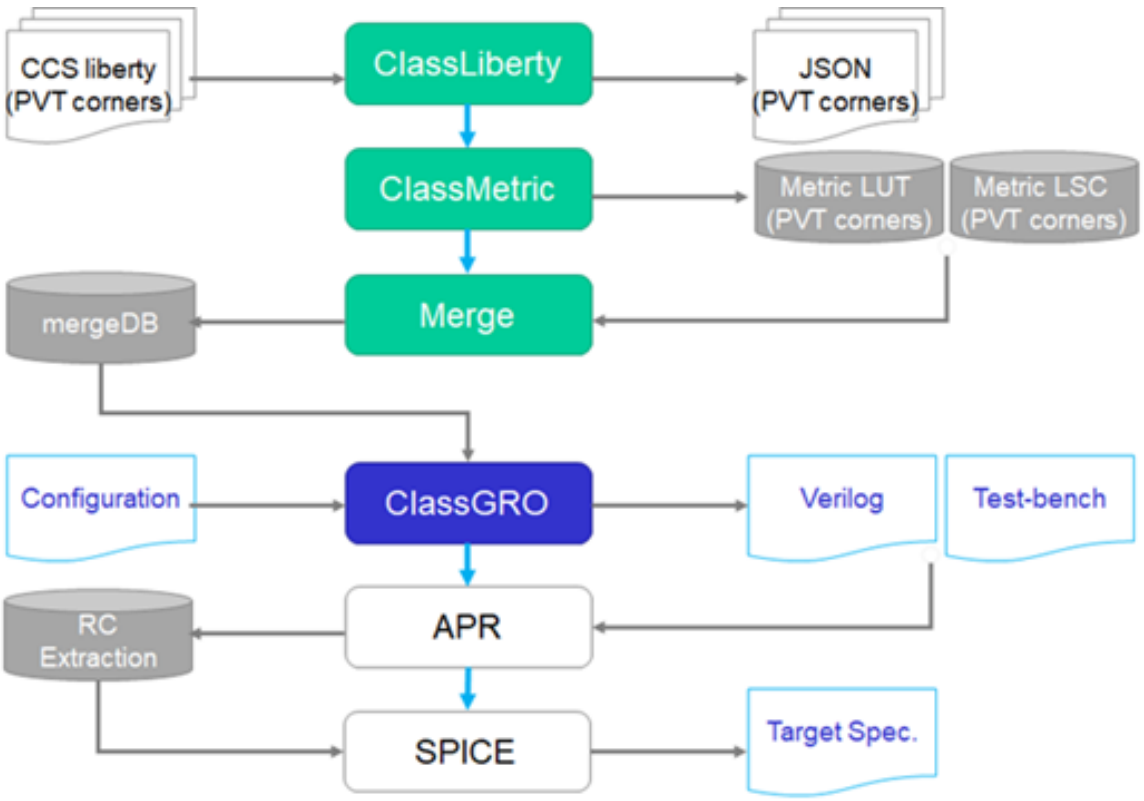

Fig. 14 Program Execution Flow

In the future, the concept of embedding monitoring circuits within chips will become increasingly widespread. This tool is designed to provide industry with a reliable and convenient solution by automatically generating the essential foundational circuits. As shown in the program execution flow in Fig. 15, GRO plays a significant role in the overall circuit design process.

