

Analog VLSI implementation of Support Vector Machine Learning and Classification (VLSI Project Report)

Agastya Seth, Suchit Jain

May 2019

1 Introduction

The research paper followed for the analysis is "Analog VLSI implementation of Support Vector Machine Learning and Classification" from the publication "2008 IEEE International Symposium on Circuits and Systems". The paper proposes an analog VLSI approach to implementing projection neural networks adapted for support vector machine with radial-basis function (RBF) kernel.

2 Literature Review

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

2.1 The mathematical problem

Constrained Quadratic Programming for SVC (Support Vector Classifier) is defined as following:

Given the training set $\{(x_i, y_i), i = 1, \dots, N\}$ drawn iid (independent and identically distributed) from a distribution $P(X, Y)$, where $x_i \in X \subseteq R^n$, input pattern for output $y_i \in Y \subseteq \{+1, -1\}$

The objective of SVC learning is to obtain optimal classifier to approximate P .

For non-linear classification, X is mapped to a higher dimension space $Z \subseteq R^m$ by $\gamma : X \rightarrow Z$ with $m \geq n$

This problem is formulated into a dual problem:

Dual CQP,

$$\min_{\alpha} \left[\frac{1}{2} \alpha^T Q \alpha - e^T \alpha \right]$$

subject to,

$$\begin{aligned} 0 &\leq \alpha \leq C \\ \alpha^T y &= 0 \end{aligned}$$

where, α is multiplier vector, each element of Q is $q_{ij} = y_i y_j k(x_i, x_j)$, $k(x_i, x_j)$ being the kernel function and $e = [1, 1, \dots, 1]^T$

The classifier thus becomes,

$$h_{\alpha, b}(x) = \text{sign} \left[\sum_{i=1}^N \alpha_i y_i k(x, x_i) + b \right]$$

the problem is thus basically solving for the optimal value of α^* and b^*

The constrained quadratic problem formulated above can be solved using a set of ordinary differential equations (ODE's):

$$\begin{aligned} \frac{d\alpha}{dt} &= \lambda \{ P_{\Omega_0}(\alpha - Q\alpha - by + e) - \alpha \} \\ \frac{db}{dt} &= \lambda (y^T \alpha) \end{aligned}$$

here, Ω_0 is called the projection operator given by,

$$\Omega_0 = \{u = (\alpha^T, b)^T \mid 0 \leq \alpha \leq C, b \in R\}$$

3 Motivation/Backdrop of the study

Support Vector Machine (SVM) is a considered to be one of the most powerful machine learning algorithm for classification, generalizing well even in sparse, high dimension settings. However, due to its constrained quadratic programming (CQP) problem it is usually very time consuming to implement. Using analog VLSI circuits to implement neural network solving SVM CQP can be a good solution because:

1. Computation can be carried out in a distributed and parallel fashion.
2. Efficient use of non-linearity of the silicon devices.
3. Can be integrated with sensor interface circuits, resulting in highly efficient smart sensor system without employing any power hungry Analog-to-Digital Convertor (ADC).

4 Objective

The objective was to recreate the circuits, provide inputs and analyze the classification performance as performed in the research paper. Further we aim to analyze the performance, perform TT corner analysis and finally perform process corner analysis for each block. However, since floating-gate CMOS cannot be implemented on virtuoso we focus on the analysis of the project block, since the simulation data for the other blocks could not be generated (without the floating-bump circuit).

5 Related Circuit and Design Methodology

5.1 Realization in VLSI

Each of the components of the above ODE's are realized using different circuits:

