International Institute of Information Technology Bangalore

ASIC Based VLSI Architecture for Support Vector Machine

Arun M – MS, IIITB

Vaishnavi Sharma – MTech, IIITB

Supervisor: Prof. Madhav Rao

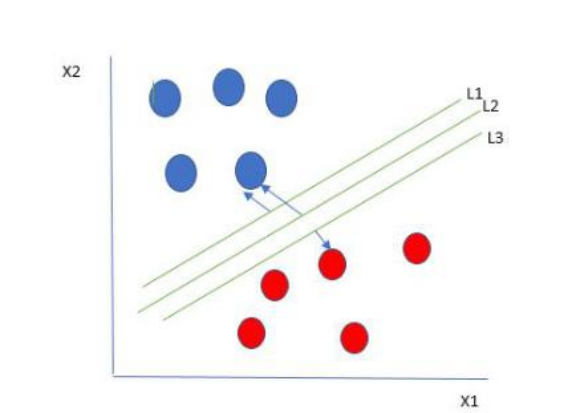
ABSTRACT/ PROBLEM STATEMENT

- This project presents an ASIC-based Support Vector Machine (SVM) inference engine designed to classify apnea events using physiological signals such as ECG. Unlike software-based implementations, which are computationally expensive and power-hungry, this ASIC provides a low-power, high-speed alternative tailored for real-time classification. The architecture is optimized for efficient SVM kernel computation and decision functions, ensuring minimal latency while maintaining high classification accuracy. A key feature of this design is its scalability, enabling the same hardware framework to be extended beyond sleep apnea detection to classify multiple datasets from different domains. The ASIC efficiently processes input features, performs classification using the SVM model, and outputs real-time predictions, making it suitable for deployment in embedded and wearable healthcare devices. Key challenges addressed in this project include hardware optimization, feature extraction, resource-efficient implementation of SVM computations, and scalability. The inference engine has been tested with different datasets to demonstrate its flexibility, showing promising results in medical and non-medical classification tasks.

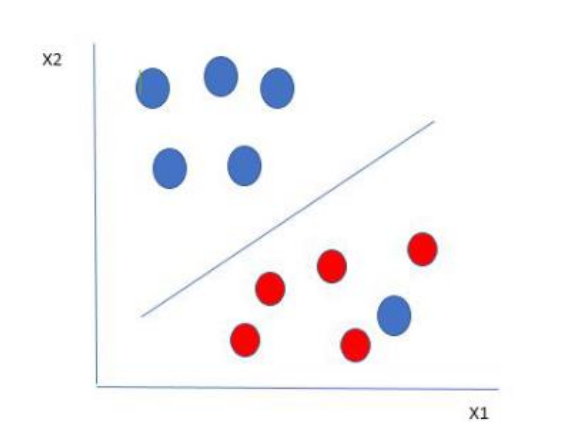
SUPPORT VECTOR MACHINE

- Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification tasks.
- SVM aims to find the optimal hyperplane in an N-dimensional space to separate data points into different classes. The algorithm maximizes the margin between the closest points of different classes.
- Support Vectors are the closest data points to the hyperplane, crucial for determining the hyperplane and margin in SVM.
- Hyperplane is a decision boundary separating different classes in feature space, represented by the equation

  wx + b = 0 in linear classification.



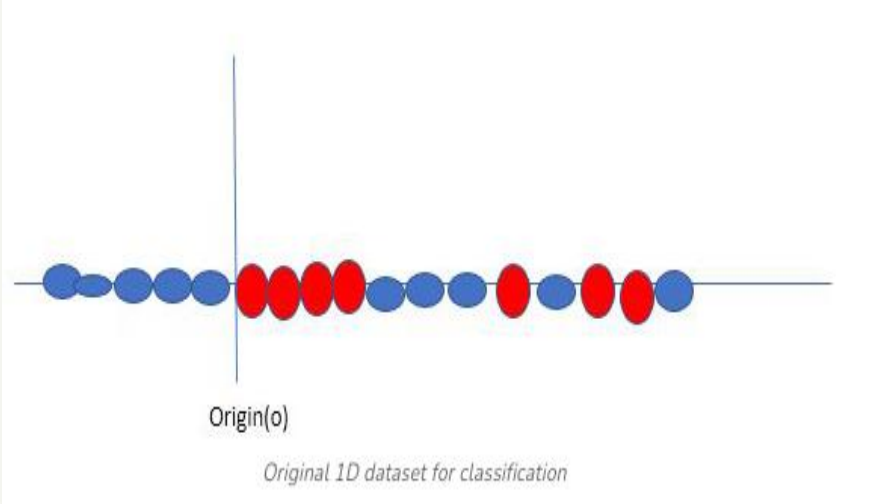
Multiple hyperplanes separate the data from two classes



