# Clustering VCD dump and Report Functional coverage.

Shivam Porwal

# Content

- Understand the problem
- Input Data
- Intent and process
- Data preprocessing
- ML Process
- Post processing
- Proof of concept
- Challanges
- Future Scope

# Let’s understand the problem

Functional coverage-

In VLSI, "functional coverage" refers to a metric that measures how much of a design's intended functionality has been exercised by the verification process.

- Existing EDA tools does not report functional coverage.
- Inadequate or Missing Functional Coverage Bins

# Let’s understand this Example

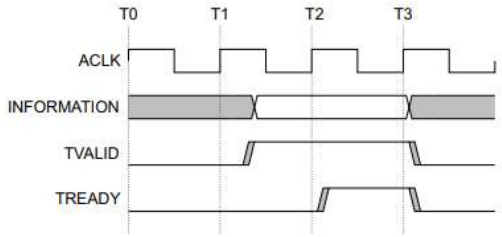



**Figure 2-1 Handshake with TVALID asserted before TREADY**

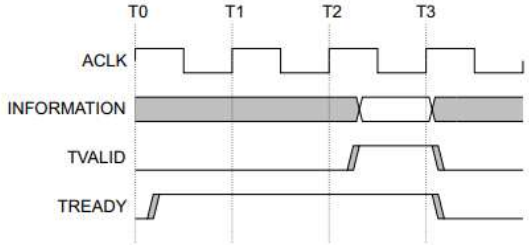
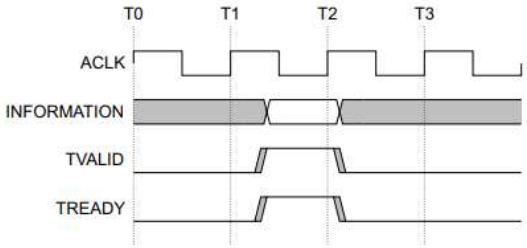