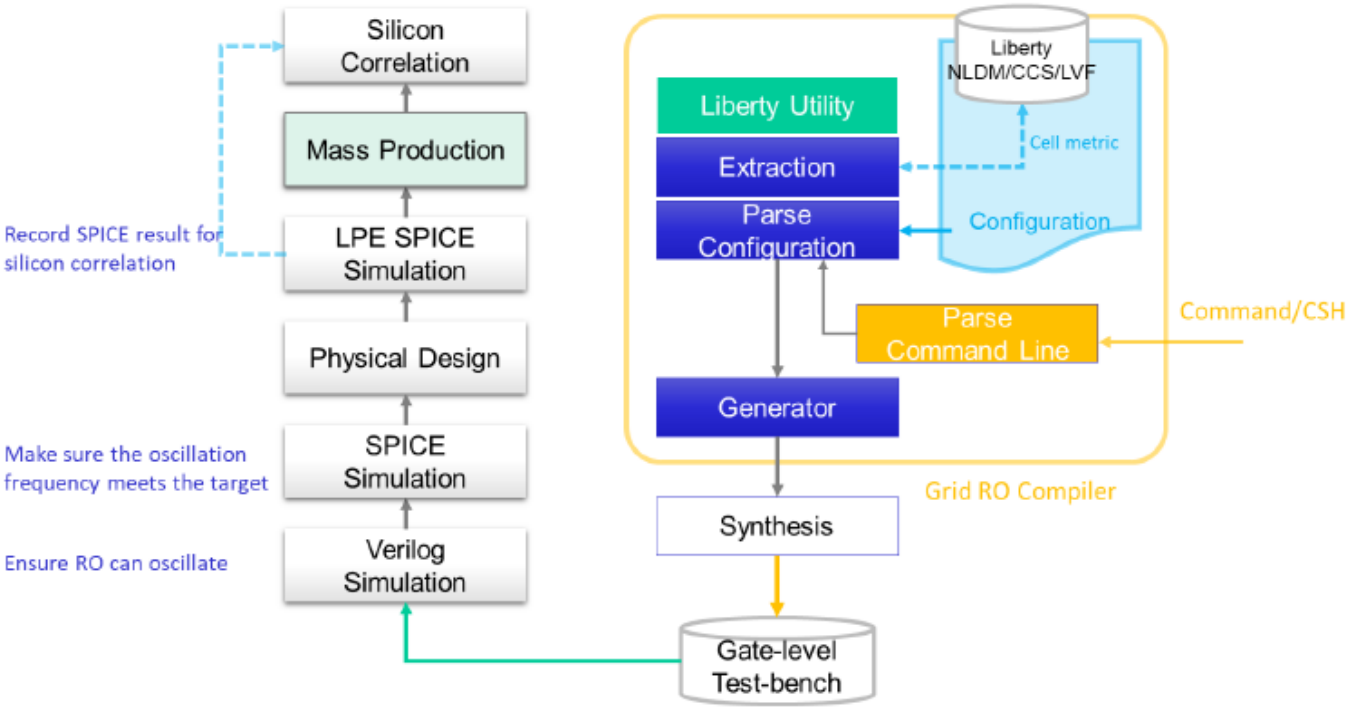

Fig. 15 Program Execution Flow

4. Ring Oscillator Architecture

A RO is a circuit composed of an odd number of inverters connected in series. By utilizing the characteristic of inverters to invert signals, the output generates an inverted square wave signal. As the signal propagates through this ring circuit, there is a time difference, or time delay, between the input and output square wave signals caused by passing through an inverters, as shown in Fig. 16.

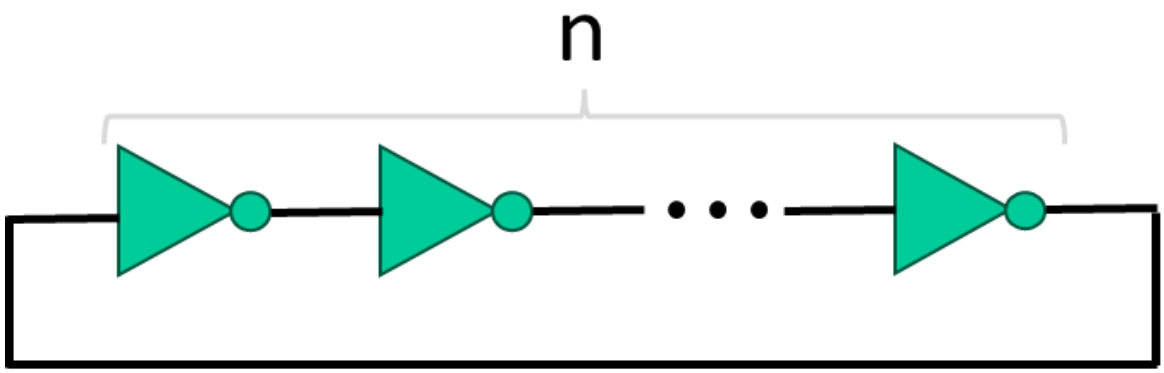

Fig. 16 Ring Oscillators.

