Identifying unique power scenarios with data mining techniques at full SoC level with real workloads

Amir Attarha
Email: amir.attarha@siemens.com

Pankaj Chauhan, Diwakar Agrawal, Satish-Kumar Agrawal, and Gaurav Saharawat
eda.sw.siemens.com/en-US/

***Abstract*- An immense amount of power data can be generated by Veloce® Power App at the SoC level while running real workloads (end-user software or applications). Exploration and analysis of all the raw power data is an extremely necessary, laborious, and repetitive task that needs to be accomplished in a timely manner for each RTL release. This paper demonstrates advanced techniques that can drastically reduce analysis and exploration time by using data mining techniques and a key power indicator tracking system for comparative analysis of various design RTL revisions or different workloads. In this methodology, data is captured for the entire duration of the workload for desired IPs and then saved in data frames friendly for data mining. A preprocessing step is performed for data wrangling (cleaning up the data, if needed) and normalizing it. Key features for the data set are extracted based on the deep understanding of requirements of power analysis at the SoC level, and then various techniques are used for clustering to categorize power scenarios for each IP based on the workload. This methodology is also powerful in profiling different workloads for power and significantly saves design teams time and effort while allowing them to focus on each category rather than one item at a time.**

I. INTRODUCTION

Early power analysis and estimation is becoming a necessary step in application-specific integrated circuits (ASIC) development as power has become the new fundamental challenge of the semiconductor industry for an entire range of products. As shown in [1], with the failure of Dennard scaling (i.e., slowing supply voltage scaling with technology shrinkage), it is expected that power consumption will progressively worsen and every inefficiency that consumes unnecessary power becomes crucial to the design and needs to be circumvented.

Stimulus is one of the key factors in determining the quality of power analysis and estimation. Using synthetic tests (defined as manual or target tests created by the verification team to exercise different power corners), for analyzing power leads to highly inaccurate conclusions; either too pessimistic or too optimistic. Hence, it is becoming common practice to use real workloads along with end-user software for power analysis as early as possible in the design cycle. To accomplish this, hardware-assisted verification platforms, such as Veloce, are heavily used in the semiconductor industry. These applications may take millions, or even billions of cycles. For instance, Dhrystone and Geekbench are commonly executed on CPU subsystems [2], while Aztech, Manhanttan, and FireStone[3] are used for the graphics processing unit (GPU). They are generally executed after the operating system (OS) boot is completed. To run these applications, either the SoC system is available, or hybrid configurable platforms, such as Veloce HYCON, are used to quickly integrate an IP to a configurable system, enabling end-user software to be run at the IP level for functional verification, performance measurements, and most importantly, power analysis.

As stated, end-user software utilizes millions to billions of cycles. This produces a massive amount of data that needs to be analyzed for each RTL release, and for each IP in the design to be reviewed for power inefficiencies and opportunities to improve them. Currently, this analysis is being performed either manually by power teams or ignored due to a lack of resources for this essential task [3]. However, this task can now be accomplished via technologies such as the Veloce PowerStream App and methods offered in this paper.

In this work, we propose new methods to extract key information from generated power data and key power metrics using data mining algorithms and a key power indicator (KPI) tracking system. Based on the power profile of each

IP and its KPIs (such as block-wise power indicators, peak power, average power, energy, effective clock frequency, clock gating efficiency, flip-flop efficiency and more), IPs are clustered in various classes that indicate similar behavior. In addition, their activity trend is individually analyzed to understand the Power Behavior State (defined as the overall behavior of the IP in terms of power, as calm, choppy, active, and “Tsunami” and abbreviated as PBS) and duration of residency in each PBS.

Power Behavior State (PBS)
Note that Power Behavior State, or PBS, is a new term that we are introducing. PBS allows the software to abstract power profiles or cycle accurate power patterns as higher-level information. That information can be then extracted through algorithms, compared and collected (without the need for visual inspection) in terms of coverage metrics (akin to functional coverage metrics in SoC verification). Figure 1 shows a typical power profile with four overlaid power behavior states. ‘Calm’ refers to the duration when power activity has minimal variations. ‘Choppy’ is defined as the duration with high rise and fall or vice versa. ‘Tsunami’ is defined as the occurrence of a sudden rise or fall like a step function. And finally, ‘active’ is referred to as the time period when there is quite a lot of power activity.