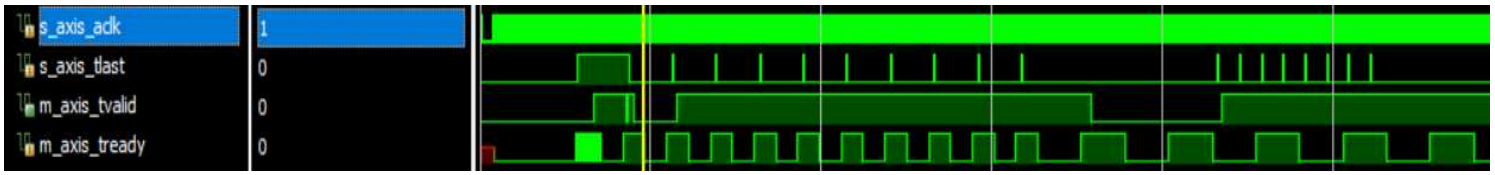

**Figure 2-2 Handshake with TREADY asserted before TVALID**




**Handshake with TVALID and TREADY asserted simultaneously**

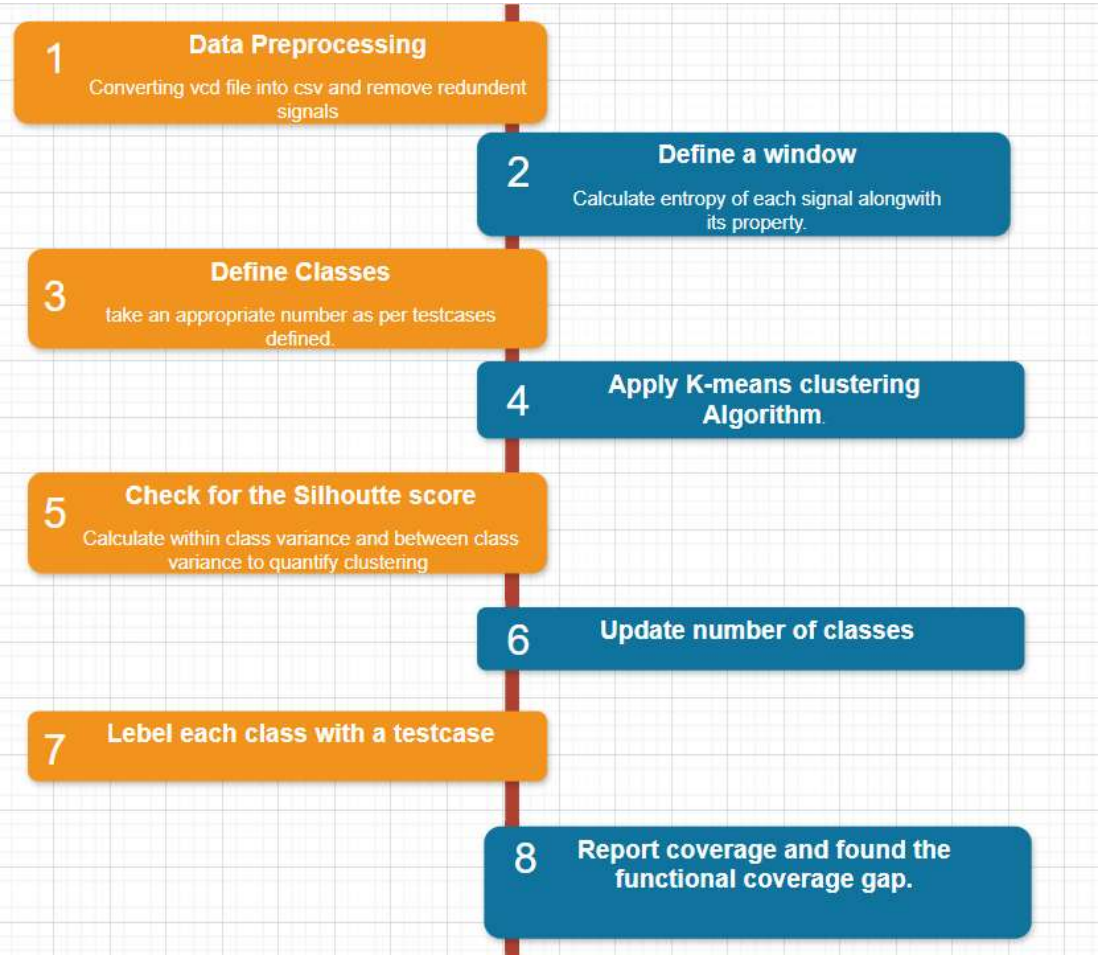
# How does Data look like

- VCD dump or value change dump is an o/p file generate by any EDA logic simulation tool.
- This file consist signal from the design and captures the transitions of those signals over a time duration or clock.




| Cycle | GCD.x | GCD.y[15:0] |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |
| 5 | 48 | 32 |
| 6 | 16 | 32 |
| 7 | 16 | 16 |
| 8 | 16 | 0 |
| 9 | 16 | 0 |
| 10 | 7 | 3 |
| 11 | 4 | 3 |
| 12 | 1 | 3 |
| 13 | 1 | 2 |
| 14 | 1 | 1 |
| 15 | 1 | 0 |
| 16 | 1 | 0 |
| 17 | 100 | 10 |
| 18 | 90 | 10 |
| 19 | 80 | 10 |
| 20 | 70 | 10 |

# Process


1
Data Preprocessing
Converting vcd file into csv and remove redundent signals
2
Define a window
Calculate entropy of each signal alongwith its property.
3
Define Classes
take an appropriate number as per testcases defined.
4
Apply K-means clustering Algorithm
5
Check for the Silhoutte score
Calculate within class variance and between class variance to quantify clustering
6
Update number of classes
7
Lebel each class with a testcase
8
Report coverage and found the functional coverage gap.