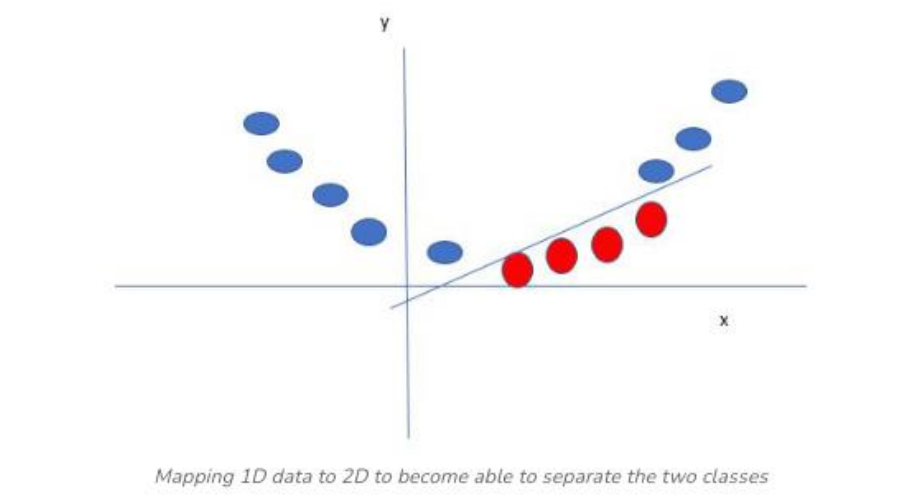
Hyperplane which is the most optimized one

SUPPORT VECTOR MACHINE (contd.)

- When data is not linearly separable (i.e., it can't be divided by a straight line), SVM uses a technique called **kernels** to map the data into a higher-dimensional space where it becomes separable. This transformation helps SVM find a decision boundary even for non-linear data.