Adding this abstraction layer along with KPI tracking enables the power team to reach key conclusions about power analysis and optimization by avoiding many long hours of expert analysis. This is accomplished by focusing on only one candidate of the entire category rather than looking at each test/IP individually.

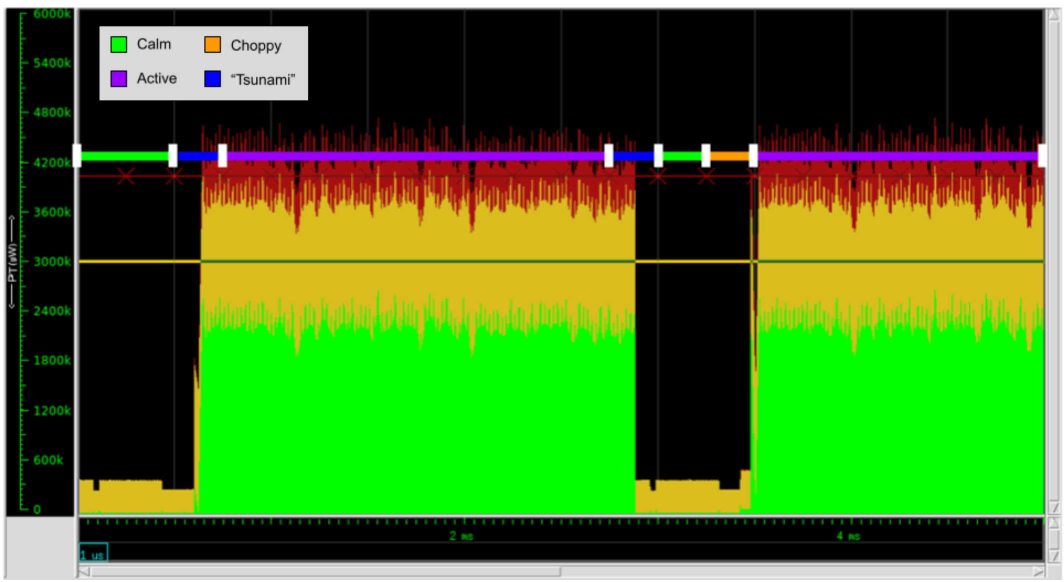


Figure 1: An example of a typical power profile

Power Behavior State			
Calm	Tsunami	Choppy	Active

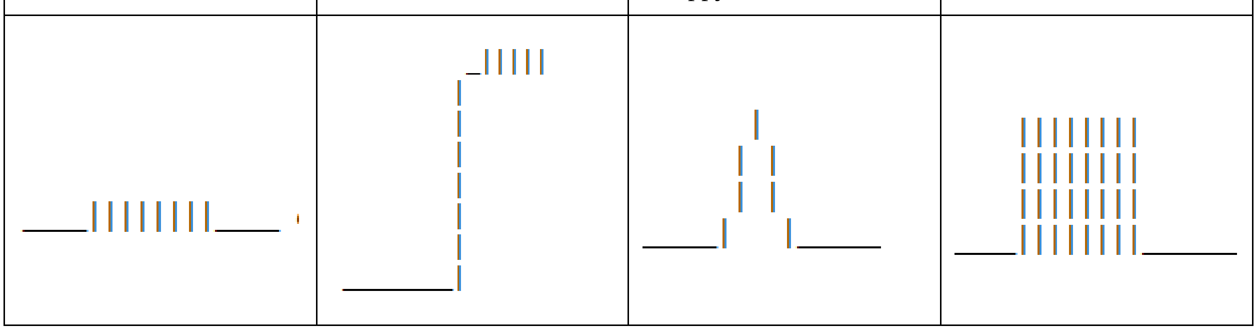


Figure 2: Power Behavior States (Calm, Choppy, Tsunami and Active)

The rest of this paper is organized as follows: Section II discusses the related work and background on both power analysis and data mining. Section III focuses on our proposed solution. Section IV covers experimental results. Finally, Section V concludes the paper.

II. Background

In this Section, we overview the background knowledge on the collection of power data in ASICs. Then, some of the key methods used for data mining are discussed since the core of this paper focuses on data mining technique for extraction of key conclusions from power data.