The complete architecture of the RO is shown in Fig. 17. At the front end, an enGate serves as the switch for the RO to ensure that the circuit can generate an oscillating clock signal. At the back end, a Ripple Counter is used to count the number of oscillations.

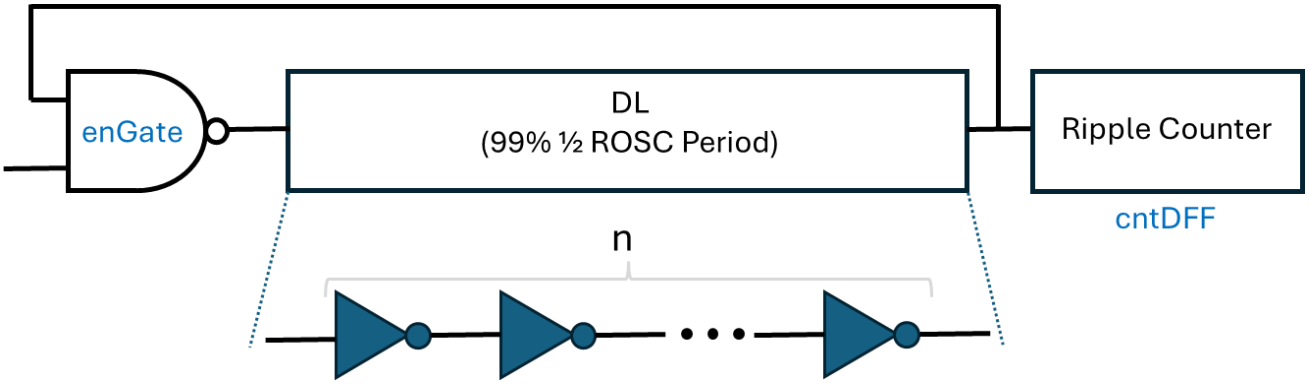

Fig. 17 Complete Architecture of Ring Oscillators.

Fig. 18 shows the basic composition of a common Ring Oscillator (RO) circuit. The input is controlled by a NAND gate with a REN pin as the switch, followed by multiple stages of delay elements. A multi-bit counter records the output frequency. If the purpose of the RO is to compare with the SPICE model of component delay, it is recommended that the delay elements account for 99% of the RO cycle to minimize the variability introduced by the control gate.

