
Sergey Y. Yurish
Editor
Advances in Microelectronics
Volume 4
IFSA

# Chapter 1
# ASIC Pre-silicon Qualification Methodology

**Ang Boon Chong, Ong Sze Wei, Phang Eng Hong, Loh Jih Keat, Quek Li Chuang and Lee Chia Cheng**

### List of Abbreviations

**ASIC -** Application Specific Integrated Circuit
**DOA -** Death On Arrival
**DRC -** Design Rule Check
**EDA -** Electronic Design Automation
**Fmax -** Maximum Frequency
**HIP -** Hard Intellectual Property
**IP -** Intellectual Property
**KGB -** Known Good Base
**LPV -** Low Power Verification
**PPA -** Performance Power Area
**QOR -** Quality Of Result
**RM -** Reference Methodology
**TAT -** Turn Around Time
**TFM -** Tool Flow Methodology
**THS -** Total Hold Slack
**TNS -** Total Negative Slack
**WHS -** Worst Hold Slack
**WNS -** Worst Negative Slack

## 1.1. Introduction

For ASIC pre-silicon qualification, the key components and type of checks is shown in Fig. 1.1. From Fig. 1.1, it is shown that ASIC pre-silicon qualification has 4 components and the validation is done through implementation regression and sign-off regression.

Ang Boon Chong
Intel, 11900 Penang, Malaysia

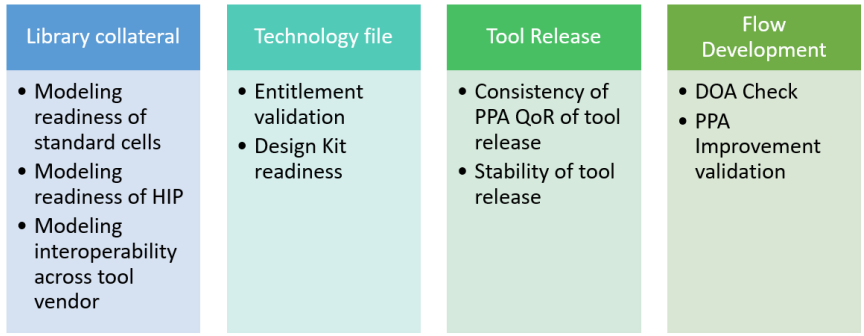


**Fig. 1.1.** ASIC Pre-Silicon Qualification Components.

In recent years, as electronic components account for 40 % of the cost of a vehicle, ISO 26262 has been adopted as the functional safety standard for the automobile industry. As illustrated in Fig. 1.2, ISO 26262 emphasizes the need to strengthen the confidence of use for all SOC development cycle tools.

| iso 26262-8 Tables 4 and 5 | | TCL1 | TCL2 | | | | TCL3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Methods | ISO 26262-8 Reference | Needs no qualification methods | ASIL | | | | ASIL | | | |
| | | | A | B | C | D | A | B | C | D |
| 1a Increased confidence for use | 11.4.7 | | ++ | ++ | ++ | + | ++ | ++ | + | + |
| 1b Evaluation of the tool development process | 11.4.8 | | ++ | ++ | ++ | + | ++ | ++ | + | + |
| 1c Validation of the software tool | 11I4.9 | | + | + | + | ++ | + | + | += | ++ |
| 1d Development in accordance with a safety standard | * | | + | + | + | ++ | + | + | += | ++ |

++ Indicates that the method is highly recommended for the identified ASIL
+ Indicates that the method is recommended for the identified ASIL

NOTE: * No safety standard is fully applicable to the development of software tools.
Instead, a relevant subset of requirements of the safety standard can be selected.
Example: Development of the software tool in accordance with ISO 26262, IEC 61508 or RTCA DO-178

**Fig. 1.2.** ISO 26262 EDA Tools Requirement [9].

For the current research of EDA tool, it is summarized as in Table 1.1 below.

To comprehend the quality of the EDA tool release, users may reference to the release notes, to understand the type of bugs fixes. However, the bugs fixed in the EDA tool release only imply two aspects:

- The rate of vendor bug fixes;
- History of the bugs exist in older version.

The rate of bugs fix is never a quality assurance indicator for the current tool release. Users can only hope that EDA vendor has done sufficient qualification prior release to

customers. Fig. 1.3 depicts the rate of tool bug fixes by year and release cadence. Typically, the current year's release will inherit the majority of bugs from the previous release. Higher bug fixes on a particular tool version, may be an indicator of the user adoption, on a particular tool version as well.

**Table 1.1.** EDA Tool Research Topic [1-17].

| EDA Developments | Years |
|---|---|
| Machine learning feature | 2021,2018,2017,2023 |
| Tool flow validation methodology | 2021 |
| New tool initiative | 2021,2020 |
| EDA tool safety standard | 2021,2017,2015 |
| Bug , validation, qualification | 2021,2011,2009,2002,2000 |

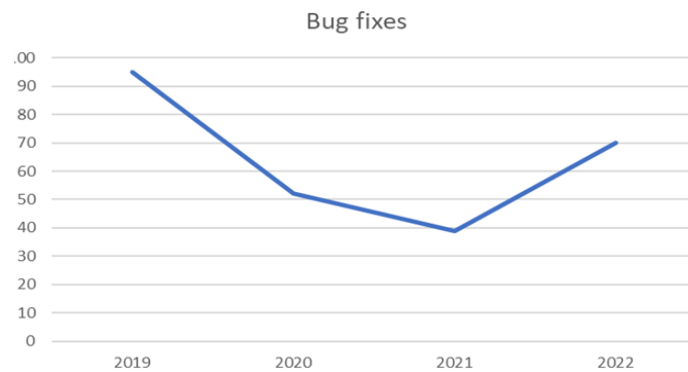


**Fig. 1.3.** Tool Bug Fixes Rate by Release Year, per Release Cadence [16].

For ISO 26262 related issues, it is currently adopted by the EDA vendors, as part of the release. The status is still no known issue but does not imply zero risk. Users are encouraged to regularly check online. The ISO 26262 release of snippet from EDA is shown in Fig. 1.4.