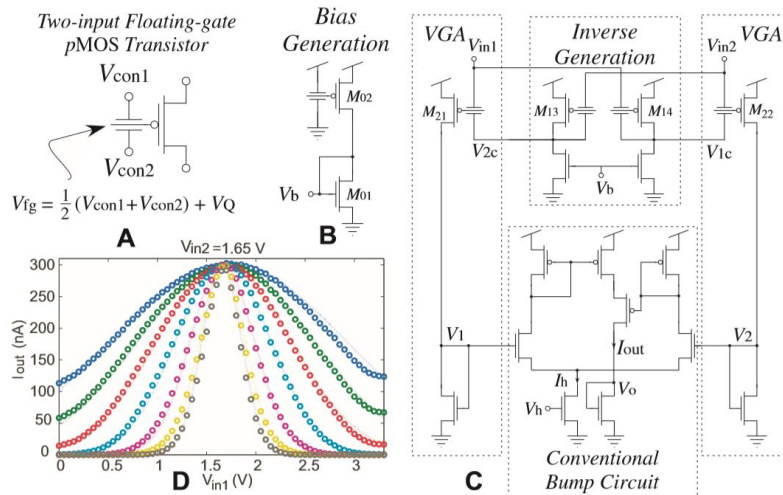
$$\frac{d\alpha}{dt} = \lambda \{ P_{\Omega_0}(\alpha - \overbrace{Q\alpha}^{KernelBlock} - \overbrace{by}^{CurrentDecisionControl(CDC)Block} + e) - \alpha \}$$

$$\frac{db}{dt} = \lambda(\overbrace{y^T \alpha}^{CurrentDecisionControl(CDC)Block})$$

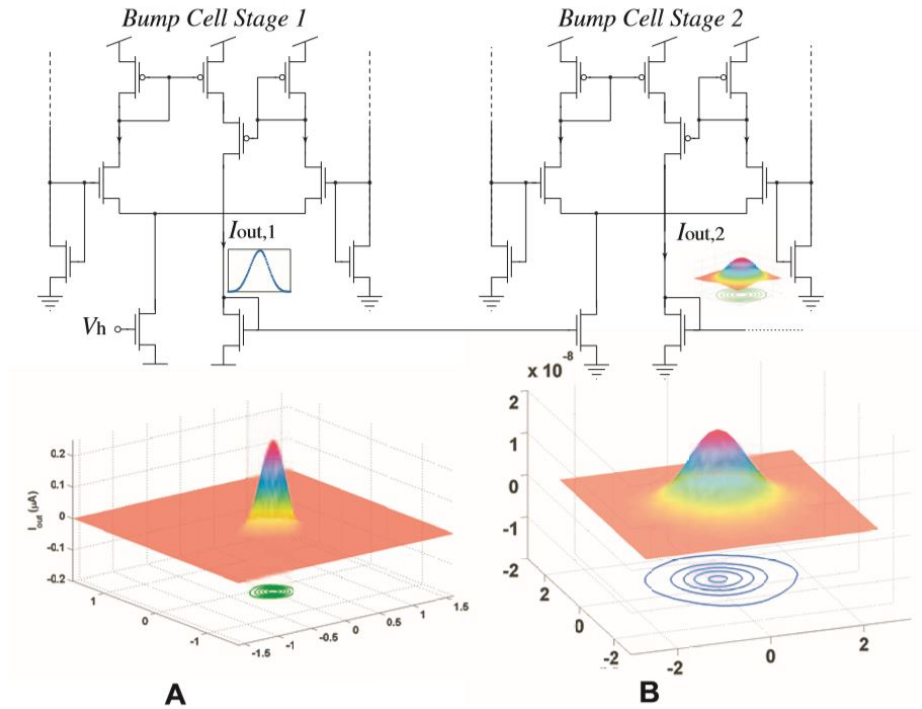
$$\Omega_0 = \{ u = (\alpha^T, b)^T | 0 \leq \alpha \leq C, b \in R \}$$

The state variables, α_i and b are set to be the current signals I_{α_i} and I_b . y_i 's are digital signals. I_{α_i} is fed to the tail current of the first stage bump circuit and then a digital *XNOR* is used to decide the polarity of the resulting current. The components of *+ve* and *-ve* currents can be easily summed by KCL. The resulting output current can be obtained by subtracting the *-ve* current from the *+ve* current using a current mirror.