# Dataset/ Preprocessing

- Converts vcd file into a csv file, where we can have multiple columns with signal names, and populate these columns with the binary and asci values over each clock period.
- Eliminate signals which toggles over a test case rather over a transaction for example reset, test_done , start etc.
- Calculate entropy for all these filtered signals and choose a signal having least entropy. Lets call it window signal.
- Define a window size using this window signal and create a matrix of window size * No. filtered signals.
- Flatten these all matrix before giving it to k-means algorithm for clustering
- Now this data is preprocessed and ready for a clustering algorithm.

# ML Process

- Lets start with initial K value which is number of testcases defined.
- Select Initial centroids and apply k-means to assign data points to the clusters.
- Calculate within class variance and between class variance to quantify the clustering.
- Update value of k if silhouette score is beyond threshold boundary.
- Redo the clustering and keep updating no of classes to reach an optimal value.

# Post Processing

- Calculate the variance matrices with in a cluster.
- Pick up few samples with higher variance and draw timing diagram.
- Label each classes as per the timing diagram of sample matrices.
- Compare these labels with the test cases defined.
- Report the coverage with existing test bench.
- Determine the coverage gap

# Proof of concept

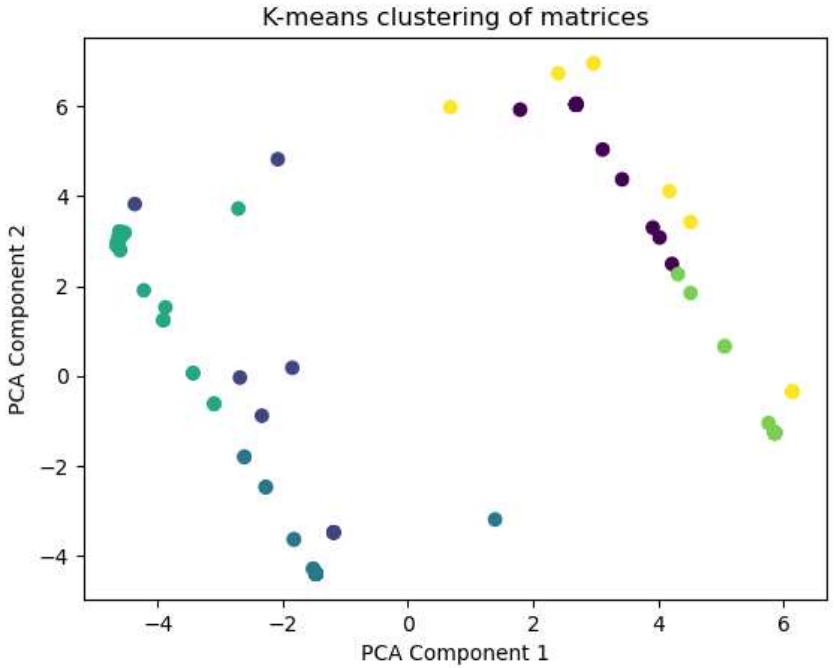
Input – Total no of matrices are 191 each of 64 clock samples.


Output- Clustered these matrices into 6 classes.