**Title**

Fusion Compiler Master List of Safety-Related Issues

**Description**

This page lists all known safety-related defects for the Fusion Compiler tool.

There are currently no known safety-related STARs for the Fusion Compiler tool. Please check this page regularly to ensure that you get the latest information about safety-related issues.

See also the Fusion Compiler Functional Safety Manual and the ISO 26262 Documentation - Synopsys Design Platform article.

**Fig. 1.4.** ISO 26262 Release by EDA Vendor [16].

For most design houses where focus is on speed of development with limited resources, they may opt to reference methodology download from EDA tool vendor as well as flow recommendation from foundry for particular node. For the type of checks in tool qualification, it typically covers the following scenarios:

- Death on arrival check;
- Limited basic quality check.

Besides the DOA check and limited basic quality check, EDA vendor will conduct repeatability or noise quality check, where it focuses on repeatability of the result with design input collateral remains unchanged. The EDA vendor noise regression [14] focus on the following effects:

- Random initial ordering;
- Random seeds;
- Random naming.

This chapter will focus on internal qualification on the EDA tool prior to production deployment, as well as the additional proposed stability methodology in the tool qualification. The implementation tool flow development for various technology will be used as reference for the discussion. The same concept can apply to other area of design development such as functional verification development and design co-simulation flow development. For library characterization and pre-silicon to post silicon correlation methodology, it is beyond the scope of this chapter discussion.

## 1.2. Tool Flow Qualification Methodology

For ASIC pre-silicon qualification, it covers the following pillars in sequence:

- Function check;
- Integration death on arrival check;
- Basic quality matrix check;
- Tool stability quality.

The tool flow qualification pillar is illustrated in Fig. 1.5. For functional check and integration check, it focuses on the features for project deployment and overall script integration as one reference flow for deployment.

### 1.2.1. Functional Check

Functional check also referred as unit or feature testing in the flow development. Functional check is important as for complete tool flow development testing, it may take up to weeks of development and full flow check does not validate the feature exhaustively due to runtime concern as it involves many features. Hence, for functional check, it is

standalone test-case to validate the new feature implementation exhaustively, prior integration into full flow. Example of the functional check are custom clock tree synthesis recipe [20] and machine learning library pruning [21] as shown in Fig. 1.6 and Fig. 1.7. For functional check, developer is expected to create its own dedicate test-case to verify the functional of the feature implemented.