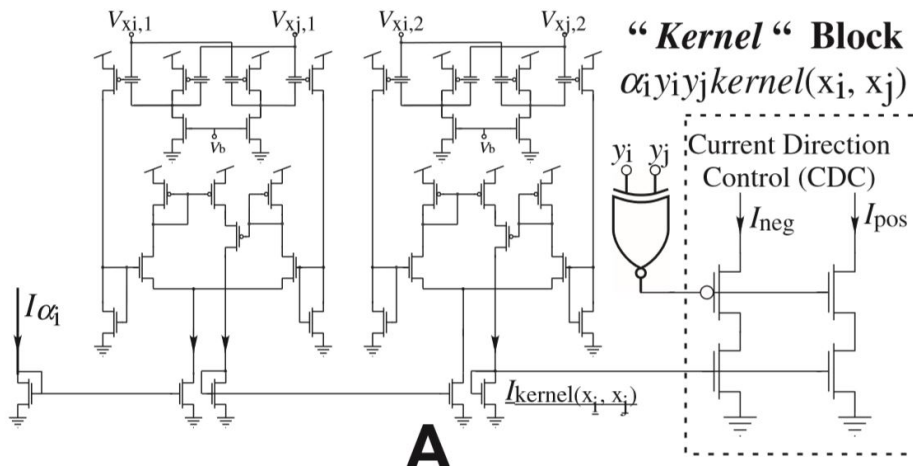
Radian-Basis Function (RBF) is chosen as the *kernel function*, $k(x_i, x_j)$ and is realized by the output current of the floating-gate bump circuit. The height of the transfer curve is controlled by the tail current I_h . The width of the bump-shaped transfer curve is adjusted by programming the common mode-charge on the floating-gate transistor M_{21} and M_{22} .



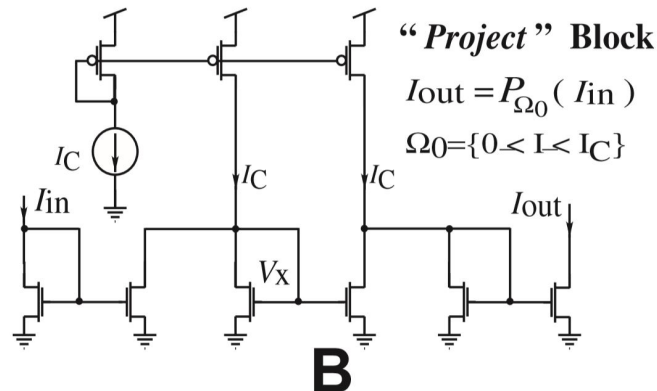
A multivariate gaussian function (a 3D bump) can be implemented by cascading the proposed bump circuits:



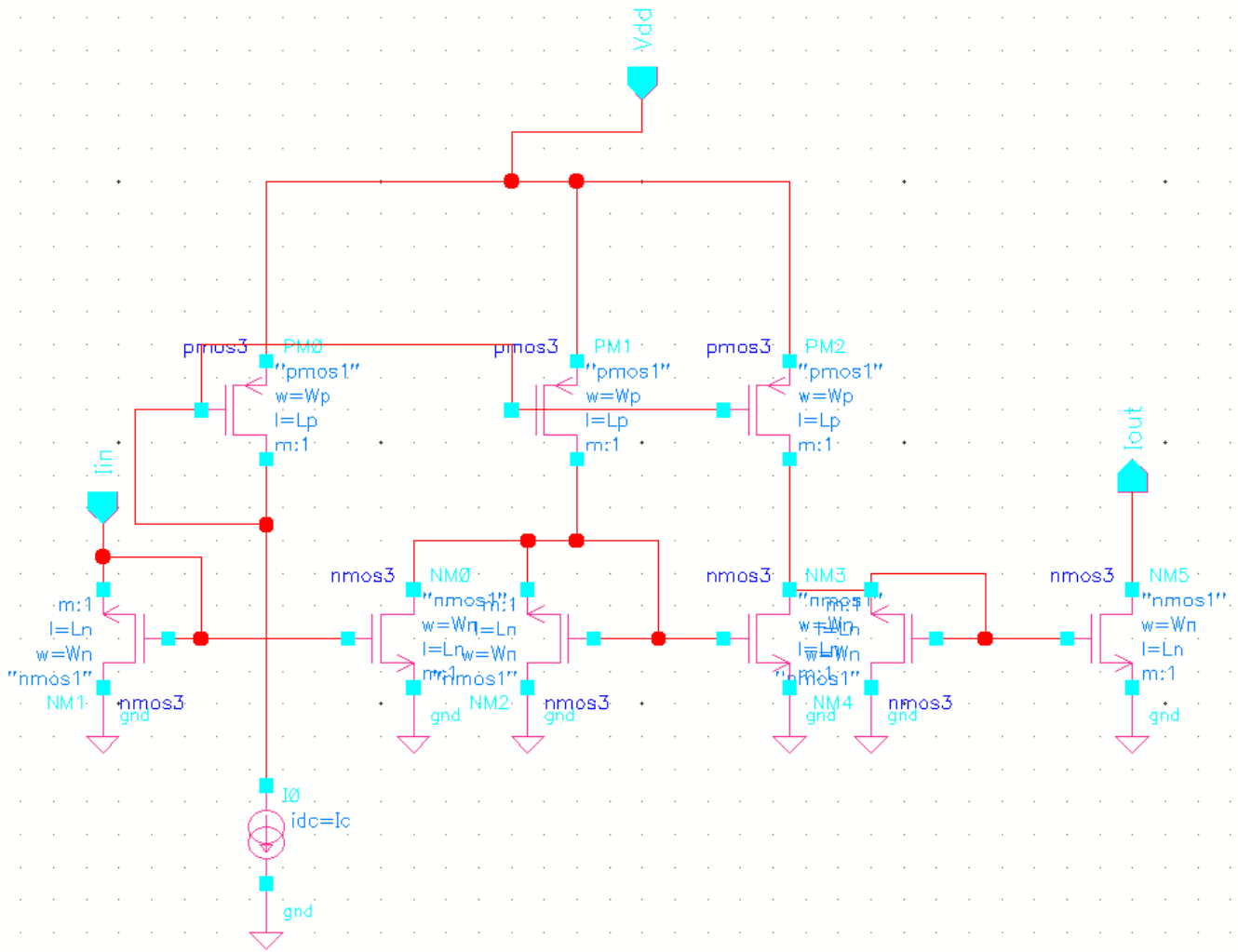
The circuit to estimate $Q\alpha$ is named the **Kernel Block**, which has inclusion of a Current Decision Control (CDC) block at the output of a floating-gate bump circuit.