Power Data Collection

Traditionally, a limited number of tests for very short periods of time were executed for power analysis using different vector formats, such as value change dump (VCD), fast signal database (FSDB), and switching activity interchange format (SAIF). However, with the complexity of new SoCs and the widespread usage of hardware-assisted verification platforms, much longer tests are being executed. In addition, there has been significant advancement in the generation of power data using hardware-assisted verification platforms, specifically Veloce PowerStream App. This technology directly generates power data that requires much less processing to compute **power profiles** and **key power indicators** (KPI) with a very high level of accuracy at per-clock-cycle granularity by adding instrumentation to the hardware. Key power indicators are important metrics for power analysis, such as average power, peak power, average energy, average CGE and more. CGE will be defined later.

"**Power Profile**", as shown in Figure 1, presents a bird's-eye view of the power behavior of a workload whose trend matches silicon. In general, generating power profiles for long-running workloads was very time-consuming and nearly impossible since power tools were able to handle a very limited number of cycles per hour (in order of 1,000 to 2,000 cycles per hour). With new technologies for power profiling such as Veloce PowerStream App, power profiles can be generated in matter of a couple hours i.e., at 100s of Kilo Hertz (KHZ); capturing power for every clock cycle for entire duration of a long test in a very short time. Power profile is generated based on every node in the design as compared with a selected set of signals in some other technologies which suffer from lack of visibility. In addition, KPIs, such as clock-gating efficiency (CGE) and Flip-Flop Efficiency (FFE), are generated in this flow for the entire SoC. CGE is defined for a flip-flop as the percentage of clocks that are gated out of the total number of clocks received by the flip-flop. CGE for a block is the average CGE of all the flip-flops in that block. FFE for a flop is defined as the percentage of flip-flop output changes in relation to the total number of received clocks. With Veloce PowerStream App, all the required data is generated at 100s of KHZ with high accuracy for entire during of the test.

Data Mining

Time-series data is defined as a series of data points sampled over time. They are very useful in identifying patterns, trends, associations, and rules. Many power metrics are captured in a time-series data like power profiles. Depending upon duration of data collection, the power data can be captured for quite a long duration. Dynamic Time Warping (DTW) [4] is one of the key methods used to analyze time-series data and find similarities among them. It is a very computationally expensive algorithm and does not scale for long time-series data. There are proposed enhancements to this algorithm, as described in [5], where they enable parallelizing a modified version of DTW on a GPU using CUDA. Also, deep convolutional networks have recently been applied to time-series classification tasks [7]. In general, they demonstrate good results in the case of sequential tasks.

Another approach utilizes key-feature extraction of time-series data and providing the features for clustering [8][9]. The feature extraction requires deep, expert knowledge, and the quality of results is highly dependent on the quality of extracted features. Experts need to identify representative features in respect to the application to identify core intent of the knowledge. This may be an extensive iterative process.

There are many machine-learning algorithms designed for feature and specific state-representation extraction [11]. However, they do not lead to optimum features and require many iterations. More sophisticated methods are presented in [12]. The graph-convolutional neural network (GCN) was used to extract the topology-related features. They cannot be fed to fully connected neural networks since they are non-Euclidean. Instead, a GCN was designed that can include the topology transformation to transform links to nodes [12]. These methods, although effective, are also computationally intensive.

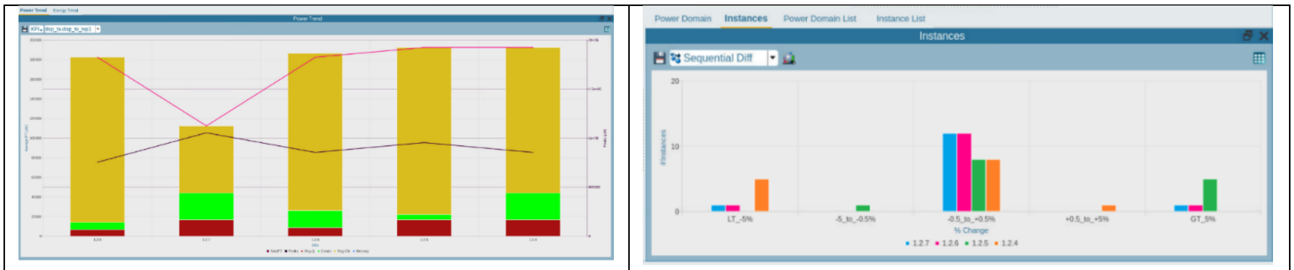
III. Proposed Solution

Massive amounts of data collected during power analysis needs to be analyzed for each RTL release, individual test, and IP. In this section, we define our solution to analyze the data automatically to enable the extraction of key information and provide key feedback to highlight possible issues, such as unexpected power peaks, low clock gating efficiency, big average power vs peak power variations and more. In this work, we take advantage of traditional data-analysis methods, as well as new data-mining methods. Artificial intelligence (AI) and machine-learning methods are considered as another tool in the toolbox rather than the only tool. Hence, we start with KPI tracking and analysis first and then we expand our work to data mining.