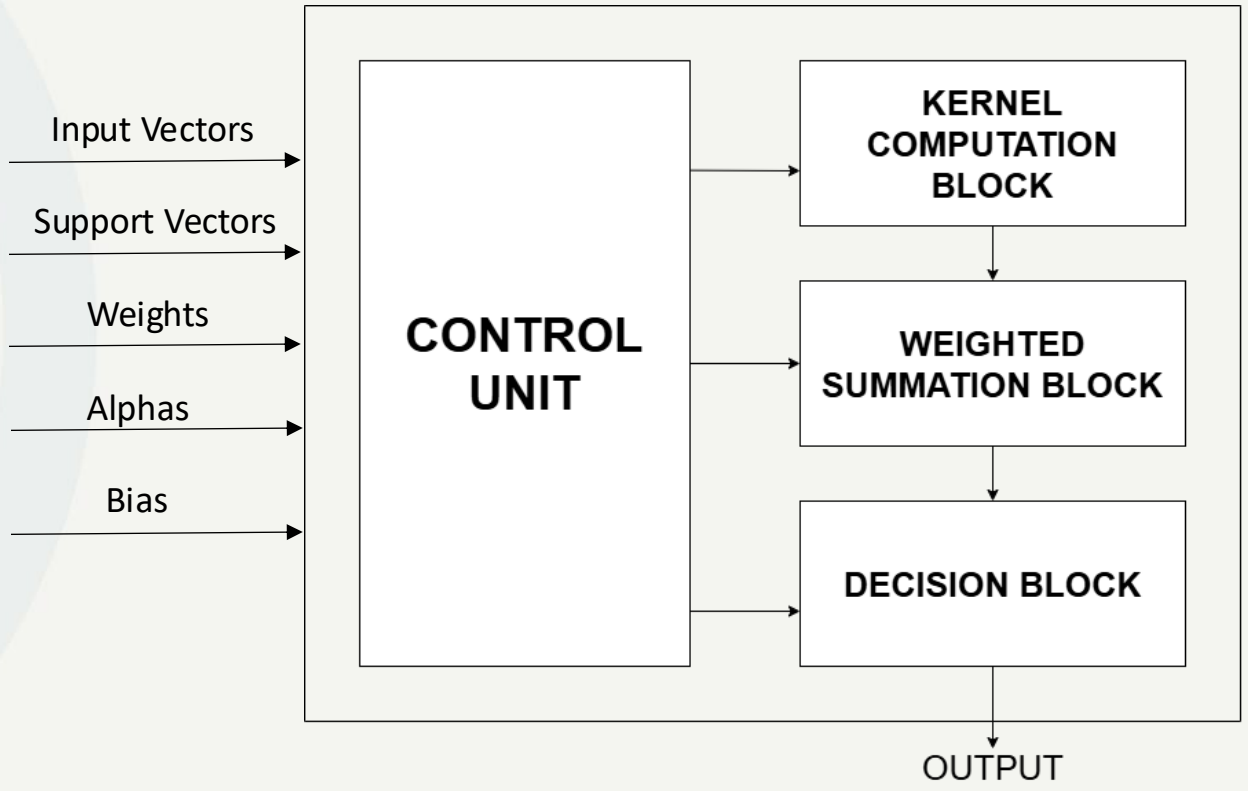


Original 1D dataset for classification



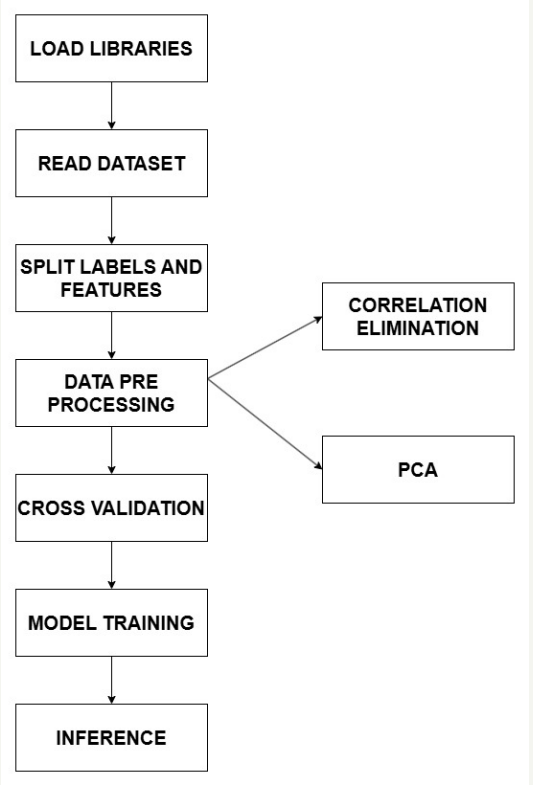
Mapping 1D data to 2D to become able to separate the two classes

