2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

An Automated approach for optimizing Circuit Marginality Validation methodologies

Vivek Sharma, Subrata Kumar Behera, Pankaj Sharma
Intel Corporation
vivek4.sharma@intel.com subrata.kumar.behera@intel.com pankaj1.sharma@intel.com

***Abstract*- This paper addresses the key innovations that mitigated Circuit Marginality Validation (CMV) challenges and enabled quick validation with quality Power On (PO) to product delivery within the aggressive timelines. The key challenges resolved in CMV includes aggressive deliverable timelines, managing multiple validation setups (~20), extreme cold temperature validation (-40°C), infrastructure utilization. Addressing these challenges and incorporating the innovations within CMV flow resulted in an end-to-end automated validation process with 50~60 % reduction in the overall validation time. The key innovations discussed in this paper are primarily a Graphical User Interface (GUI) based Jenkins framework for managing multiple validation infrastructure from a single host server, a dynamic BIOS knob configuration as per the validation content and an Anti-condensation temperature control solution for extreme cold temperatures.**

Keywords - Circuit Marginality Validation (CMV), extreme cold temperature validation, dynamic BIOS knobs, post-silicon validation, anti-condensation validation solution, jenkins framework.

I. Introduction

Circuit Marginality Validation (CMV) is a post-silicon validation team that executes in a system environment corresponding to the customer along with extreme voltage, temperature, and frequency conditions. CMV is responsible for identifying and assisting with the debug of circuit marginality issues on System on Chip (SoC) along with providing voltage/frequency (VF) curves that is fused into each silicon for manufacturing. The primary purpose of performing CMV is to identify circuit issues for Design Engineering and Silicon Debug [1], providing content to the silicon binning team for the accurate binning of units and defining quality voltage guardbands (GB) required on all the production units to meet the manufacturing goals.

On the execution front, CMV involves running functional stress test content under various environmental conditions like extreme hot and cold temperatures, frequencies across different IPs, etc. and sweeps the voltage from high to low until the stress content starts failing. The last passed voltage point is known as the minimum voltage point or Vmin. The Vmin represents the minimum voltage point at which the SOC's functionality operates perfectly.

Typically, SoC system validation is performed with multiple parallel validation setups, configuring each setup with multiple configurations as the test case varies across Intellectual properties (IPs) like core, cache, etc. Extended temperature (-40°C temperature compatible) system validation setup includes an anticondensation chamber on top of each system to avoid condensation but still on the long duration test runs, condensation was observed.

To mitigate all these challenges, we have developed a fully automated end-to-end solution to perform CMV validation. The proposed solution includes a Jenkins based framework which can control multiple validation setups from a single setup. An auto configuring python-based framework is incorporated in the CMV flow which can configure dynamic bios knobs as per the different IP test cases resulting in the whole SoC CMV validation in one shot. Also, an Anti-condensation solution is implemented which can control the thermal tool using python in such a manner that no condensation is observed even after running the test cases continuously for 24 hours at extreme negative cold temperatures.

II. Optimizations in existing CMV methodologies

CMV is a validation process developed to expose and isolate circuit failures which can be debugged and fixed in silicon design. Usually, an SoC architecture consists of different IPs with distinct voltage rails. Since CMV is

performed by margining voltages across IPs, an SoC can be categorized into different domains like core, cache, ddr, system agent.

A typical CMV setup consists of a Host PC connected to a System under test (SUT). The SoC is loaded on the SUT platform and thermal head is attached on top of it for providing the extreme hot and cold temperatures. The following section describes the existing methodologies with improved solutions:

A. Parallel queuing multiple setups from a single Server

Typically, CMV validation is performed by engineers handling multiple setups manually, queue in the test cases w.r.t each domain one at a time. Managing ~20 CMV systems is time consuming for setup preparation and validation, this necessitates the allocation of additional resources. To overcome stated problems, we implemented a Jenkins based server-client (or master-slave) automation model as shown in Fig. 1.