KPI Tracking and Analysis

Traditional comparative analysis and statistical analysis are still very efficient methods to analyze data and discover inconsistencies and opportunities for improvement. Power data for entire workloads using PowerStream App can be collected at 100s of KHZ allowing for these commonly used methods to be used. Note that this data needs to be collected weekly for each RTL release, every workload, and every IP, creating a significant challenge for data analysis. With this level of data complexity, simple spreadsheets or web pages do not suffice. A KPI database system has been designed that keeps track of key information in a database and allows analysis for:

- Comparison of power profile and metrics among IPs
- Impact of workload on each IP in the design
- Comparison of same workload across various RTL releases or even projects
- Comparison of the impact of different workloads on the same IP or on entire SoC



(a)	(b)

Figure 3: Tracking and analysis of Key power indicators based on RTL release and power metrics

Figure 3(a) illustrates key power information tracking across multiple RTL releases over the lifetime of the project. Figure 3 (b) shows the distribution average power of IPs compared with a reference model. One could focus on the list of IPs which use 5% or more power in comparison with the reference model and investigate whether this is expected or not. Figure 4 shows the comparison of power activity of an IP across various end-user software tests. Different colors indicate different power components, such power related to flops, memories and combinational circuit.

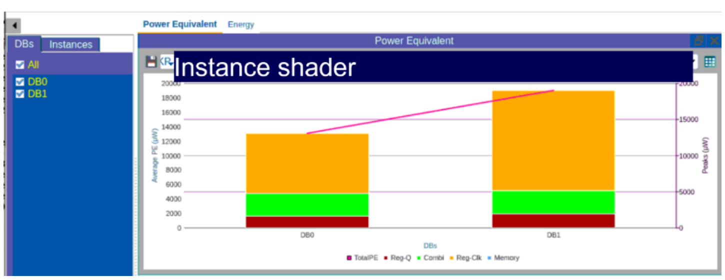


Figure 4: comparison of power activity of an IP (shader) across various end-user software tests.

Data Mining

There are two main objectives we address in this Section. First, categorizing all the instances in a design that behave similarly within a testcase; hence, a power engineer needs to analyze much less data (i.e., analyzing a handful of scenarios rather than 100s of instances). Second, for each category, analysis is done to understand the PBS of the instance and duration that each state lasted per PBS. This information allows the power team to find overall dominant PBS during an entire regression of tests and instances. Then, power teams can apply optimization schemes to the design that provide an advantage across many tests leading to significant energy savings.

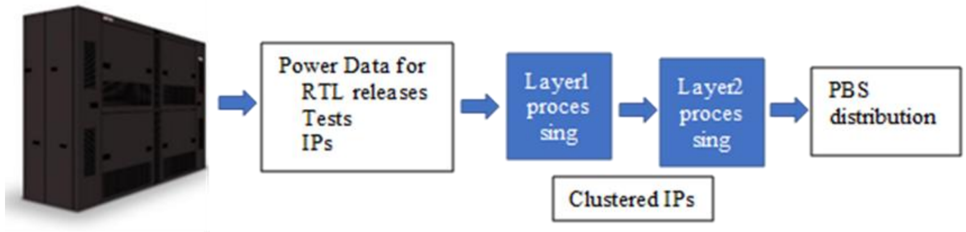


Figure 5: Multi-layer data mining approach for extracting PBS distribution

We are applying a classic data mining process as shown in Figure 6 with additional training feature extraction. Initially, power data is collected including power data at clock cycle granularity and its derivatives. In addition, higher level KPIs such as CGE and FFE are calculated, and the data is categorized. Then, data cleaning or wrangling is performed to ensure the received data is useful and consistent. Note that some data are collected per clock cycles, and some are calculated over a specific window due to the nature of the data. For instance, CGE is only meaningful across a window, not at a discrete point. We also performed a noise cancellation step to smooth the data and avoid false indications since there are many localized fluctuations. Later, data normalization is performed. This step enables us to compare results from various design sizes and various tests.