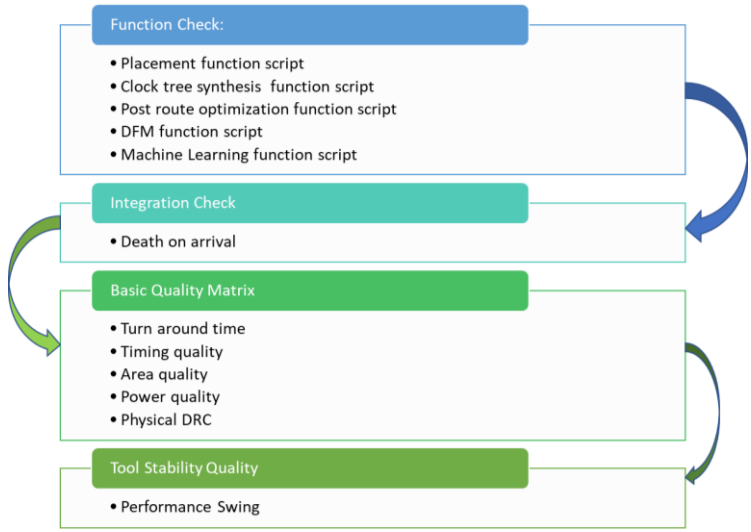


**Fig. 1.5.** Tool Qualification Pillars [22].

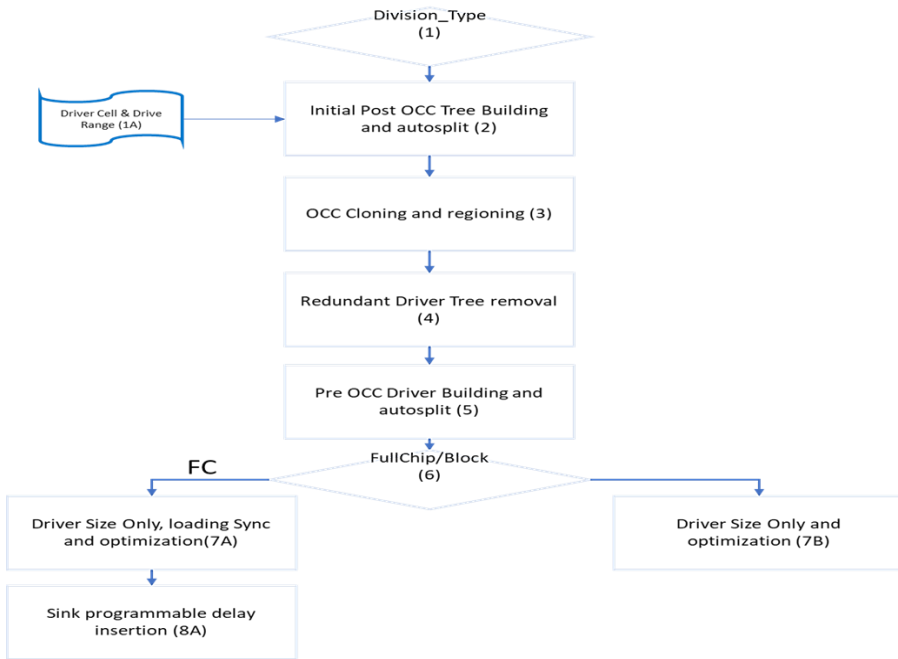


**Fig. 1.6.** Custom Clock Tree Synthesis [20].

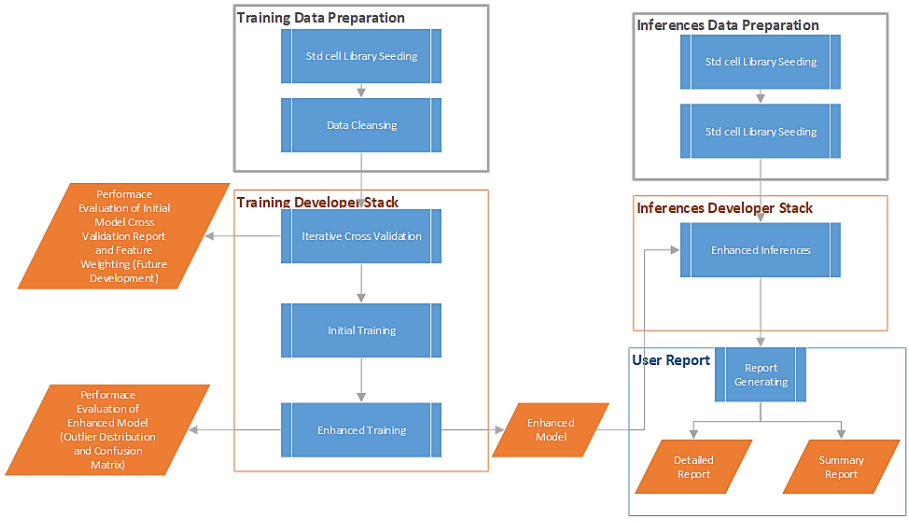


**Fig. 1.7.** Machine Learning Library Pruning [21].

### 1.2.2. Integration Check

Integration check is done after the functional check. For integration check, the priority focus is DOA prevention. Integration check is done prior reference flow release from development team or EDA vendor. Example of the design reference flow release is shown in Fig. 1.8. Based on the reference flow given, User just need to download the reference script from the EDA tool vendor, then configured the required variables, before kicking off the project deployment. Example of flow deployment is shown in Fig. 1.9 below.

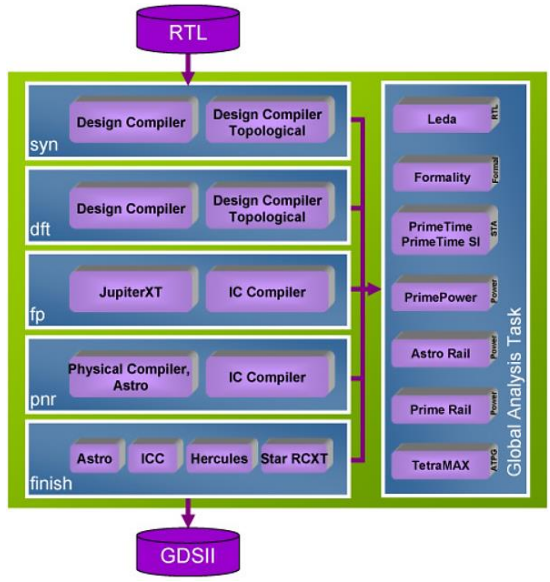


**Fig. 1.8.** ASIC Reference Flow [18].

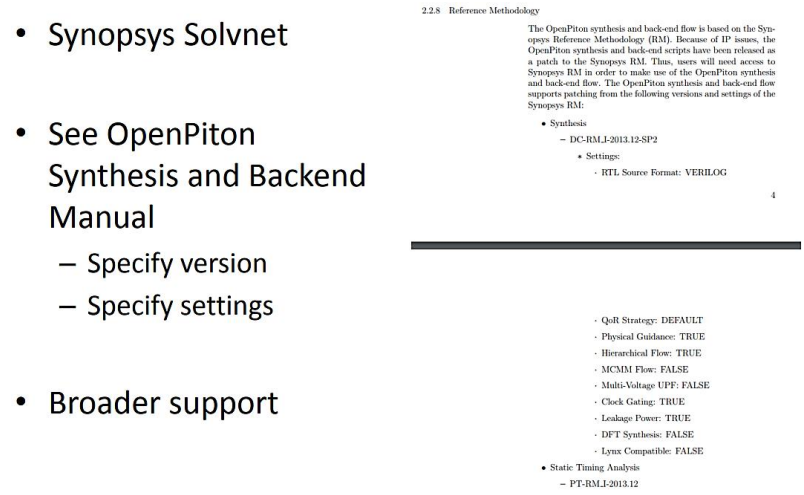


**Fig. 1.9.** Field RM Flow Deployment Example [19].

### 1.2.3. Basic Quality Matrix Check

For basic quality matrix check, the standardized reference designs are used to evaluate the flow from performance, power, area (PPA) and turnaround time (TAT). For timing quality check, it involved worst negative slack (WNS), worst hold slack (WHS), total negative slack (TNS), total hold slack (THS), timing design rule check (DRC) and crosstalk noise. To enable the basic quality matrix check, a good base reference is required. Classical method for establishing base reference run is the multiple frequency regression. The optimal frequency can be derived by plotting the target frequency against frequency achieved or target frequency against total negative slack. The spread of frequency sweep is typically 50 MHz or 100 MHz apart, so that the peak achieved frequency or spike of total negative slack can be clearly visible from the graph plotting. The base frequency also can be scaled from older technology design to estimate the possible base frequency. The automated design flow for basic quality matrix check is illustrated in Fig. 1.10.

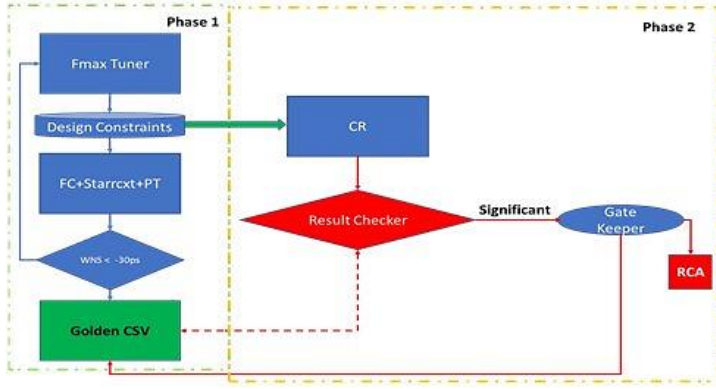


**Fig. 1.10.** Automated Basic Quality Matrix Check Flow [2].

The details flow of Fmax tuner illustrated in Fig. 1.10 is shown in Fig. 1.11.