Jenkins is an open-source automation tool [2] with a stable GUI offering a variety of tailored features. All Host PCs (CMV setups) are connected with the master server through a JNLP (Java Network Launch Protocol) based connection as shown in Fig. 1. The master server controls and monitors all the CMV setups. The implemented GUI based interface is based on groovy language model where all the required environmental conditions (temperature, frequencies, voltage settings etc.) are provided. The user can select based on the test requirements and trigger execution on multiple setups simultaneously, the master server manages all the load in slave systems. Incase all the setups are already under test execution (busy state) then the job will be queued until the current execution is completed. The main limitation observed with parallel queuing automation, it was only able to complete one domain's test case at a time. This issue is further elaborated in section II.*B.*

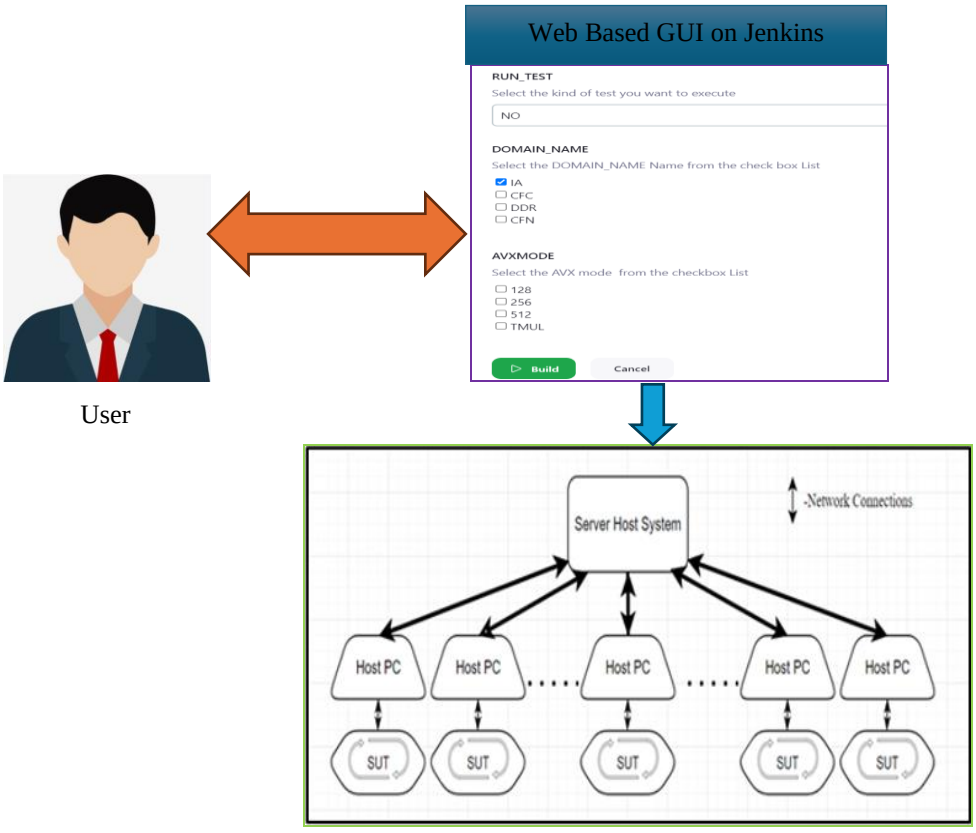

Figure 1. Parallel queuing multiple setups from Single Server - Jenkins model

B. BIOS knob automation using PythonSV

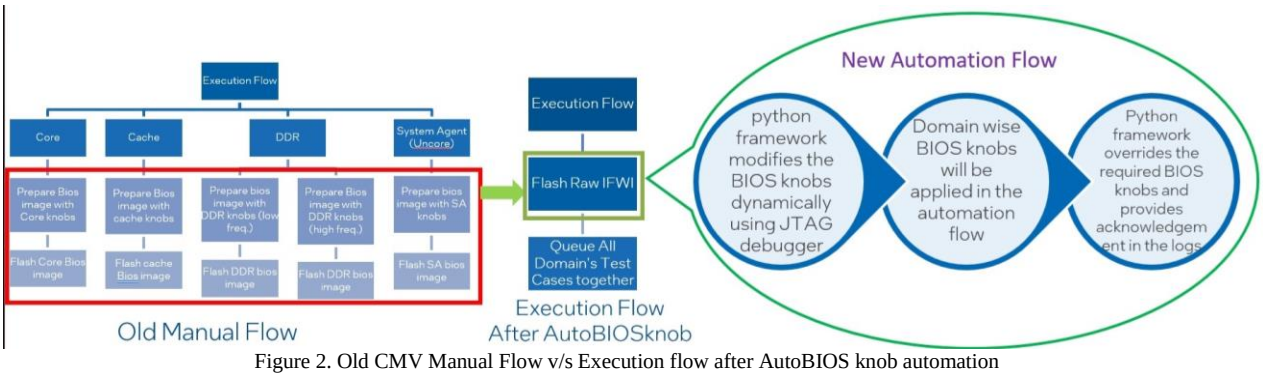

Figure 2. Old CMV Manual Flow v/s Execution flow after AutoBIOS knob automation

CMV SoC validation can be segregated into different domains for e.g. Core, Cache, DDR, System agent (rest of Uncore). Each domain having separate stress test content and thus require different sets of BIOS knob images that needs to be prepared and flashed. The whole preparing and flashing process was tedious and time consuming. Also, all the domain's test cases can't be queued in simultaneously due to this disconnection.