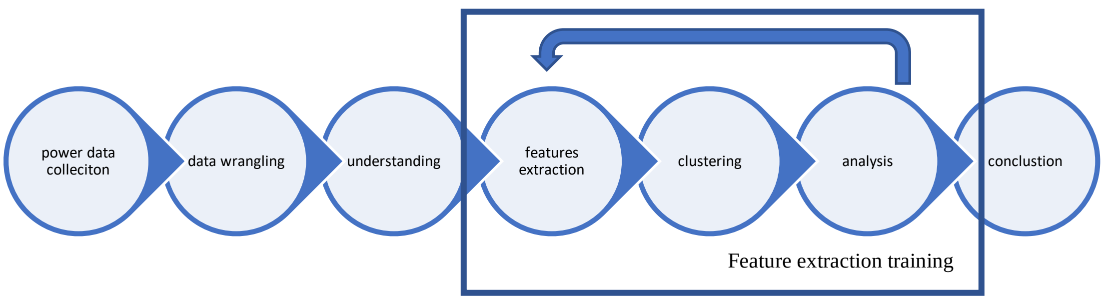


Figure 6: Data mining flow from start to meaningful insight

Feature extraction

In knowledge base programming, expert knowledge is the key to converge with meaningful results quickly; otherwise, massive unsupervised algorithms can be used which are computationally expensive and require many iterations to reach desired data. In addition, unsupervised methods require constant updates to the training set as new information is continually provided. In this solution, we chose the former approach to take advantage of our expert knowledge in the field and extracted an initial set of features for analysis. The features were used for the rest of the data mining process, and the result of this step were viewed and analyzed. Noncontributing features were removed, and newly generated features were added. This is an iterative process. Note that a feature can be defined as a meaningful property of dataset. For instance, average, mean, max power can be considered among several simple features of a dataset.

Machine learning model

With features extracted and golden state identified manually for a small set of the scenarios, we modeled the problem as a clustering problem. In general, the training set is extremely large as we are collecting data for every clock cycle.

One commonly used analysis algorithm for dataset clustering is k-means [13][14] used for clustering of a dataset. It randomly selects k points from the dataset and then assigns each datapoint to a cluster whose centroid is closest to that point. The new cluster centroid is recalculated. Its initial point or seed is important. Another important factor in the performance of k-means is associated with the number of clusters. We have evaluated different numbers of clusters and decided to set for four clusters which are sufficient for our dataset. In the next section, we explore experimental results.

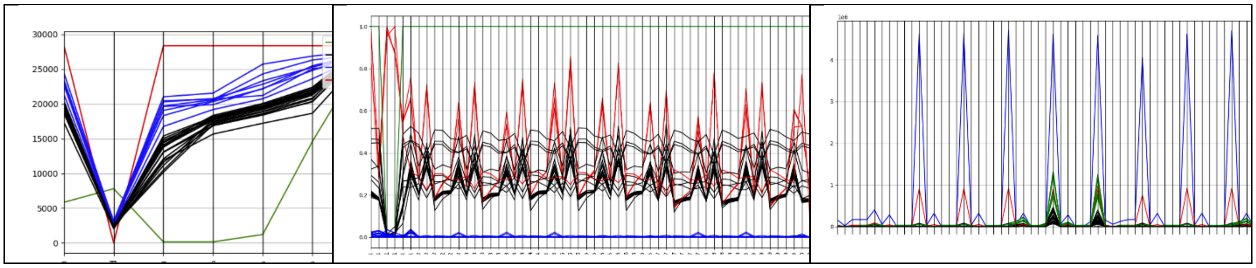


Figure 7: various experiments for key feature extraction-X-axis represents feature list and Y-axis shows the deviation for each cluster

IV. Experimental Results

Three different designs were used for experimentation. Design information is shown in Table 1. Different end-user software tests were used to capture power data using Veloce PowerStream App. The data is generated per hierarchical instance based on the size of the instance. If an instance is too small, data is not generated as it may not provide any value. However, the user has control to select the minimum size as needed.

The end-user software tests are real applications and the number of cycles needed to complete them is added to the table. Power data was collected for tens of instances per design for a long-time window. Analyzing this volume of data on a weekly basis is a massive challenge as it takes many hours of experts' time.