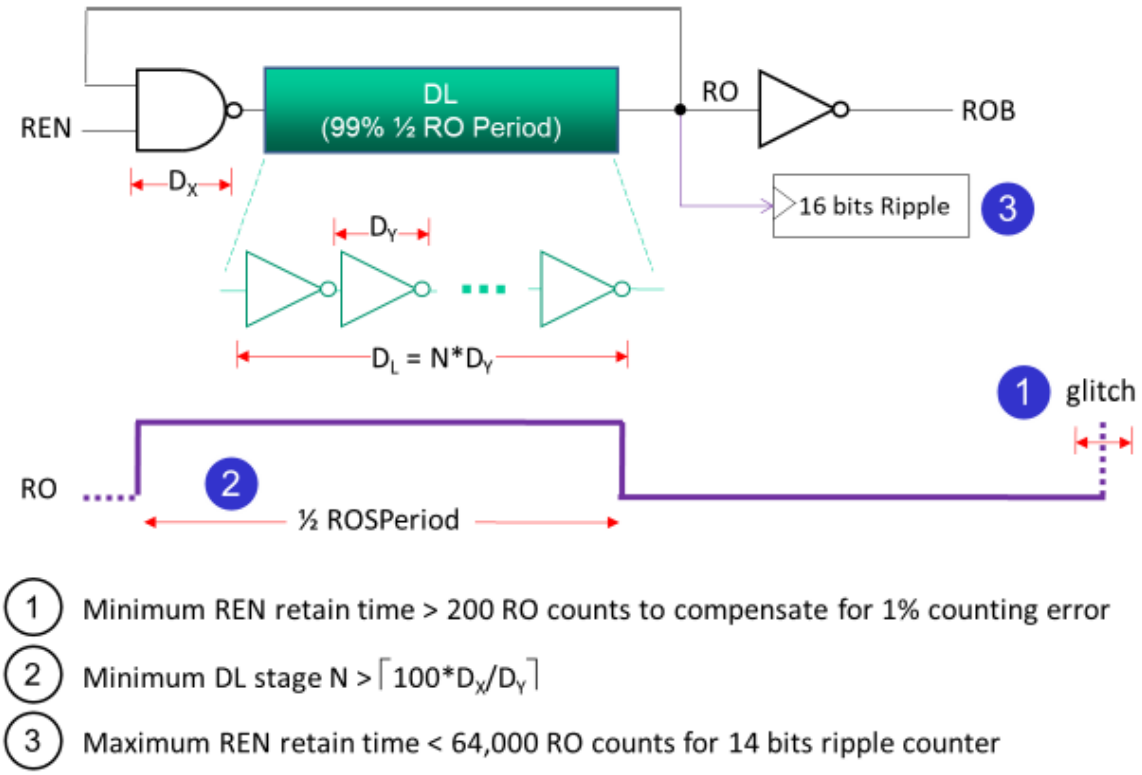

Fig. 18 RO Design Guideline

5. Automating Ring Oscillator Design with GRO

During the design process of Ring Oscillators (ROs), traditional methods often require engineers to perform tedious manual configurations, which are not only time-consuming but also prone to errors. To address these challenges, we developed GRO (Generalized Ring Oscillator), an efficient tool dedicated to automating the design of ROs.

The core goal of GRO is to simplify the RO design process, enabling users to complete the entire workflow from design to testing with less time and effort. With its intuitive interface and powerful automation features, GRO provides a comprehensive solution:

- **Automated Design Generation**: Users only need to input basic design requirements (such as enGate, invGate, cntDFF, and target delay), and GRO automatically generates the corresponding RO structure, eliminating the need for manual parameter adjustments.
- **Support for Multi-Corner Simulations**: GRO supports simulations under various process, voltage, and temperature (PVT) conditions, helping users quickly analyze the performance of ROs and output detailed test results.
- **Efficient Configuration and Export**: With GRO's one-click operation, users can easily complete the RO design process and export related files, such as SPICE netlists, test platform scripts, and clock signal configurations.
- **Error Risk Reduction**: GRO includes a built-in error-checking mechanism that provides instant feedback on potential issues, ensuring every step of the design process adheres to standards and avoiding common errors associated with manual design.

The value of GRO lies not only in saving time but also in improving the reliability and efficiency of the design. With this tool, engineers can focus more on high-level circuit optimization rather than tedious foundational design tasks. Whether for beginners or experienced designers, GRO offers significant convenience and benefits.

6. Core Features

- Through the intuitive UI interface, professionals from any field can easily get started.
- Complete all pre-deployment tasks quickly without requiring any programming background.
- Automatically generate configuration files.
- User-friendly options for configuring delay line settings to suit individual preferences.

7. Getting Started

7.1. Start executable file

Before using the tool, users must ensure that the **"GRO_GUI"** and **"GRO"** executable files are located in the same directory. Once this is confirmed, you can proceed to use the tool by following the steps outlined below.

7.2. User Interface Overview

In this tool, all functionalities are distributed across the **"Create .f"** and **"RO Compiler"** pages. This section will explain the operations of these two pages in sequence , as shown in Fig. 19.

- *Create .f Panel*:
  Based on the user's specified lib file, enGate, invGate, cntDFF, and delay line settings, the tool automatically generates a config file.
- *RO Compiler Panel*:
  Based on the user's selected Config file and Output Path, the tool automatically generates the makefile, as well as folders required for the workflow, including spice, synthesis, testbench, and verilog.