This problem was addressed by implementing a python-based framework within the existing CMV flow. A JTAG (Joint Test Action Group) debugger tool [4] is required on the SUT for accessing and modifying SoC's internal features. The python-based framework allows user to dynamically modify the BIOS knobs. The framework is implemented in such way that it will detect the Domain (Core, CFC, etc.) by reading the set voltage rail through python. Then will override the required sets of BIOS knobs dynamically for every iteration and reset the knobs back to original after each iteration. With this methodology the whole preparing and flashing of multiple BIOS images was bypassed as shown in Fig.2. The main advantage of this improvement is now all the different domain's test cases can be queued simultaneously.

C. Anti-Condensation solution by Thermal tool control methodology

Thermal margining tools are widely used by post-silicon validation teams by varying the case and junction temperature of the thermal tool [3]. Typically, hot temperature validation does not cause major damage to the SoC, but the main challenge arises with the cold temperature (extended temperature or ETemp: 100℃ to -40℃) validation. Usually, a single CMV test case runs for 3 to 7 hours and ideally having isolated thermal chamber setups should avoid any condensation on the SoC and board. But when the stress test content runs on the SoC for a longer duration, the SoC temperature get raised unconventionally. Due to this the overall temperature changes from the set cold temperature (-40℃) leading to condensation in longer execution runs (10+ hours). This issue ultimately leads to board damage, boot issues, manual monitoring in lab (by removing SoC form socket and checking for droplets) and underutilization of ETemp setups.

The condensation issue was eliminated by implementing a Thermal tool control methodology within the CMV flow. The Anti-condensation automation follows as:

- After every test case completion tool's API is controlled through the python code.
- The temperature is slowly incremented from -40℃ upto 50℃, by stepping up 10℃ after every minute as shown in Fig. 3.
- Thermal head halts temperature at 50℃ for 10 minutes in case any water droplets (if present) can be evaporated.
- After that the new queued test case resumes.

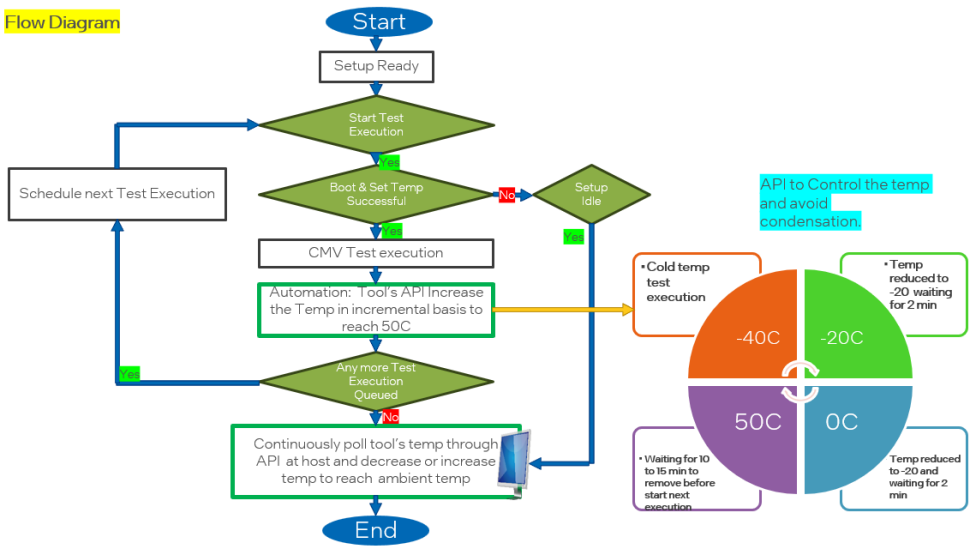

Figure 3. Flow of Anti-Condensation solution by Thermal tool control methodology

III. RESULTS

CMV is one of the major time-consuming processes in Post silicon validation cycle [5]. The typical CMV validation requires significant amount of manual efforts, multiple head counts and long timeline for complete end-to-end validation. This paper proposes some innovative methodologies for addressing these issues and the consolidated automations are shown in Fig. 4.