By applying the two-layer approach discussed earlier, we were able to cluster the power profile based on the extracted features in four categories. The power profiles located in each cluster show very similar behavior. For instance, Figure 8 shows three power profiles for three randomly selected instances of Design 1 in cluster 1. We also compared power profiles from different clusters. As you can see from Figure 9, power profiles for Design 1, cluster two, three, and four show different behaviors; hence, placed in different clusters. In other words, our feature extraction and clustering have successfully provided the expected results.

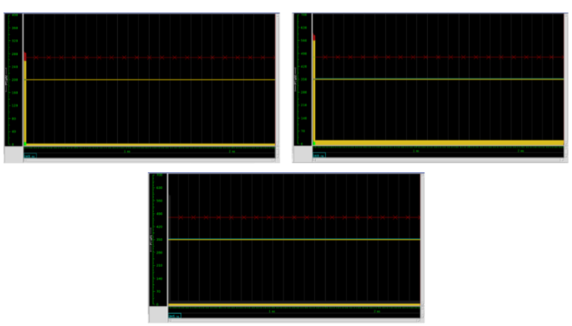


Figure 8: Three instances from Design 1 Cluster 1

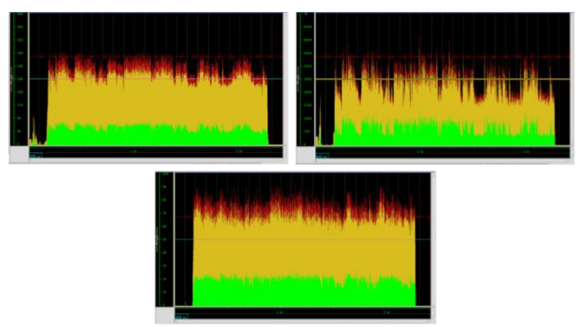


Figure 9: Instances from Design 1 Cluster 2, 3, & 4

TABLE I: Different design types and end user software tests statistics

	Type	Number of instances data collected	Test length (million cycle)
Design 1	CPU subsystem	73	240
Design 2	GPU	48	850
Design 3	CPU	26	14

TABLE II: Clustering of different IPs based on the end-user software tests

	Number of instances	Number of instances after data wrangling	Number of clusters	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Design 1	81	73	4	50	2	18	3
Design 2	49	48	4	1	37	9	1
Design 3	26	26	4	8	2	4	12

The second layer of data mining was used to generate PBS. We reduced the scope of the problem from more than 150 IPs to 4 representative IPs per design using step one as shown in Table II. In the second step, we used the algorithm trained previously to identify PBS states. Figure 10 shows the duration of time each IP resides in each PBS. This enables designers and power engineers to immediately focus on the right scope for their power optimization, power estimation and even dynamic IR drop candidates. Figure 10 shows the PBS distribution for different designs. For instance, for Design 1 you can see that the test behaves differently on various clusters; allowing power team to target right estimation and optimization on the desired cluster.