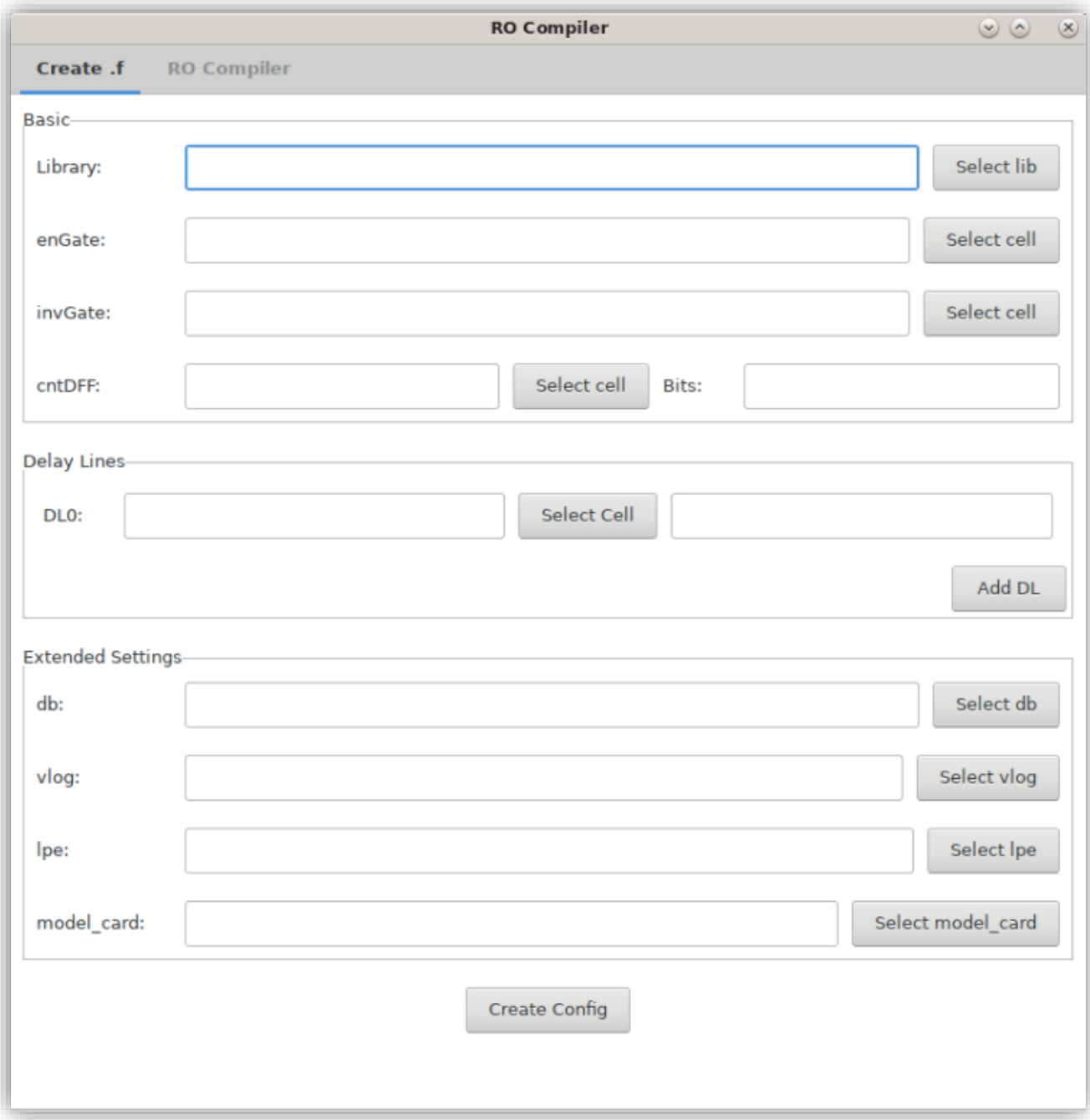

Fig. 19 GRO Main Window.