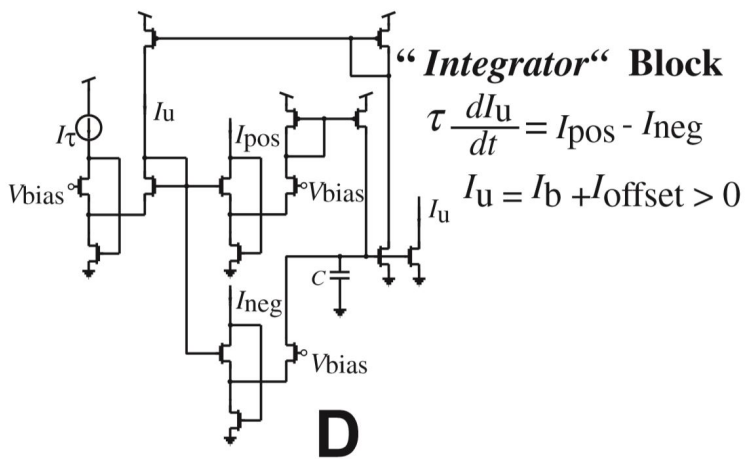
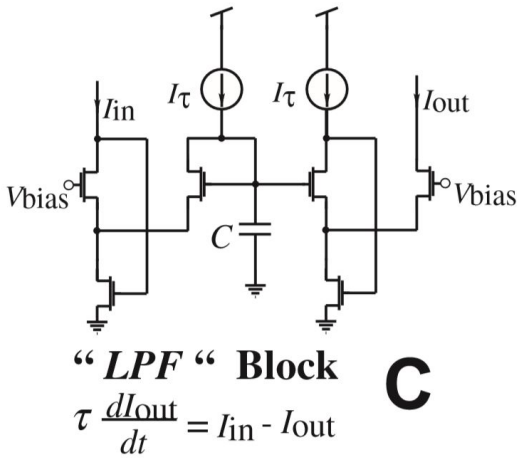
The **Project Block** is a current mirror based circuit used to implement P_{Ω_0} : if $I_{in} < I_c$, then $I_{out} = I_{in}$ if $I_{in} > I_c$, then $I_{out} = I_c$, i.e. the output current would be clamped at I_c . This is because the node V_x will be pulled to ground, thus I_c being clamped at the output side.