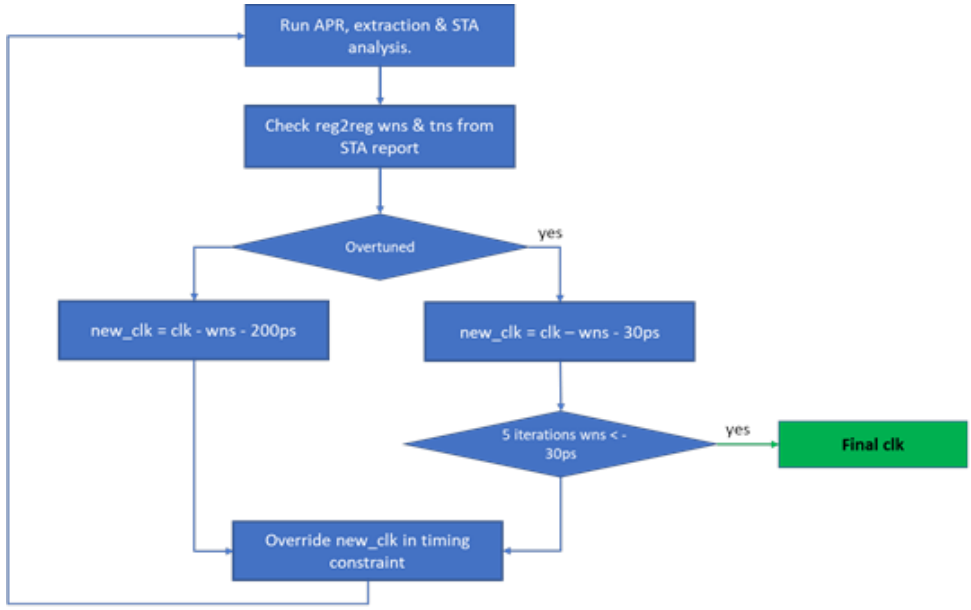


**Fig. 1.11.** Fmax Turner [2].

### 1.2.4. Stability Quality Check

For stability check, it focuses on repeatability. For most design development, the stability check only targets to sign-off tool development. For example, in timing sign-off, correlation is done against the spice simulation for 1000 critical paths as well as timing quality validation against older tool version release, for any outliers review. Similarly, for physical verification sign-off, it will validate of the physical design rule check (DRC) against existing tape out DRC report. The stability check qualification can be summarized as in Fig. 1.12 and Fig. 1.13.

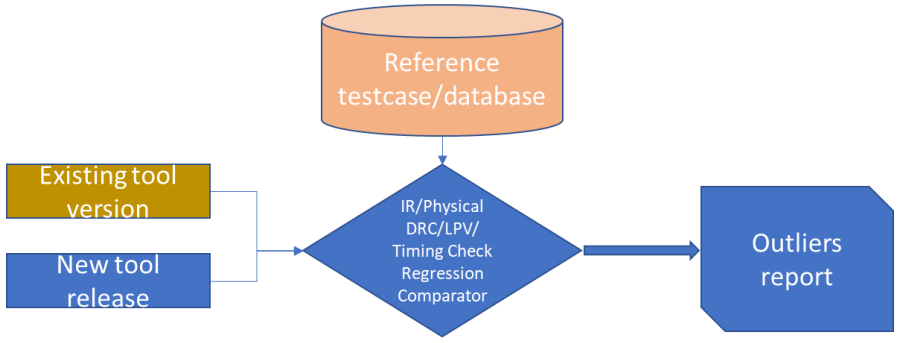


**Fig. 1.12.** Signoff Tool Qualification in Outlier Review [22].

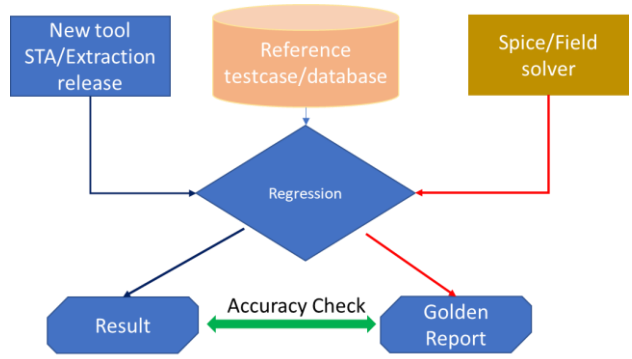


**Fig. 1.13.** Signoff Tool Qualification in Accuracy Check [22].

From Fig. 1.12 and Fig. 1.13, existing check deploys in the field, mainly focus on the outlier review with the existing tape out tool version as golden, as well as accuracy check with golden sign-off tool such as Hspice and Field solver for RC parasitic extraction.

For implementation tool stability check, the methodology is shown in Fig. 1.14. To enable stability quality check, 1 ps frequency adjustment counter is implemented, to adjust the period for the regression. The maximum cumulative adjustment is below 1 % of the max frequency. The results from the stability regression will be used for quality swing analysis. The test-case candidates for the basic quality matrix and the tool stability check, are beyond the discussion of this chapter.