BLOCK DIAGRAM



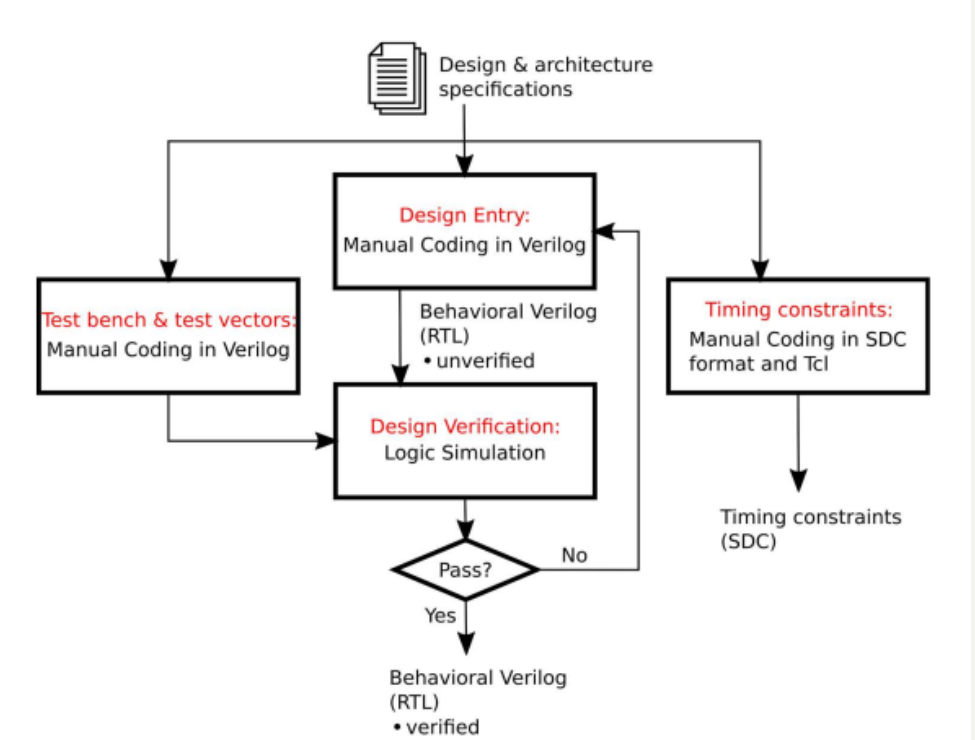
METHODOLOGY

The methodology can be divided into: software and hardware phase

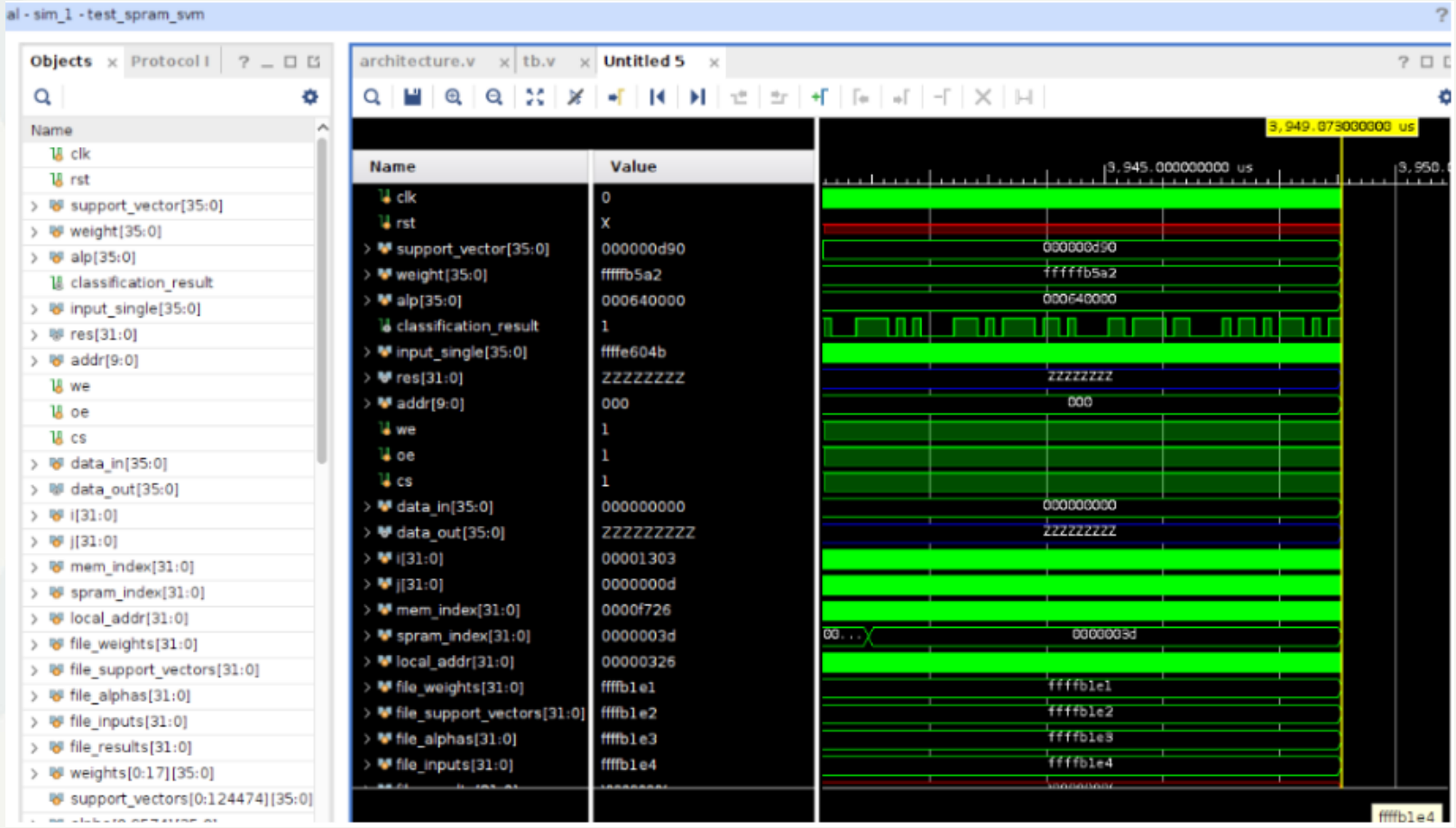


- Through correlation elimination method, number of features were reduced from 66 to 18.
- Using PCA, it was further reduced to 13.

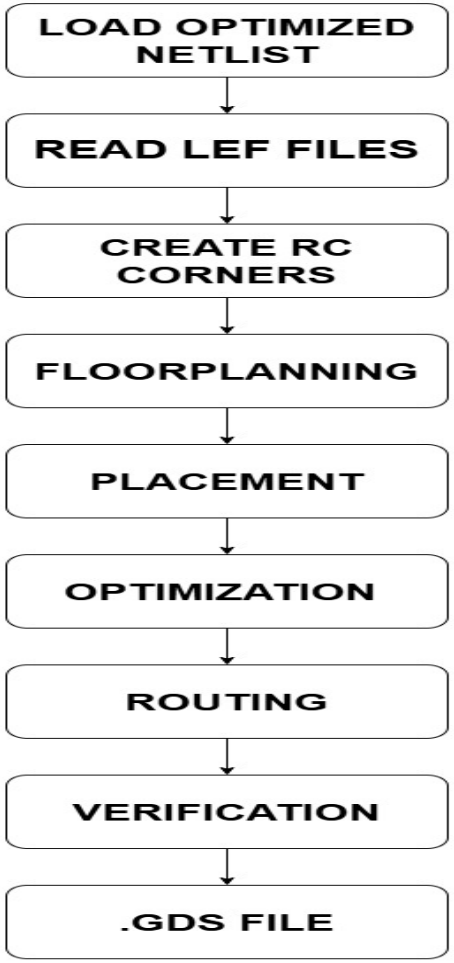
METHODOLOGY (contd.)



SIMULATION


al - sim_1 - test_spram_svm
Objects
Protocol I
Name
clk
rst
support_vector[35:0]
weight[35:0]
alp[35:0]
classification_result
input_single[35:0]
res[31:0]
addr[9:0]
we
oe
cs
data_in[35:0]
data_out[35:0]
i[31:0]
j[31:0]
mem_index[31:0]
spram_index[31:0]
local_addr[31:0]
file_weights[31:0]
file_support_vectors[31:0]
file_alphas[31:0]
file_inputs[31:0]
file_results[31:0]
weights[0:17][35:0]
support_vectors[0:124474][35:0]
architecture.v
tb.v
Untitled 5
Name
Value
clk 0
rst x
support_vector[35:0] 000000d90
weight[35:0] fffffb5a2
alp[35:0] 000640000
classification_result 1
input_single[35:0] ffffe604b
res[31:0] zzzzzzzz
addr[9:0] 000
we 1
oe 1
cs 1
data_in[35:0] 000000000
data_out[35:0] zzzzzzzzz
i[31:0] 00001303
j[31:0] 0000000d
mem_index[31:0] 0000f726
spram_index[31:0] 0000003d
local_addr[31:0] 00000326
file_weights[31:0] ffffb1e1
file_support_vectors[31:0] ffffb1e2
file_alphas[31:0] ffffb1e3
file_inputs[31:0] ffffb1e4
3,949.875000000 us
3,945.000000000 us
ffffb1e4

PHYSICAL DESIGN (contd.)