- Jenkins is an open-source automation tool which is implemented such that it triggers multiple setups from a single host through command line interface (CLI). Resulting in much optimized method for validation as compared to the traditional method.
- The dynamic bios knob automation is based on a python framework which is scalable with minor modifications.
- Anti-condensation thermal control solution is also a python-based solution. Utilizing this solution resulted in zero condensation during extreme negative temperature validations. Thus, saving costly hardware devices from damage along with 100% utilization of infrastructure.

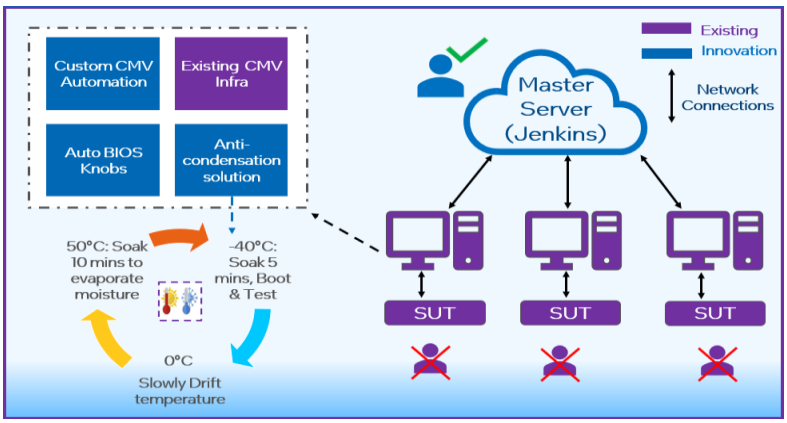

Figure 4. Innovative approach (in blue colour) improving the Existing methodology (in purple)

Listed are the implemented solutions presented in this paper: -

A. A GUI based Jenkins framework was developed which worked as a central server host for controlling multiple setups (20+) from a single setup. Following are the impacts and results:

- Better resource planning, headcount, time, and resource saving by 25% from running parallel executions overnight and weekends with least monitoring.
- Avoid manual test queuing individually on each setup with minimal monitoring.
- Schedule pull in by 25-30% as shown in Fig. 5.
- Left shift of 2 weeks has been achieved with this implementation as compared with the manual approach.

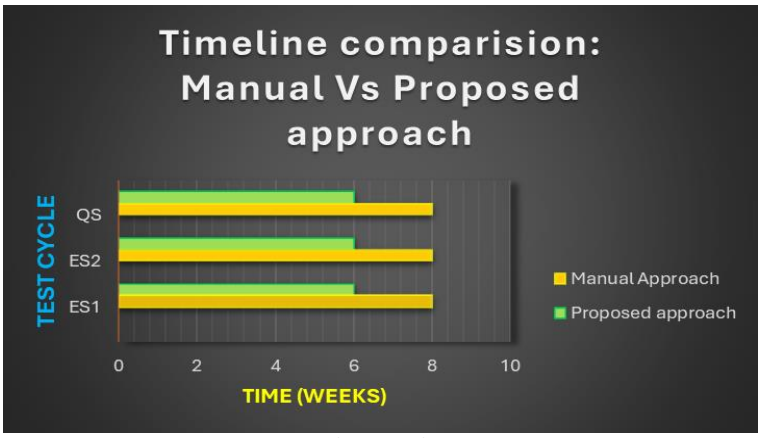

Figure 5. Timeline comparison before and after Parallel queuing automation

B. Since CMV required different BIOS knobs for different domains (IPs), a dynamic BIOS knob override framework is implemented within the existing CMV validation flow. Following are the key impacts and results:

- Automation reduces more than 80% of manual efforts during setup preparations, comparison graph is shown in Fig. 6.
- All the domain's test cases can be queued at once.
- The number of times the Integrated Firmware image (IFWI) needs to be prepared and flashed is reduced from 6 to just 1.
- Any mistakes in manual BIOS knob settings can be avoided.

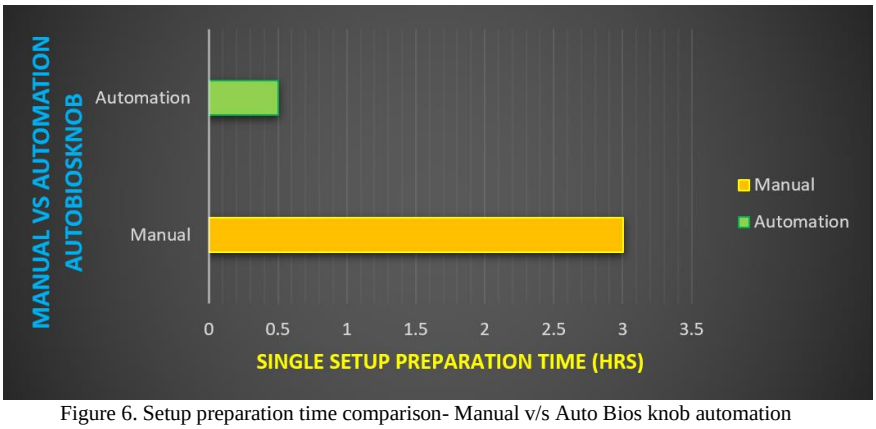

Figure 6. Setup preparation time comparison- Manual v/s Auto Bios knob automation