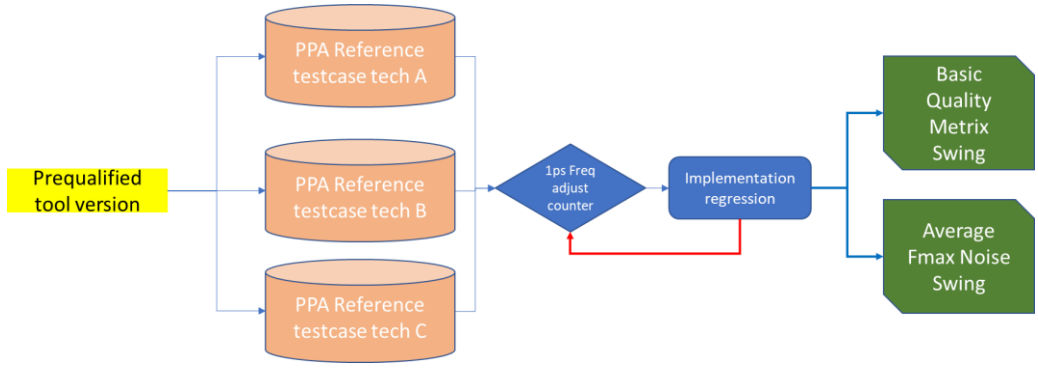


**Fig. 1.14.** Tool Stability Check Methodology [22].

The different between the basic quality check against the tool stability check, is the tool stability check is validated based on the same tool version, across technology with 1 ps frequency adjuster. For the basic quality check, the tool version can be different across technology, different tool version for flow release milestone and different tool version for the same technology but different release milestone. The goal for basic quality matrix check is to validate any outliers against known good base (KGB) runs. Hence, the basic quality matrix allows different tool version comparison, different tool flow methodology

comparison as well as cross technology benchmark. The tool stability check validates the tool sensitivity on particular technology as well as across technology, based on the same design.

## 1.3. Result

### 1.3.1. Integration Quality Check

The overview of death on arrival (DOA) and quality of result (QoR) for various design and tool flow release indicators is shown in Fig. 1.15.


**Fig. 1.15.** DOA Release Metric Indicator [22].

### 1.3.2. Basic Quality Check

The same sample of the basic quality check snapshot is shown in Fig. 1.16 and Fig. 1.17. Fig. 1.16 shows the quality of result (QoR) based on sign-off regression.

From Fig. 1.17, the turnaround time (TAT) quality check is separated from the rest of basic quality check, as turnaround time (TAT) also affected by the machine workload. There is work in progress for machine workload balancing mechanism, which is beyond

the scope of this chapter. For confidential reason, all critical parameters such as technology node, process, voltage, temperature and actual performance target will not be revealed in this discussion.

| | | | | BlockName | | |
|---|---|---|---|---|---|---|
| Flow | Stage | Metric | Target | Base | Comp-0 | Delta-0 |
| FEV_APR | fev_syn2apr | non_equivalent | ∓ 10% | 0.0 | 0.0 | 0% |
| FEV_APR | disk_usage | overall | ∓ 10% | 1.309096 | 1.312196 | 0.2% |
| SGDFT_NETLIST_DRC | disk_usage | overall | ∓ 10% | 0.428536 | 0.732896 | 71.0% |
| SAGE_ATPG | disk_usage | overall | ∓ 10% | 0.481812 | 0.483084 | 0.3% |
| VCLP_APR | disk_usage | overall | ∓ 10% | 0.609212 | 0.609564 | 0.1% |
| EXTRACTION | star_pv | errors_starrc | ∓ 10% | 0.0 | 0.0 | 0% |
| EXTRACTION | star_pv | open_nets | ∓ 2% | 0.0 | 0.0 | 0% |
| EXTRACTION | star_pv | open_nets_in_spef | ∓ 10% | 0.0 | 0.0 | 0% |
| EXTRACTION | star_pv | percentage_good_nets | ∓ 10% | 100.0 | 99.9531 | -0.0% |
| EXTRACTION | star_pv | shorted_nets | ∓ 2% | 0.0 | 7.0 | 700.0% |
| EXTRACTION | disk_usage | overall | ∓ 10% | 0.494364 | 0.495108 | 0.2% |
| NOISE | disk_usage | overall | ∓ 10% | 3.175484 | 2.97474 | -6.3% |
| POWER | test1 | dynamicpoweruw | ∓ 10% | 4184.78076824 | 4141.71359362 | -1.0% |
| POWER | test1 | leakage_poweruw | ∓ 10% | 314.1657 | 313.7258 | -0.1% |
| POWER | test1 | pvscenario | ∓ 10% | func.nominal.TM_100.tttt | func.nominal.TM_100.tttt | 0% |
| POWER | test2 | pvscenario | ∓ 10% | func.highvcc.TT_100.tttt | func.highvcc.TT_100.tttt | 0% |
| POWER | test3 | pvscenario | ∓ 10% | func.highvcc.TM_100.tttt | func.highvcc.TM_100.tttt | 0% |
| POWER | test4 | pvscenario | ∓ 10% | func.nominal.TT_100.tttt | func.nominal.TT_100.tttt | 0% |
| POWER | disk_usage | overall | ∓ 10% | 5.01274 | 4.723236 | -5.8% |
| LV_CALIBRE | denall | total_errors | ∓ 10% | 11.0 | 11.0 | 0% |
| LV_CALIBRE | drc | total_errors | ∓ 10% | 5.0 | 37.0 | 640.0% |
| LV_CALIBRE | disk_usage | overall | ∓ 10% | 1.069016 | 1.076668 | 0.7% |
| LV_ICV | badpolydummyid | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | check_oas | total_errors | ∓ 10% | 1203.0 | 0.0 | -100.0% |
| LV_ICV | check_sp | total_errors | ∓ 10% | 1194.0 | 1210.0 | 1.3% |
| LV_ICV | con2cell | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | denall | total_errors | ∓ 10% | 11.0 | 15.0 | 36.4% |
| LV_ICV | drc_IL | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | drc_IPall | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | drc_TUC | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | drcd | total_errors | ∓ 10% | 2.0 | 2.0 | 0% |
| LV_ICV | drcgclmp | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | gnacpin | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | hgate | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | hier_denall_block | total_errors | ∓ 10% | 9.0 | 9.0 | 0% |
| LV_ICV | libintegrity | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | single_bump | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| LV_ICV | trclvs | total_errors | ∓ 10% | 0.0 | 0.0 | 0% |
| CALIBER | caliber_func_nominal_TT_100_tttt | must_viols | ∓ 10% | 1033.0 | 1058.0 | 2.4% |
| CALIBER | disk_usage | overall | ∓ 10% | 1.947972 | 1.835536 | -5.8% |
| STA_PT | func_highvcc_TM_100_tttt__min | int__lt_n50 | ∓ 10% | 42.0 | 0.0 | -100.0% |
| STA_PT | func_highvcc_TM_100_tttt__min | int__n50_n30 | ∓ 10% | 73.0 | 4.0 | -94.5% |
| STA_PT | func_highvcc_TM_100_tttt__min | tns | ∓ 10% | -17024.449999999993 | -5826.889999999996 | -65.8% |
| STA_PT | func_highvcc_TM_100_tttt__min | wns | ∓ 10% | -336.07 | -198.23 | -41.0% |
| STA_PT | func_highvcc_TM_100_tttt__min | tns_c2c | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_highvcc_TM_100_tttt__min | wns_c2c | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_highvcc_TM_100_tttt__min | tns_int | ∓ 10% | -7218.069999999995 | -1163.6299999999976 | -83.9% |
| STA_PT | func_highvcc_TM_100_tttt__min | wns_int | ∓ 10% | -114.3 | -42.73 | -62.6% |
| STA_PT | func_highvcc_TT_100_tttt__max | int__lt_n50 | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_highvcc_TT_100_tttt__max | int__n50_n20 | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_highvcc_TT_100_tttt__max | tns | ∓ 10% | -337366.9200000002 | -190663.26000000004 | -43.5% |
| STA_PT | func_highvcc_TT_100_tttt__max | wns | ∓ 10% | -470.0 | -342.66 | -27.1% |
| STA_PT | func_highvcc_TT_100_tttt__max | tns_c2c | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_highvcc_TT_100_tttt__max | wns_c2c | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_highvcc_TT_100_tttt__max | tns_int | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_highvcc_TT_100_tttt__max | wns_int | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_nominal_TM_100_tttt__min | int__lt_n50 | ∓ 10% | 22.0 | 0.0 | -100.0% |
| STA_PT | func_nominal_TM_100_tttt__min | int__n50_n30 | ∓ 10% | 65.0 | 0.0 | -100.0% |
| STA_PT | func_nominal_TM_100_tttt__min | tns | ∓ 10% | -7160.699999999999 | -20978.21000000001 | 193.0% |
| STA_PT | func_nominal_TM_100_tttt__min | wns | ∓ 10% | -180.31 | -151.55 | -16.0% |
| STA_PT | func_nominal_TM_100_tttt__min | tns_c2c | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_nominal_TM_100_tttt__min | wns_c2c | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_nominal_TM_100_tttt__min | tns_int | ∓ 10% | -5091.36 | -1.2 | -100.0% |
| STA_PT | func_nominal_TM_100_tttt__min | wns_int | ∓ 10% | -180.31 | -1.2 | -99.3% |
| STA_PT | func_nominal_TT_100_tttt__max | int__lt_n50 | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_nominal_TT_100_tttt__max | int__n50_n20 | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_nominal_TT_100_tttt__max | tns | ∓ 10% | -333526.2399999997 | -163706.00000000003 | -50.9% |
| STA_PT | func_nominal_TT_100_tttt__max | wns | ∓ 10% | -636.74 | -503.44 | -20.9% |
| STA_PT | func_nominal_TT_100_tttt__max | tns_c2c | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_nominal_TT_100_tttt__max | wns_c2c | ∓ 10% | 0.0 | 0.0 | 0% |
| STA_PT | func_nominal_TT_100_tttt__max | tns_int | ∓ 10% | -568.9899999999999 | -82.0 | -85.6% |
| STA_PT | func_nominal_TT_100_tttt__max | wns_int | ∓ 10% | -81.86 | -8.51 | -89.6% |
| STA_PT | disk_usage | overall | ∓ 10% | 1.845584 | 1.797588 | -2.6% |
| PT_ECO | disk_usage | overall | ∓ 10% | 2.424356 | 2.321332 | -4.2% |
| FILL_DP_FC | disk_usage | overall | ∓ 10% | 1.25268 | 1.252988 | 0.0% |
| APR_ECO | disk_usage | overall | ∓ 10% | 1.396752 | 1.38864 | -0.6% |