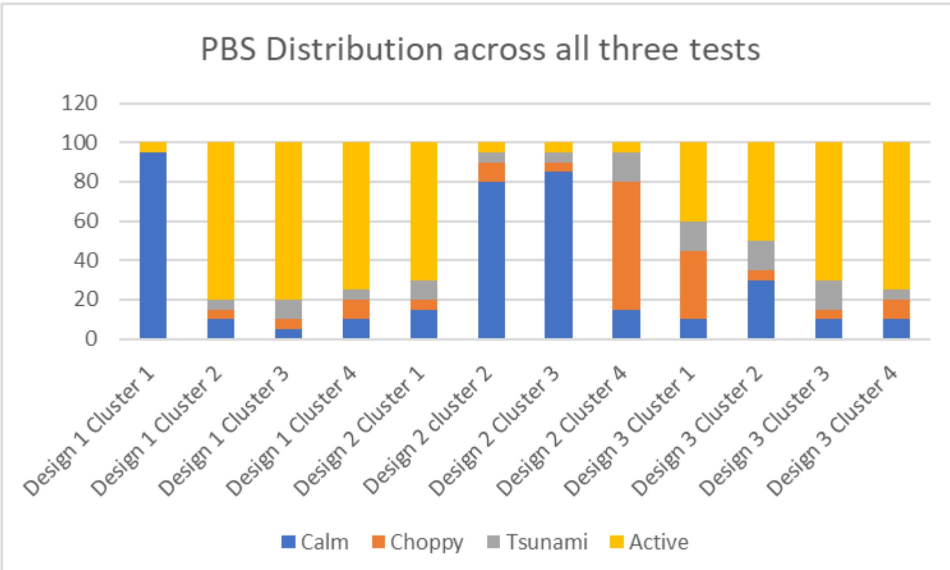


Figure 10: PBS distribution across all three tests

V. Conclusion

In this paper, we discussed a fast and productive method for generation of power profiles and key power indicators. Two approaches for analyzing the generated power data were presented to provide a comprehensive solution. The traditional methods for analyzing this data were demonstrated by collecting the right metrics and enabling basic statistical and comparative analysis. The core innovative approach of this paper focuses on advanced data mining schemes to generate key features that allow categorizing all the power profiles in four categories or clusters. Automated clustering of power profiles / KPIs help save designers their precious times and allow them focus on right cluster and identify problem areas. In the second layer, the power behavior of extracted clusters is analyzed, and each category is formulated in terms of power behavior state (such as ‘calm’, ‘Choppy’, ‘Tsunami’, ‘Active’). In addition to automation, this brings the concept of power data one layer abstraction higher so the user can extract meaningful insight and conduct comparison and optimization.

Acknowledgment

We would like to thank our colleague Andrew Gaspar for performing many experiments and helping us to organize the data.

References

[1] M. B. Taylor, "A Landscape of the New Dark Silicon Design Regime," in *IEEE Micro*, vol. 33, no. 5, pp. 8-19, Sept.-Oct. 2013, doi: 10.1109/MM.2013.90.

[2] H. W. Oh, K. N. Cho and S. E. Lee, "Design of 32-bit Processor for Embedded Systems," 2020 International SoC Design Conference (ISOCC), 2020, pp. 306-307, doi: 10.1109/ISOCC50952.2020.9332944.

[3] K. Straube, J. Lowe-Power, C. Nitta, M. Farrens and V. Akella, "Improving Provisioned Power Efficiency in HPC Systems with GPU-CAPP," *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, 2018, pp. 112-122, doi: 10.1109/HiPC.2018.00021.

[4] M. Muller, ¨Information Retrieval for Music and Motion." Springer, 2007.

[5] H. Zhu, Z. Gu, H. Zhao, K. Chen, C. -T. Li and L. He, "Developing a pattern discovery method in time-series data and its GPU acceleration," in Big Data Mining and Analytics, vol. 1, no. 4, pp. 266-283, December 2018, doi: 10.26599/BDMA.2018.9020021.

[6] H. Abe, H. Yokoi, M. Ohsaki and T. Yamaguchi, "Developing an Integrated Time-Series Data Mining Environment for Medical Data Mining," Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), 2007, pp. 127-132, doi: 10.1109/ICDMW.2007.47.

[7] Cui, Z., Chen, W., and Chen, Y. "Multi-scale convolutional neural networks for time-series classification," arXiv preperent arXiv

[8] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann and K. Schindler, "Learning Aerial Image Segmentation From Online Maps," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6054-6068, Nov. 2017, doi: 10.1109/TGRS.2017.2719738.

[9] M. Moradi A., S. A. Sadrossadat and V. Derhami, "Long Short-Term Memory Neural Networks for Modeling Nonlinear Electronic Components," in *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 5, pp. 840-847, May 2021, doi: 10.1109/TCPMT.2021.3071351.

[10] S. Woo and C. Lee, "Incremental Feature Extraction Based on Gaussian Maximum Likelihood," *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 2019, pp. 1-4, doi: 10.1109/ITC-CSCC.2019.8793458.

[11] X. Chen, J. Guo, Z. Zhu, R. Proietti, A. Castro and S. J. B. Yoo, "Deep-RMSA: A deep-reinforcement-learning routing modulation and spectrum assignment agent for elastic optical networks," Proc. Opt. Fiber Commun. Conf. Expo., pp. 1-3, 2018.

[12] L. Xu, Y. -C. Huang, Y. Xue and X. Hu, "Deep Reinforcement Learning-Based Routing and Spectrum Assignment of EONs by Exploiting GCN and RNN for Feature Extraction," in *Journal of Lightwave Technology*, vol. 40, no. 15, pp. 4945-4955, 1 Aug. 1, 2022, doi: 10.1109/JLT.2022.3175865.

[13] W. Zhao, H. Ma, and Q. He, "Parallel K-Means Clustering Based on MapReduce," vol. 5931, Springer Berlin/Heidelberg, 2009, pp. 674-679.

[14] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding," SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007.