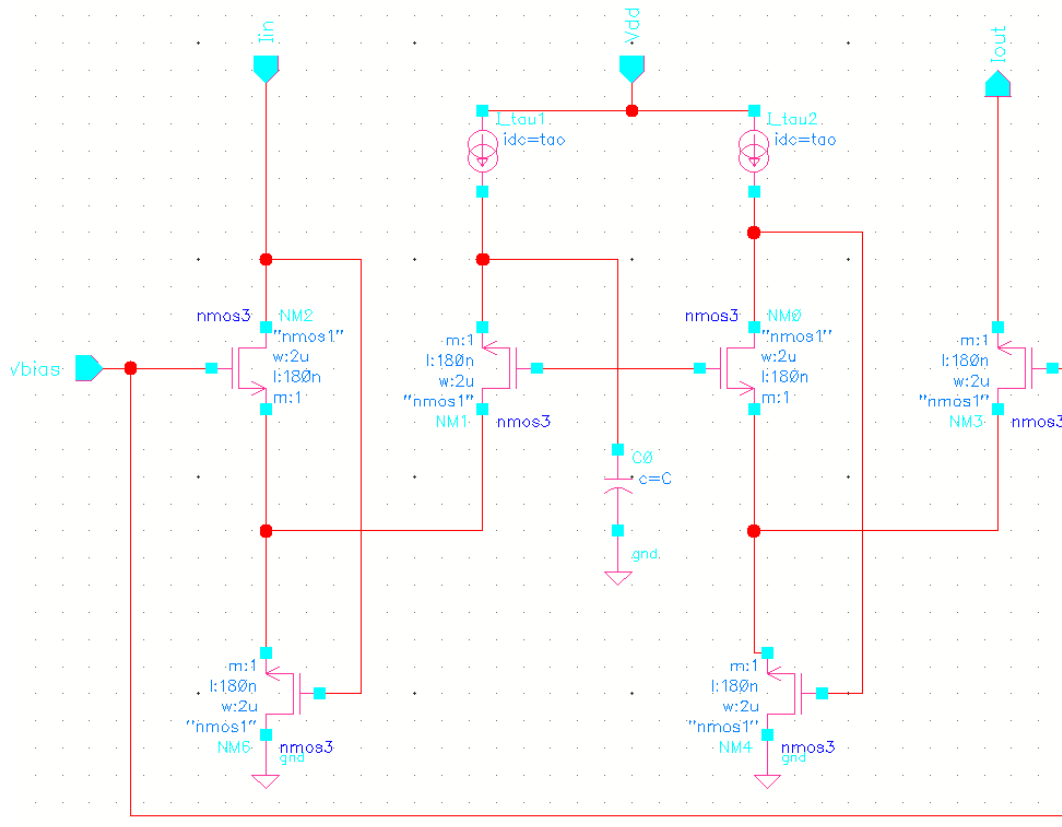
Virtuoso implementation of the project block