**Fig. 1.16.** Basic Quality Check Snapshot [22].


Results ...

Show violations only

| | | | | BlockName | parmcschbsddr | |
|---|---|---|---|---|---|---|
| Flow | Stage | Metric | Target | Base | Comp-0 | Delta-0 |
| PREPARE_INDICATOR | clock_route_opt | elapsed_time | ∓ 10% | 11.67 hours | 9.24 hours | -20.8% |
| PREPARE_INDICATOR | compile_final_opto | elapsed_time | ∓ 10% | 2.91 hours | 2.43 hours | -16.5% |
| PREPARE_INDICATOR | compile_initial_opto | elapsed_time | ∓ 10% | 3.92 hours | 3.22 hours | -17.9% |
| PREPARE_INDICATOR | cts | elapsed_time | ∓ 10% | 0.92 hours | 0.76 hours | -17.4% |
| PREPARE_INDICATOR | fill | elapsed_time | ∓ 10% | 1.67 hours | 1.50 hours | -10.2% |
| PREPARE_INDICATOR | finish | elapsed_time | ∓ 10% | 0.49 hours | 0.43 hours | -12.2% |
| PREPARE_INDICATOR | floorplan | elapsed_time | ∓ 10% | 0.51 hours | 0.40 hours | -21.6% |
| PREPARE_INDICATOR | import_design | elapsed_time | ∓ 10% | 0.33 hours | 0.27 hours | -18.2% |
| PREPARE_INDICATOR | init_floorplan | elapsed_time | ∓ 10% | 0.08 hours | 0.08 hours | 0% |
| PREPARE_INDICATOR | initial_map | elapsed_time | ∓ 10% | 0.82 hours | 0.70 hours | -14.6% |
| PREPARE_INDICATOR | insert_dft | elapsed_time | ∓ 10% | 0.30 hours | 0.28 hours | -6.7% |
| PREPARE_INDICATOR | logic_opto | elapsed_time | ∓ 10% | 1.11 hours | 0.92 hours | -17.1% |
| PREPARE_INDICATOR | quick_route | elapsed_time | ∓ 10% | 4.74 hours | 10.30 hours | 117.3% |
| PREPARE_INDICATOR | read_upf | elapsed_time | ∓ 10% | 0.09 hours | 0.11 hours | 22.2% |
| PREPARE_INDICATOR | redefine | elapsed_time | ∓ 10% | 0.10 hours | 0.08 hours | -20.0% |
| PREPARE_INDICATOR | route_auto | elapsed_time | ∓ 10% | 5.18 hours | 11.02 hours | 112.7% |
| PREPARE_INDICATOR | route_opt | elapsed_time | ∓ 10% | 31.16 hours | 26.99 hours | -13.4% |
| PREPARE_INDICATOR | setup_timing | elapsed_time | ∓ 10% | 0.16 hours | 0.14 hours | -12.5% |