C. An Anti-condensation temperature control solution implemented in the CMV flow resolved all the condensation issues during the extreme negative temperature (-40℃) executions for long hours continuous runs. Following are the key results and impacts:

- Even after running the test cases 24/7, no condensation (water droplets) was observed.
- Minimal monitoring of test case executions with sustained cold temperature.
- Infrastructure utilization increased by 40~50%, setup utilization in a day is shown in Fig. 7.
- Better resource utilization, efficiency improved by more than 50%.
- No instances of HW damage observed due to condensation.

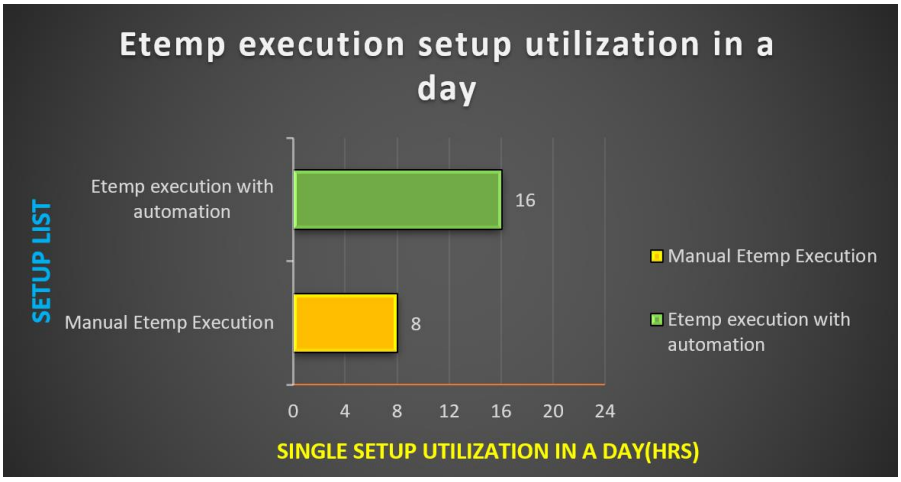

Figure 7. Single Etemp execution setup utilization in a day

Enabling all these solutions together resulted in around 50~60% reduction in overall validation time, ultimately resulted in meeting the aggressive product delivery deadlines without compromising with the quality of validation.

IV. Conclusion

CMV being a major time-consuming function in Post silicon validation cycle, required enhancements in the existing methodologies. This paper discussed the key advancements done in CMV methodologies that resulted in delivering aggressive product deliverables. The key challenges faced in CMV were handling multiple setups individually, preparing and flashing different IFWIs for different domains & condensation issue during cold temperature validation. Addressing all these pain points of manual interventions and clubbing them together resulted in an end-to-end automated CMV validation process with 50~60% reduction in overall validation time.

The methodology advancements presented in this paper can be adopted by CMV teams across organizations (with some minor modifications). Ultimately, the aim & motivation for this paper is to share the improved methodologies across validation teams to optimize the complete CMV process for future products.

Acknowledgment

Thanks to the PSV Staff for their valuable feedback.

References

[1] S. Shaikh, "Test and characterization of high-speed circuits," in *29th VLSI Test Symposium, Dana Point*, CA, USA, 2011.

[2] V. Armenise, "Continuous Delivery with Jenkins: Jenkins Solutions to Implement Continuous Delivery," in *2015 IEEE/ACM 3rd International Workshop on Release Engineering*, Florence, Italy, 2015.

[3] R. Mohammed, R. Sahan, Y. Xia and Y. -f. Pang, "High performance air-cooled temperature margining thermal tools for silicon validation," in *27th Annual IEEE Semiconductor Thermal Measurement and Management Symposium*, San Jose, CA, USA,, 2011.

[4] K. Lee, Y. Lee, H. Lee and K. Yim, "A Brief Review on JTAG Security," in *2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, Fukuoka, Japan, 2016.

[5] S. Mitra, S. A. Seshia and N. Nicolici, "Post-silicon validation opportunities, challenges and recent advances," in *Design Automation Conference*, Anaheim, CA, USA, 2010.