- Synthesis of the design was done using Cadence Genus
- The entire physical design flow was performed in Cadence Innovus using custom scripts for 45nm technology node
- All Violations were checked in accordance to 45nm rule files.

PHYSICAL DESIGN (contd.)

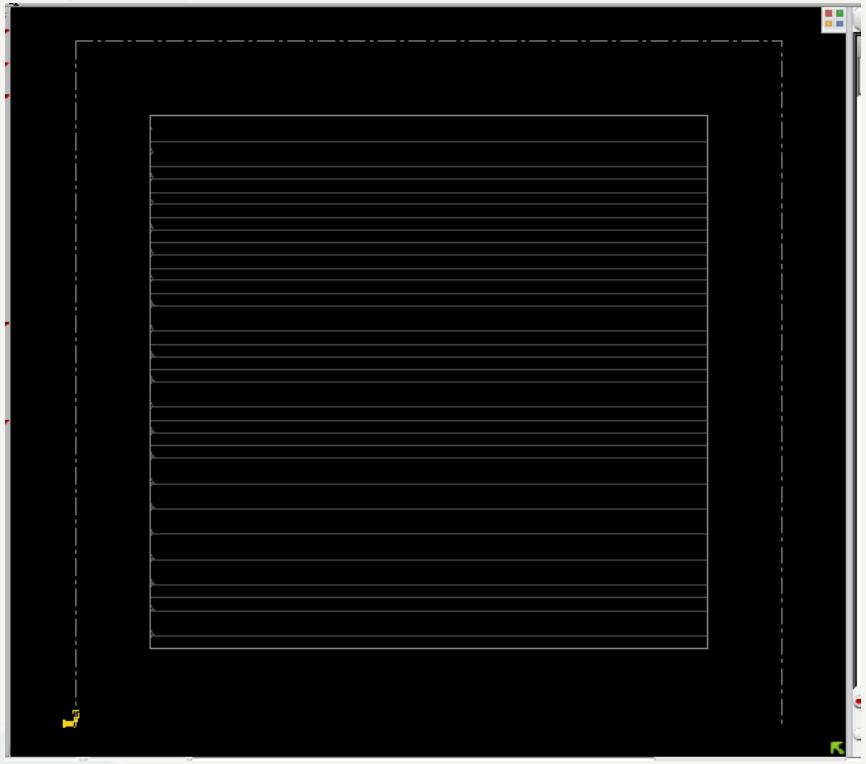


Fig: Floor planning

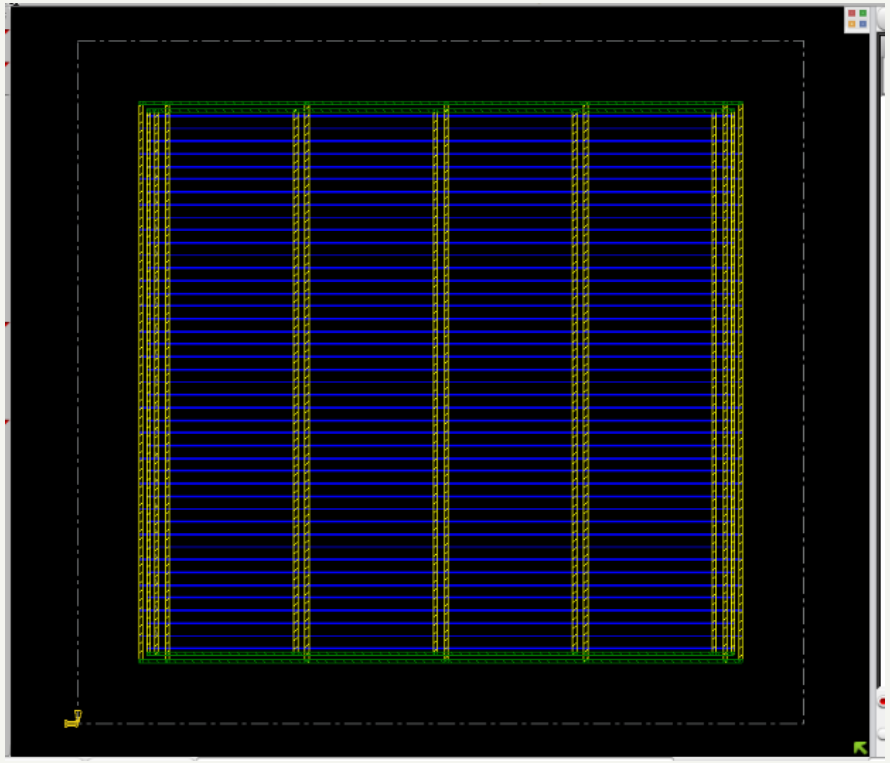


Fig: Power planning

PHYSICAL DESIGN (contd.)