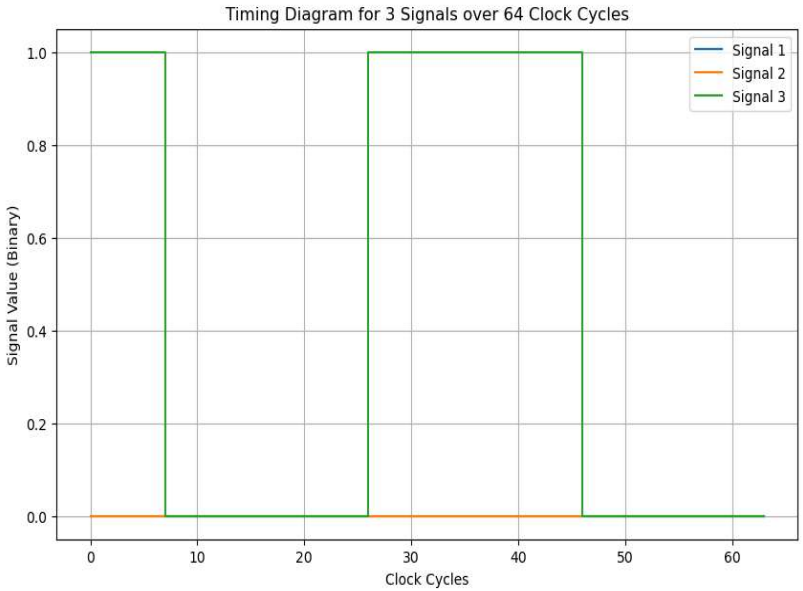
```
from sklearn.metrics import silhouette_score

# After fitting the KMeans model
score = silhouette_score(data_points, kmeans.labels_)
print(f"Silhouette Score: {score}")
```

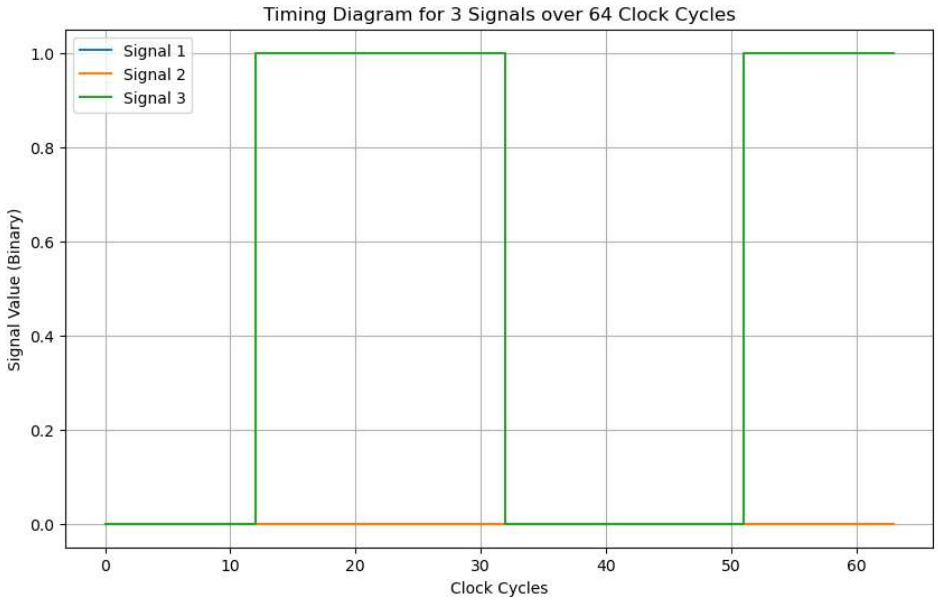
✓ 0.1s

```
Silhouette Score: 0.6830314934269549
```



# Post processing




These are 2 matrices of same class, which covers one IDLE test case of axi stream protocol where ready is toggling but tvalid and tlast are low.

# Challenges

VCD is a complicated format and converting it into csv was a challenge.

Deciding optimal window size requires understanding of data sample behaviour.

Defining threshold to quantify clustering.

Labelling of data requires manual effort.

# Usecase

- Functional coverage gaps are hard to find in large data set.
- Determine the cases covered with in a test case by just randomizing the input signals.
- Can eliminate the need of writing coverpoints to some extent.
- Reduces manual intervention and improves the quality of product.

# Future Scope for Next year

Enable dynamic and auto window detection.

Using it for custom logic.

Working with multi bit signals.

Working with complex designs having multiple levels/instances.

Improve the quality of classification.

Reducing manual efforts in labelling the classes.

Thank You