8. Usage Flow Explanation

8.1. Configuration File

1. Select lib file:
   - Click **"Select lib"** and choose the desired lib file.
2. Configure enGate:
   - Click **"Select cell"** and choose the cell to be used as enGate.
3. Configure invGate:
   - Click **"Select cell"** and choose the cell to be used as invGate.
4. Configure cntDFF:
   - Click **"Select cell"** and choose the cell to be used for cntDFF, then enter the desired bits value in the Bits field.
5. Select db file:
   - Click **"Select db"** and choose the desired db file.
6. Select vlog file:
   - Click **"Select vlog"** and choose the desired vlog file.
7. Select lpe file:
   - Click **"Select lpe"** and choose the desired lpe file.
8. Select model_card file:
   - Click **"Select model_card"** and choose the desired model_card file.
9. Configure DL:
   - Click **"Select Cell"** to choose the cell for building the delay line, then enter the desired delay (ps) in the adjacent field.
   - If multiple delay lines need to be configured, click the **"Add DL"** button below. The UI will generate new configuration fields for users to set up.
10. Create Configuration file:
    - After completing the above settings, click the **"Create Config"** button. In the pop-up window, select the path to save the config file to complete the config file creation process.

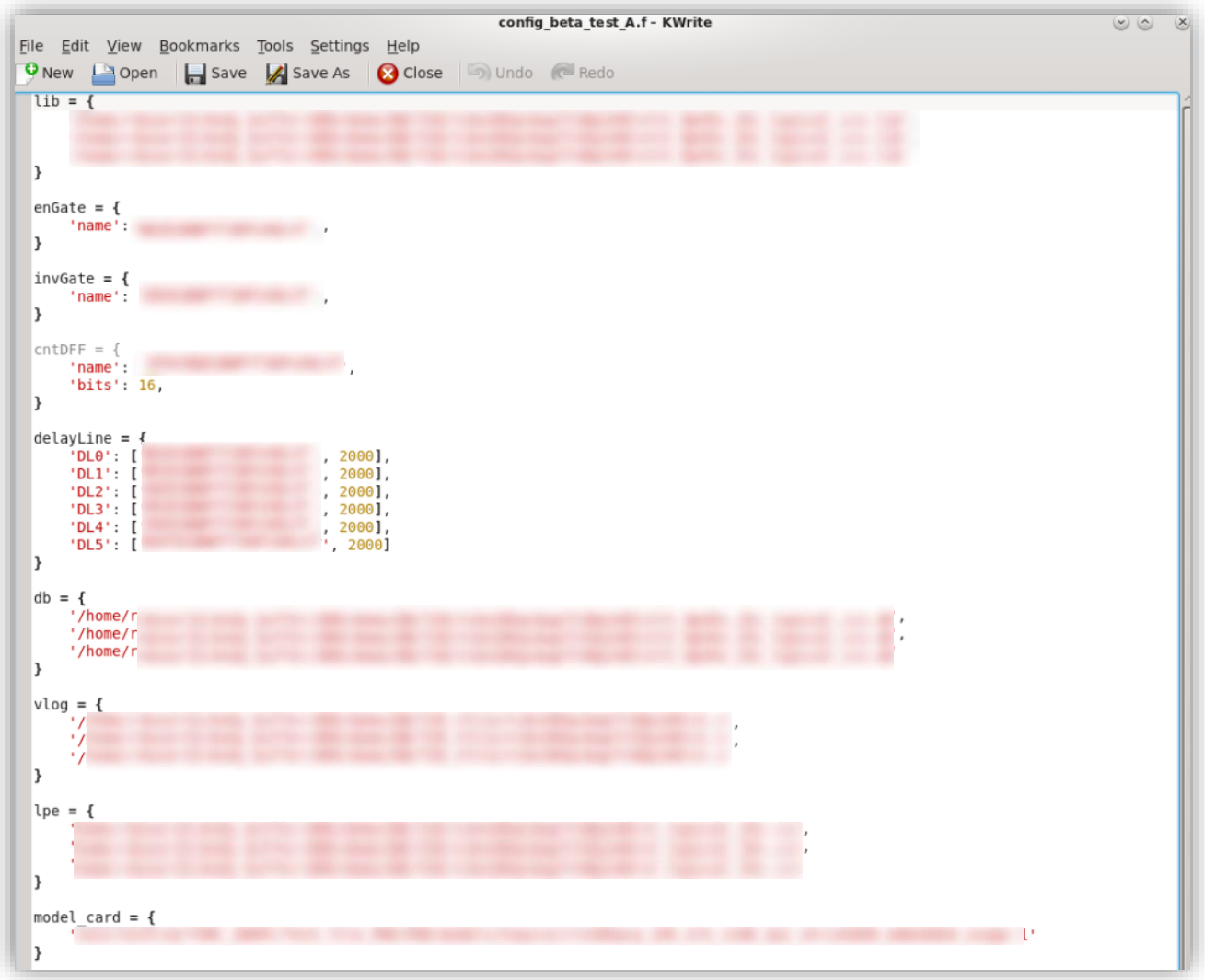

Fig. 20 Configuration File Generated by GRO

8.2. RO Compiler

1. Select Config file:
   - Click **"Select Config"** and choose the config file you want to use.
2. Select output path:
   - Click **"Select Output Path"** and choose the file path for the output.
3. Check the actions:
   - **Initialize Project :** Create and initiate an RO project directory only.
   - **Initialize Libraries :** Convert liberty NLDM/CCS (defined in the config) to JSON DB.
   - **Build RO :** Generate RO design and run synthesis, gate-sim, and spice flows.
4. Run Command:
   - Click **"Run Command"** to start executing the RO compiler.

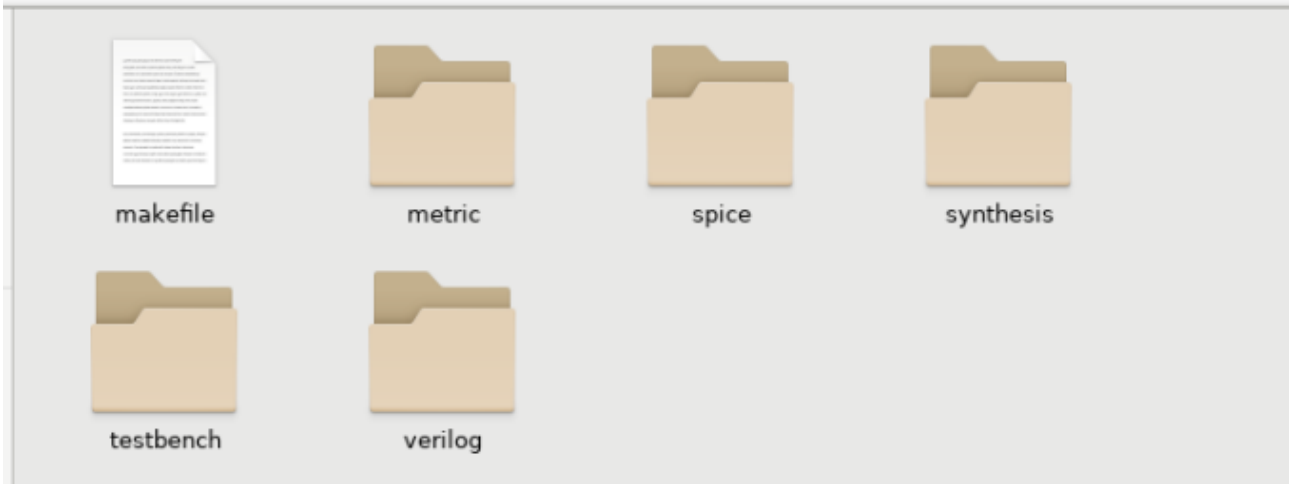

Fig. 21 Configuration Settings Automatically Generated by GRO Compiler