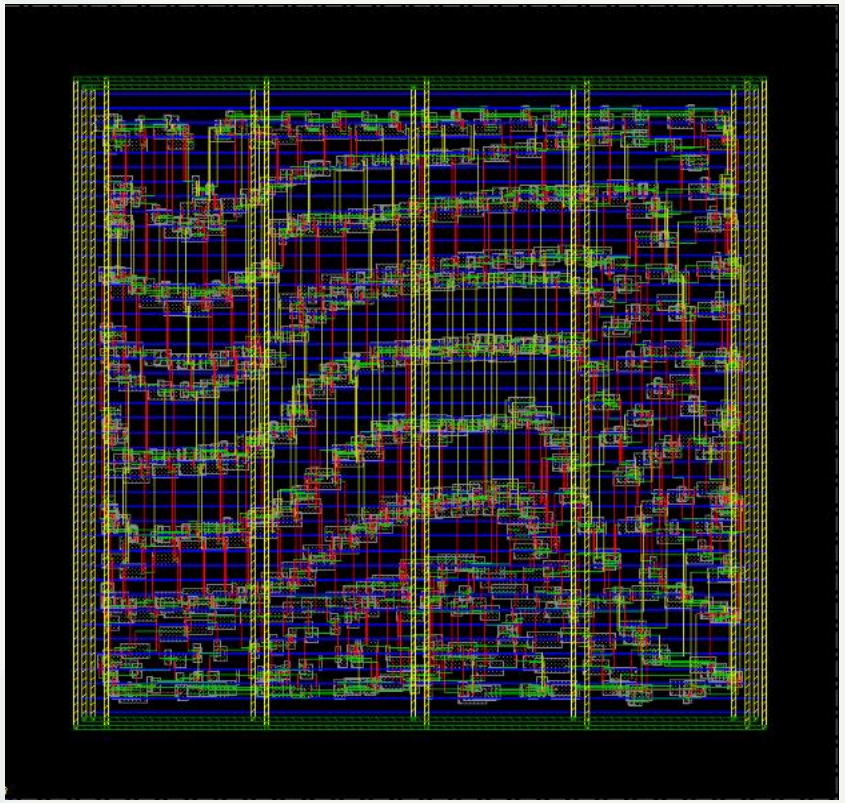


Fig: Standard Cell Placement

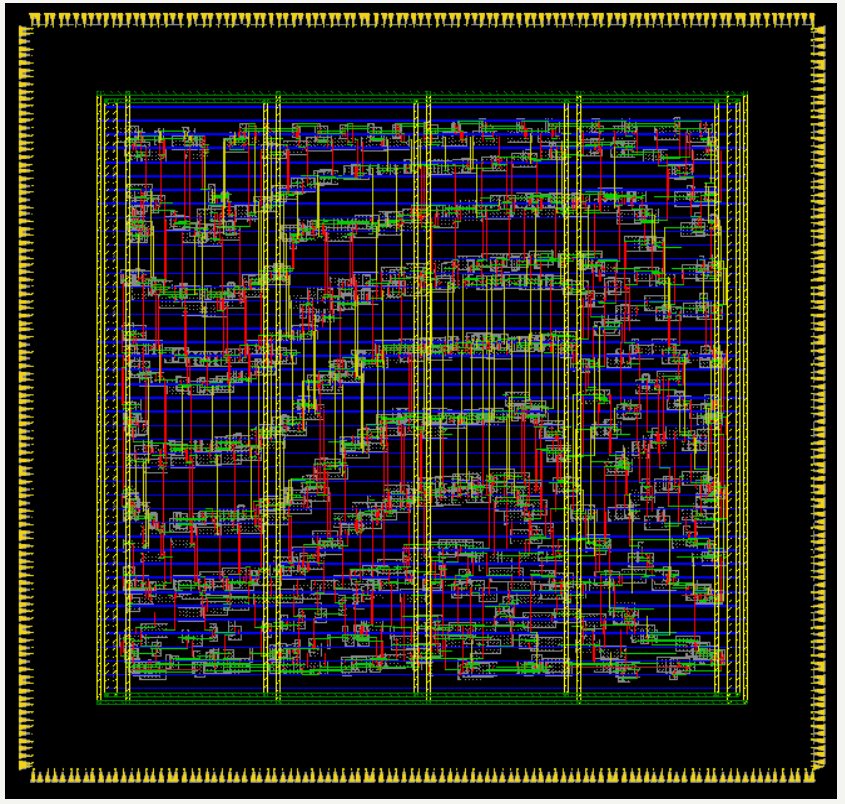


Fig: Pin Placement

PHYSICAL DESIGN (contd.)

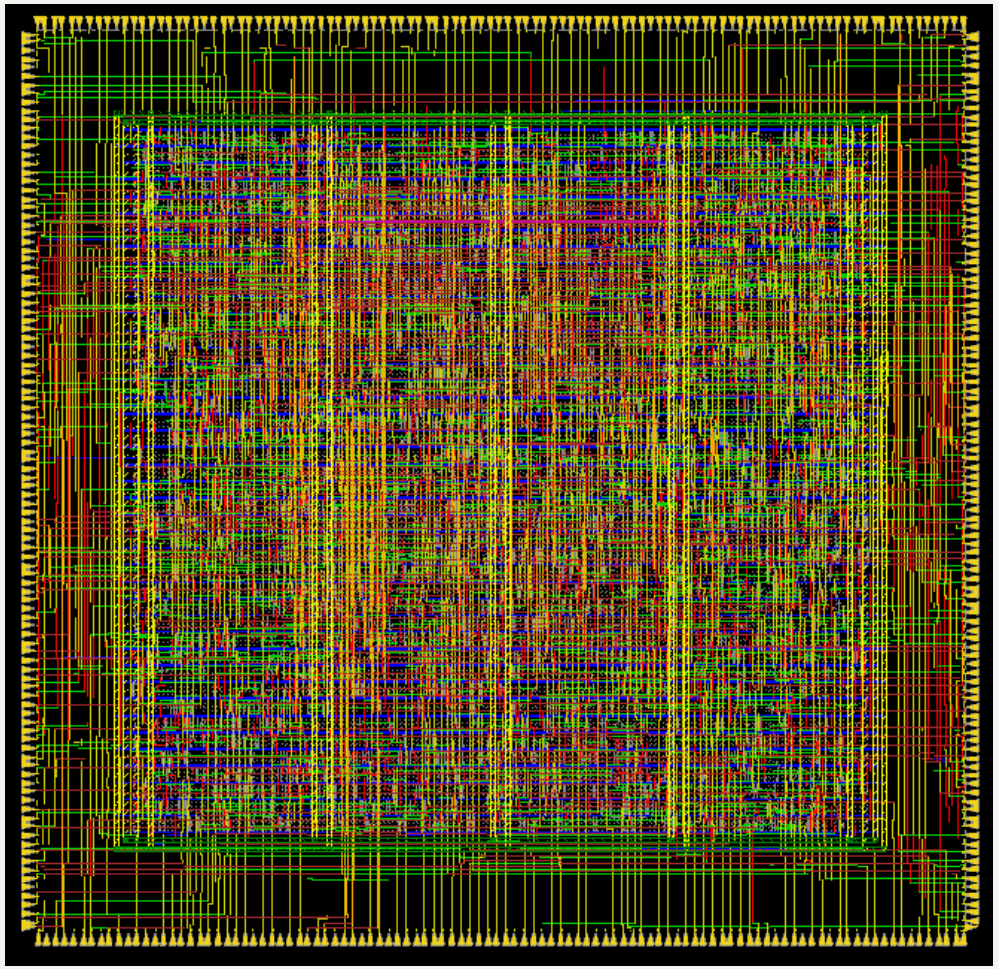


Fig: Final Stage (after routing)

RESULTS

1) Synthesis :

The design was synthesized in Cadence Genus using 45nm technology node and the following results were observed.

Parameter	Value
Area	5762.152 um^2
Gates	1349
Power	234.968 nW

RESULTS (contd.)

2) DRC Violations:

```
#-report svm_architecture.drc.rpt        # string, default="", user setting
 *** Starting Verify DRC (MEM: 11953.1) ***

  VERIFY DRC ..... Starting Verification
  VERIFY DRC ..... Initializing
  VERIFY DRC ..... Deleting Existing Violations
  VERIFY DRC ..... Creating Sub-Areas
**WARN: (IMPVFG-1198):  The number of CPUs requested 24 is larger than that verify
 Use 'setMultiCpuUsage -localCpu' to specify the less cup nunber if the verify are
  VERIFY DRC ..... Using new threading
 VERIFY DRC ..... Sub-Area: {170.880 168.960 255.000 252.700} 9 of 9  Thread : 3
 VERIFY DRC ..... Sub-Area: {170.880 0.000 255.000 84.480} 3 of 9  Thread : 1
 VERIFY DRC ..... Sub-Area: {0.000 0.000 85.440 84.480} 1 of 9  Thread : 0
 VERIFY DRC ..... Sub-Area: {0.000 168.960 85.440 252.700} 7 of 9  Thread : 2
 VERIFY DRC ..... Sub-Area: {170.880 84.480 255.000 168.960} 6 of 9  Thread : 7
 VERIFY DRC ..... Sub-Area: {0.000 84.480 85.440 168.960} 4 of 9  Thread : 2
 VERIFY DRC ..... Sub-Area: {85.440 0.000 170.880 84.480} 2 of 9  Thread : 2
 VERIFY DRC ..... Sub-Area: {85.440 168.960 170.880 252.700} 8 of 9  Thread : 7
 VERIFY DRC ..... Thread : 7 finished.
 VERIFY DRC ..... Thread : 1 finished.
 VERIFY DRC ..... Thread : 3 finished.
 VERIFY DRC ..... Thread : 2 finished.
 VERIFY DRC ..... Thread : 4 finished.
 VERIFY DRC ..... Thread : 5 finished.
 VERIFY DRC ..... Thread : 0 finished.
 VERIFY DRC ..... Thread : 8 finished.
 VERIFY DRC ..... Sub-Area: {85.440 84.480 170.880 168.960} 5 of 9  Thread : 6
 VERIFY DRC ..... Thread : 6 finished.

  Verification Complete : 0 Viols.
```

RESULTS (contd.)