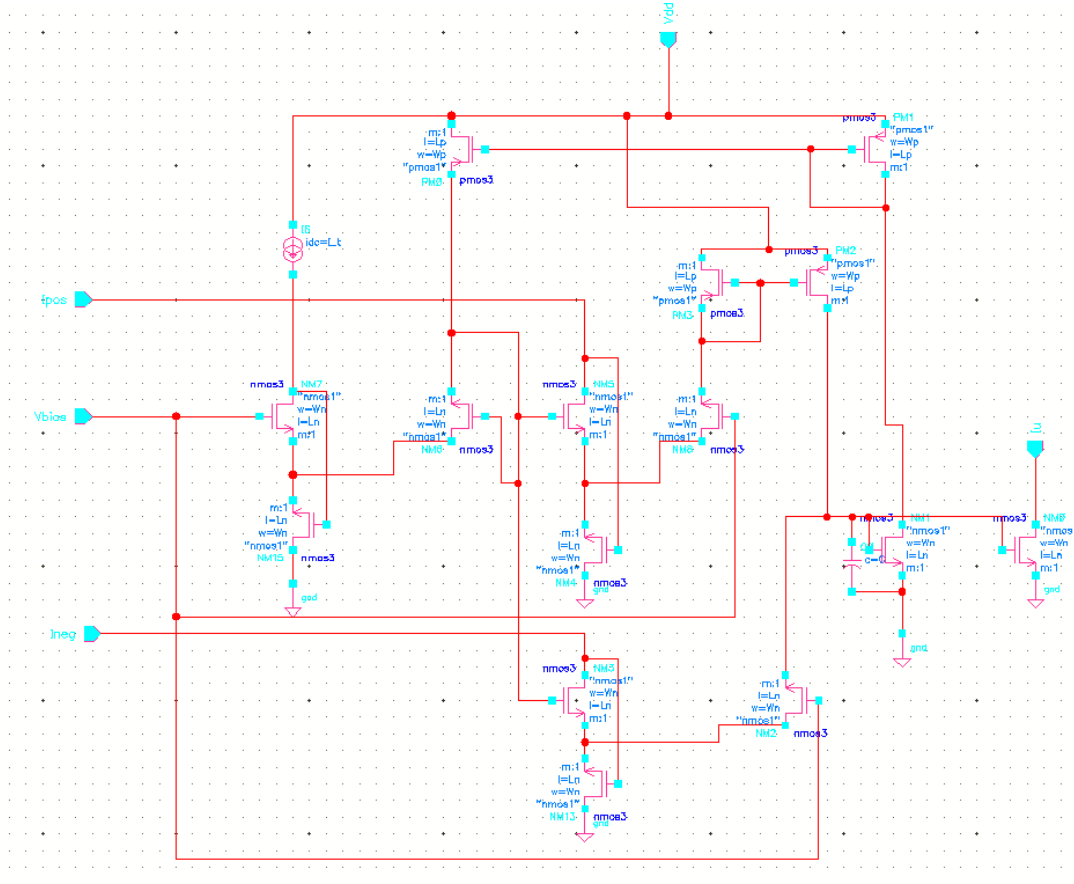
To complete the computations of the dual CQP problem a translinear **Low-Pass-Filter (LPF) Block** and a translinear **Integrator Block** with MOS transistors operating in the sub-threshold region are used:



Virtuoso implementation of the LPF block

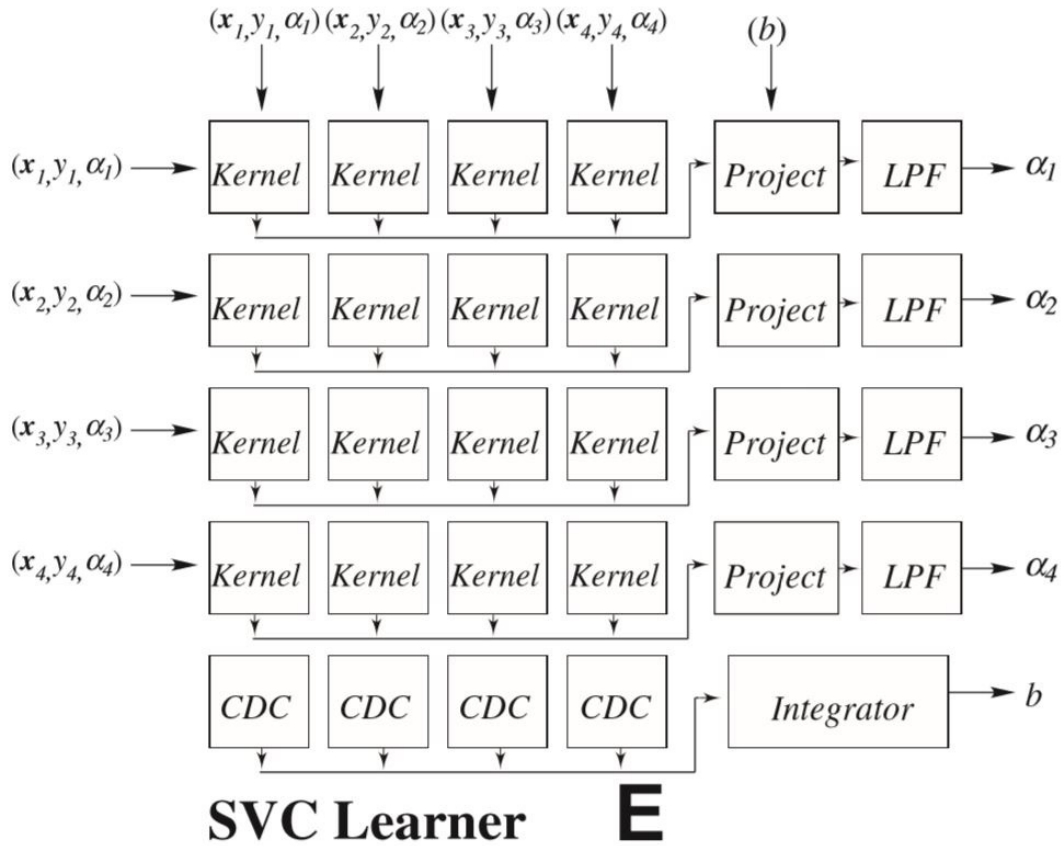


Virtuoso implementation of the Integrator block



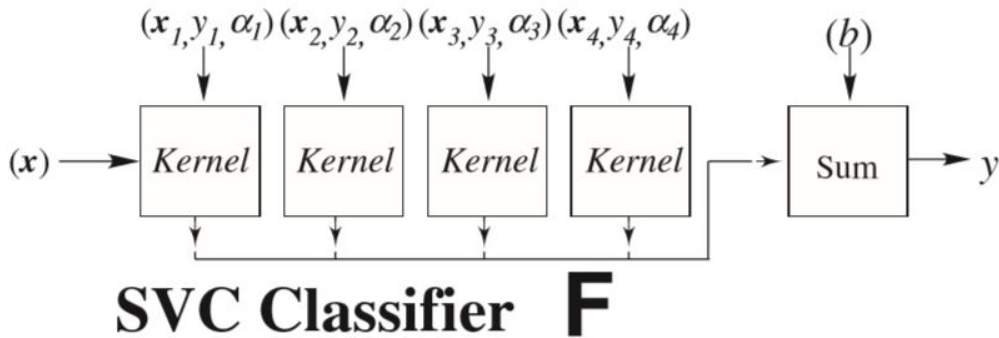
Since b is bipolar (can be $= ve$ or $-ve$), a current offset is added to keep the output current in the *Integrator Block* $+ve$. This offset is then subtracted before the project block.

For the *SVC learning process* all the components are bought together as in the circuit given below:



The learning process involves the computation of the values of optimal α and b .

The circuit to implement the *svc classifier* is same as the kernel block minus the *XNOR* circuit. The sign of the output classifier current of output node tells the classification result:



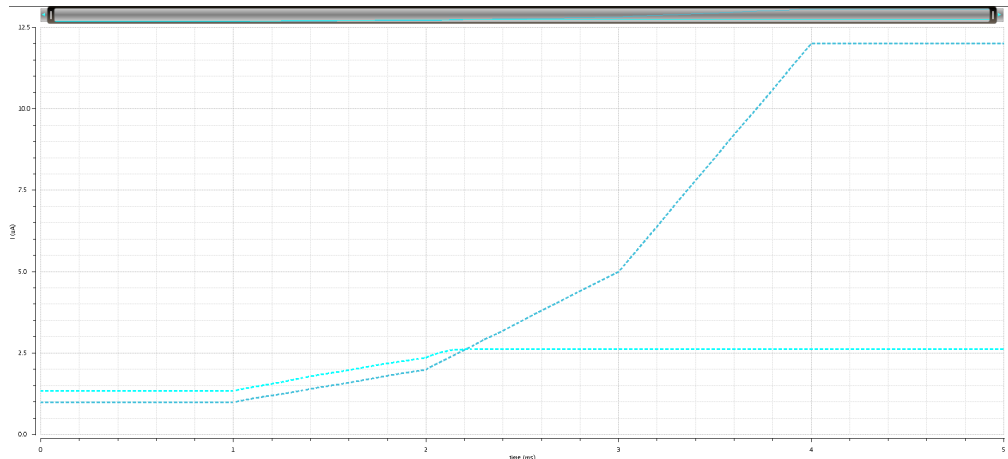
6 Parameters Chosen

We used the 'gpd180' library, which we used in our class labs for all of the transistors. The transistors were given variable parameters for sizes (i.e. Ln, Lp, Wn, Wp). We performed parametric analysis for various transistor sizes and decided to go with $Wp/Wn = 2u$ and $Ln/Lp = 180n$ for all transistors since we didn't have any ratioed circuits.