**Fig. 1.17.** Runtime Basic Quality Check Snapshot [22].

For basic quality check, as the same design is sweep across technology for different voltage, designers can derive the technology entitlement directly, as shown in Fig. 1.18 and Fig. 1.19.

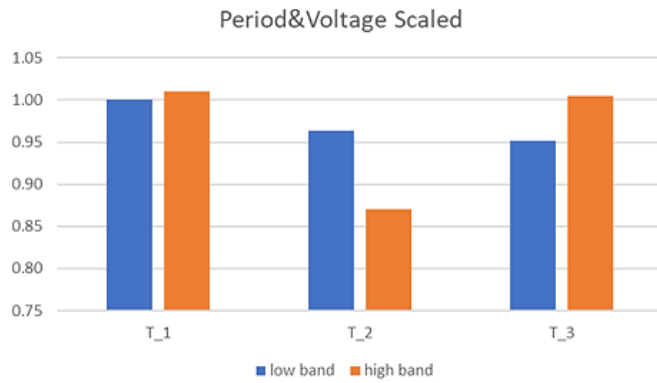


**Fig. 1.18.** Performance Sweep Across Technology [2].

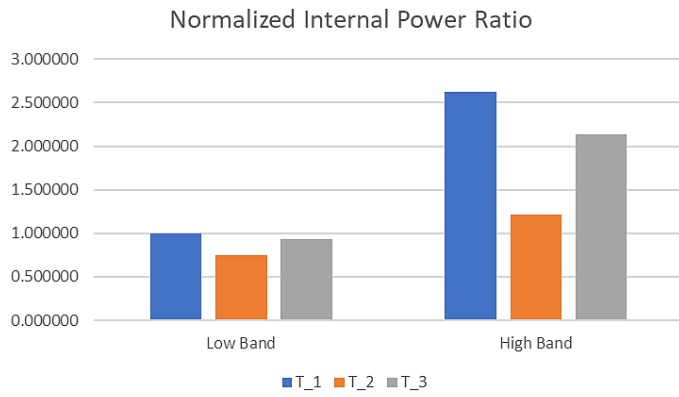


**Fig. 1.19.** Power Sweep Across Technology [2].

### 1.3.3. Stability Check

For stability check, the regression is done on the same design, across technology. For this evaluation, pure standard cell only design is used. For the performance noise quality, it is illustrated in Fig. 1.20.

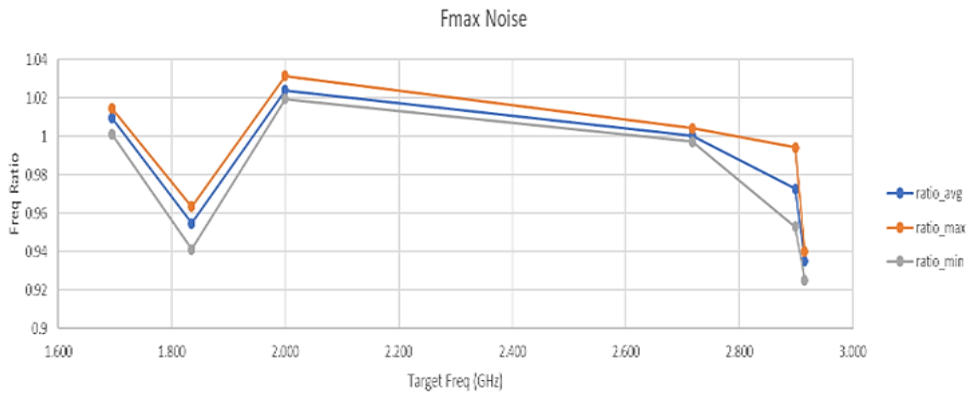


**Fig. 1.20.** Fmax Noise Quality [22].

From Fig. 1.20, for most of the performance target, the swing is around 2 %, except for one performance target, the swing is around 4 %. Further root causing is required, to eliminate the possibility of collateral quality issue.

For the power noise quality plot, it is illustrated in Fig. 1.21. For power noise quality, total power is used as total power is dominated by dynamic power. For power noise quality, total power is used as total power is dominated by dynamic power. From Fig. 1.16, it is concluded that the power noise quality swing around 2 % to 5 % range, for less than 1 % noise injection. Though the power swing percentage is larger than performance swing, further evaluation to be done with power recovery feature through signoff tool.

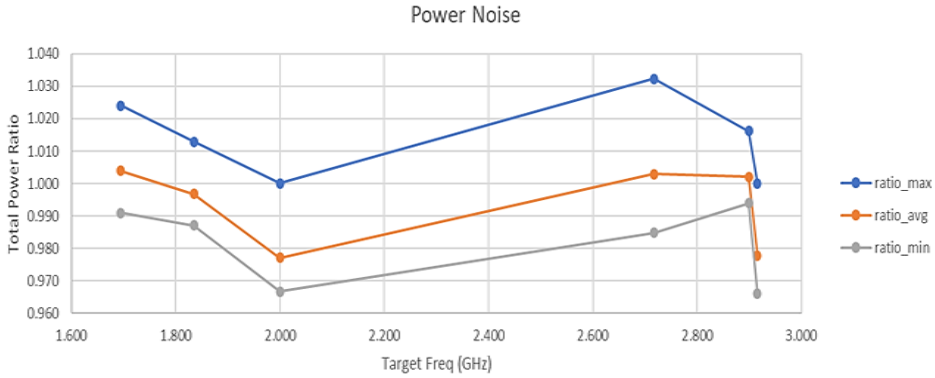


**Fig. 1.21.** Power Noise Quality [22].

## 1.4. Summary

The overview of ASIC pre-silicon qualification will increase the confident of tool deployment. It will reduce the risk of product launching delay due to tool flow methodology quality. The ASIC pre-silicon qualification criteria can be summarized in Table 1.2 below.

ASIC pre-silicon qualification increases the confident of flow and collateral release to project deployment. Hopefully, the sharing will benefit the readers.

**Table 1.2.** ASIC Pre-Silicon Qualification Criteria.

| CRITERIA | IMPLEMENTATION TOOL | VERIFICATION TOOL |
|---|---|---|
| Only deterministic tool is qualified | √ | √ |
| Combined flow should be qualified unless clear partitioning separation can be demonstration | √ | √ |
| Configuration management assurance process should be applied to tools being qualified | √ | √ |
| Qualification should satisfy the objective stated in the specification | √ | √ |
| Tool development operation requirements should be reviewed | √ | √ |
| Tool interoperability and robustness regression should be completed | √ | √ |
| Potential errors should be analyzed | √ | √ |

## References


[1]. G. Huang, J. Hu, et al., Machine Learning for Electronic Design Automation: A survey, ACM Transactions on Design Automation of Electronic Systems, Vol.26, Issue 5, Sept. 2021, 40.

[2]. A. B. Chong, L. J. Keat, P. E. Hong, Q. L. Chuang, L. C. Cheng, Performance Auto Sweep, Design and Reuse, https://www.design-reuse.com/articles/50720/performance-auto-sweep.html

[3]. J. Chen, I. H.-R. Jiang, et al., DATC RDF-2021: Design Flow and Beyond ICCAD Special Session Paper, in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD'21)*, 2021, pp. 1-6.

[4]. M. Slovik, Automotive SOC Brings Functional Safety to CNN Accelerator Cores, ASIL D Control, Electronic Design, https://www.electronicdesign.com/markets/automotive/article/21157036/electronic-design-automotive-soc-delivers-functional-safety-for-cnn-accelerator-cores-and-asil-d-control

[5]. H. Arbel, Bug Escapes and The Definition of Done, Semiconductor Engineering, https://semiengineering.com/bug-escapes-and-the-definition-of-done/

[6]. H.. D. Foster, Out of the Verification Crisis: Improving RTL Quality, https://www.eetasia.com/out-of-the-verification-crisis-improving-rtl-quality/

[7]. J. Chen, I. H.-R. Jiang, et al., DATC RDF-2020: Strengthening the Foundation for Academic Research in IC Physical Design, in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD'20)*, 2020, pp. 1-6.

[8]. M. Pandey, Machine learning and systems for building the next generation of EDA tools, in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC'18)*, 2018, pp. 411-415.

[9]. M. A. White, For SOC ISO 26262 Compliance, Should All EDA Tools be TCL1, https://semiengineering.com/for-soc-iso-26262-compliance-should-all-eda-tools-be-tcl1/

[10]. A. Kahng, Keynote Presentation, ANSYS Executive Breakfast, Machine Learning in EDA: Opportunities and Value Propositions, ANSYS Executive Breakfast Keynote, http://vlsicad.ucsd.edu/Presentations/talk/KahngANSYS-DACBreakfast_talk_DISTRIBUTED2.pdf

[11]. S. Parkinson, Safety in SOCs: Accelerating the Road to ISO 26262 certification for the ARC EM Processor, Design and Reuse, https://www.design-reuse.com/articles/38748/safety-in-soc-iso-26262-certification-arc-em-processor.html

[12]. M. P. C. Mok, K. C. K. Lo, Y. Jiao, Y. K. Li, CPIPQ: A common platform for silicon IP qualification, in *Proceedings of the IEEE International Conference on ASIC*, 2011, pp. 405-408

[13]. N. Bombieri, F. Fummi, G. Pravadelli, M. Hampton, F. Letombe, Functional qualification of TLM verification, in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*, 2009, pp. 190-195.

[14]. A. B. Kahng, S. Mantik, Measurement of inherent noise in EDA tools, in *Proceedings of the International Symposium on Quality Electronic Design (ISQED'02)*, 2002, pp. 206-211.

[15]. D. J. Hathaway, Quality of EDA CAD tools: definitions, metrics and directions, in *Proceedings of the International Symposium on Quality Electronic Design (ISQED'2000)*, 2000, pp. 395-405.

[16]. SolvNetPlus, https://solvnet.synopsys.com/

[17]. D. Yu, H. Foster, T. Fitzpatrick, Survey of Machine Learning (ML) Applications in Functional Verification (FV), in *Proceedings of the Design and Verification Conference (DVCon'23)*, 2023.

[18]. D. Manners, GloFo Certifies 20 nm and 14 nm Design Flows, https://www.electronicsweekly.com/news/business/manufacturing/glofo-certifies-20nm-and-14nm-design-flows-2013-05/

[19]. OpenPiton, Getting work with OpenPiton, https://parallel.princeton.edu/openpiton/tutorial_slides/hpca17/openpiton-hpca17-asic.pdf
[20]. A. B. Chong, Hybrid Multisource Clock Tree Synthesis, in *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems (ICECS'21)*, 2021, pp. 1-6.
[21]. A. B. Chong, S. Bima, N. K. Aun, C. P. Chun, Design Space Optimization with Machine Learning Library Pruning, in *Proceedings of the International Conference on Microelectronics (ICM'21)*, 2021, pp. 86-90.
[22]. A. B. Chong, Q. L. Chuang, L. C. Cheng, K. J. Ian, T. S. Hong, P. E. Hong, L. J. Keat, Implementation Tool Qualification Methodology for ASIC Design, in *Proceedings of the 4th International Conference on Microelectronic Devices and Technologies (MicDAT'2022)*, Corfu, Greece, 21-23 September 2022, pp. 82-90.