3) Connectivity and Antenna Violations:

```
******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:01.2  MEM: 0.000M)

innovus 157>  *** Starting Verify Antenna (MEM: 12724.0) ***

Report File: svm_architecture.antenna.rpt
  VERIFY Antenna ..... Starting Verification
  VERIFY Antenna ..... Using new threading
 VERIFY Antenna..... Thread : 5 finished.
 VERIFY Antenna..... Thread : 23 finished.
 VERIFY Antenna..... Thread : 8 finished.
 VERIFY Antenna..... Thread : 19 finished.
 VERIFY Antenna..... Thread : 4 finished.
 VERIFY Antenna..... Thread : 15 finished.
 VERIFY Antenna..... Thread : 11 finished.
 VERIFY Antenna..... Thread : 21 finished.
 VERIFY Antenna..... Thread : 22 finished.
 VERIFY Antenna..... Thread : 7 finished.
 VERIFY Antenna..... Thread : 17 finished.
 VERIFY Antenna..... Thread : 6 finished.
 VERIFY Antenna..... Thread : 13 finished.
 VERIFY Antenna..... Thread : 10 finished.
 VERIFY Antenna..... Thread : 3 finished.
 VERIFY Antenna..... Thread : 9 finished.
 VERIFY Antenna..... Thread : 18 finished.
 VERIFY Antenna..... Thread : 2 finished.
 VERIFY Antenna..... Thread : 20 finished.
 VERIFY Antenna..... Thread : 1 finished.
 VERIFY Antenna..... Thread : 16 finished.
 VERIFY Antenna..... Thread : 12 finished.
 VERIFY Antenna..... Thread : 0 finished.
 VERIFY Antenna..... Thread : 14 finished.
Verification Complete: 0 Violations
 *** End Verify Antenna (CPU: 0:00:02.7  ELAPSED TIME: 0.00  MEM: 0.0M) ***

******* DONE VERIFY ANTENNA ********
```

RESULTS (contd.)

4) Setup and Hold Analysis:

Clock constraints: 10.3 ns

Maximum operating frequency: 97.08 MHz

```
Path 1: MET Setup Check with Pin kernel_kernel_result_reg[31]/CK
Endpoint:   kernel_kernel_result_reg[31]/D (^) checked with  leading edge of
'clk'
Beginpoint: X[96]                          (v) triggered by  leading edge of '@'
Path Groups: {clk}
Analysis View: setup_analysis
Other End Arrival Time          0.015
- Setup                         0.126
+ Phase Shift                  10.300
- Uncertainty                   0.010
= Required Time                10.179
- Arrival Time                  9.929
= Slack Time                    0.250
```

Fig: Setup Time

```
Path 1: MET Hold Check with Pin sum_module_decision_value_reg[0]/CK
Endpoint:   sum_module_decision_value_reg[0]/D (v) checked with  leading edge
of 'clk'
Beginpoint: alpha[0]                           (v) triggered by  leading edge
of '@'
Path Groups: {clk}
Analysis View: hold_analysis
Other End Arrival Time          0.012
+ Hold                          0.001
+ Phase Shift                   0.000
+ Uncertainty                   0.010
= Required Time                 0.023
  Arrival Time                  0.116
  Slack Time                    0.092
```

Fig: Hold Time

CHECKING WITH OTHER DATASETS

The same SVM engine was tried on various datasets so as to check the generic property of the inference engine.

Dataset	Support Vectors	No of features	No of Classes	Software accuracy	Hardware accuracy
Sleep Apnea	9575	13	2	79.8	77.4
Iris	9	5	3	100	73.1
Wine Quality	1069	11	6	56	42
Movie	216	16	2	69.7	58.65

From the table, it is clear the design does work for various classification datasets. However, it depends on number of support vectors since they decide the decision boundaries and therefore right number of classifications. It also needs a lot of improvements when it comes to multiclass classifications.

FUTURE IMPROVEMENTS

- Re-modelling the design to work for Posit format. This will help in working with large datasets using lesser bits and hence, lesser hardware resources.
- Improving the kernel computation block to support polynomial, RBF and sigmoid functions.
- Improving the design's performance with multi-class classification problems

International Institute of Information Technology Bangalore

26/C, Electronics City, Hosur Road,

Bengaluru – 560 100, Karnataka, India

www.iiitb.ac.in

https://www.facebook.com/IIITBofficial/

https://twitter.com/IIITB_official

https://in.linkedin.com/school/iiitbofficial/

https://www.youtube.com/user/iiitbmedia

https://www.instagram.com/iiitb_official/