7 Type of Analysis

Most blocks required involved inputs to be given to properly test their functioning. We decided to execute analyses on the blocks which were pertinent to observe.

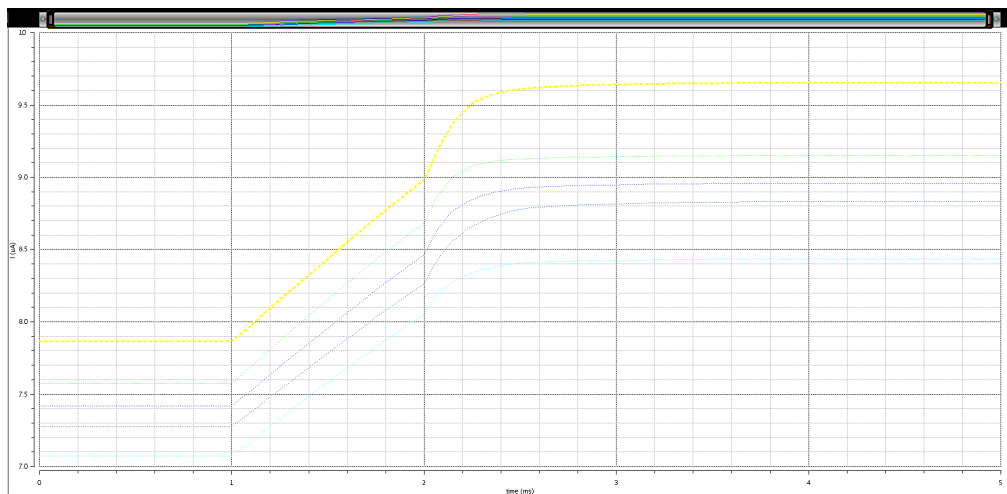
7.1 Project Block



We began with the analysis of the 'Project' Block. The block operation works on multiple current mirrors. The block simply compares the current I_{in} with the I_C current. The output current I_{out} is equal to I_{in} if $I_{in} < I_C$ and is equal to I_C for $I_{in} > I_C$.

This is shown in the graph above. We chose an input current $I_{in} = 2\mu A$. The output current I_{out} follows the input till the $2\mu A$ mark, and then saturates to that value.

7.2 Result at Process Corner



Since this is an analog circuit, we couldn't find delay for the circuit. However, we did calculate process corners for the instantaneous power. We found that the highest power is given by the FF corner, and the lowest power is given by the SS process corner.

8 References

- [1] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
- [2] D. Tank and J. Hopeld, Simple neural optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit, in IEEE Trans. Circuits and Systems, Volume: 33, Issue: 5, May 1986, pp.533 - 541
- [3] A. Bouzerdom and T.R. Pattison, Neural network for quadratic optimization with bound constraints, in IEEE Trans. Neural Networks, 42 (1993), pp. 293-304.
- [4] X.-Y. Wu, Y.-S. Xia, J. Li, and W.-K. Chen, A high performance neural network for solving linear and quadratic programming problems, in IEEE Trans. Neural Networks, 7 3 (1996), pp. 67-72.

- [5] Y.-S. Xia and J. Wang, A general methodology for designing globally convergent optimization neural networks, in IEEE Trans. Neural Networks, 9 6 (1998), pp. 1331-1343.
- [6] Y.-S. Xia and J. Wang, Global Asymptotic and Exponential Stability of a Dynamic Neural System with Asymmetric Connection Weights, in IEEE Trans. Automatic Control, VOL. 46, NO. 4, APRIL 2001, pp. 635-638.
- [7] Y.-S. Xia and J. Wang, A One-Layer Recurrent Neural Network for Support Vector Machine Learning, in IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART B: CYBERNETICS, VOL. 34, NO. 2, APRIL 2004, pp. 1261-1269.
- [8] S.-Y. Peng, P. E. Hasler, and D. V. Anderson, An analog programmable multi-dimensional radial basis function based classifier, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications.