Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής
Εργαστήριο Σχεδίασης Μικροηλεκτρονικών Κυκλωμάτων

Σχεδίαση Αναλογικών Ολοκληρωμένων Κυκλωμάτων Χαμηλής Κατανάλωσης για Υλοποίηση του Αλγορίθμου Support Vector Machine

Διπλωματική Εργασία

του

Γουρδουπάρη Μάριου

Επιβλέπων: Παύλος Π. Σωτηριάδης
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής
Εργαστήριο Σχεδίασης Μικροηλεκτρονικών Κυκλωμάτων

Σχεδίαση Αναλογικών Ολοκληρωμένων Κυκλωμάτων Χαμηλής Κατανάλωσης για Υλοποίηση του Αλγορίθμου Support Vector Machine

Διπλωματική Εργασία

του

Γουρδουπάρη Μάριου

Επιβλέπων: Παύλος Π. Σωτηριάδης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 2α Ιουλίου 2021:

Παύλος Π. Σωτηριάδης	Νεκτάριος Κοζύρης	Γεώργιος Στάμου
Καθηγητής	Καθηγητής	Καθηγητής
Ε.Μ.Π.	Ε.Μ.Π.	Ε.Μ.Π.

Αθήνα, Ιούνιος 2021

.....

Γουρδουπάρης Μάριος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών, Ε.Μ.Π.

Περίληψη

Στόχος της παρούσας διπλωματικής είναι η σχεδίαση μίας αρχιτεκτονικής αναλογικών ολοκληρωμένων κυκλωμάτων χαμηλής κατανάλωσης και τάσης τροφοδοσίας 0.6V για την υλοποίηση ενός αλγορίθμου Support Vector Machine με ικανότητα για on-chip μάθηση. Η αρχιτεκτονική του συστήματος και τα βασικά δομικά της κυκλωματικά μέρη αναλύονται, ενώ καινοτόμες κυκλωματικές αρχιτεκτονικές προτείνονται για την υλοποίηση πολυμεταβλητών ακτινικών συναρτήσεων βάσης. Η υλοποίηση αυτή παρουσιάζει πολύ χαμηλή κατανάλωση ισχύος, με όλα τα τρανζίστορ να λειτουργούν στην περιοχή υποκατωφλίου. Η προτεινόμενη αρχιτεκτονική εκτελεί τόσο την διαδικασία της μάθησης όσο και αυτή της ταξινόμησης με έναν αποκλειστικά αναλογικό και μαζικά παράλληλο τρόπο. Η αποτελεσματικότητα και η ακρίβεια του συστήματος επιβεβαιώνεται εκτελώντας τη μάθηση και την ταξινόμηση του SVM με ένα πραγματικό dataset. Οι είσοδοι του συτήματος είναι διανύσματα 13 διαστάσεων στη μορφή αναλογικών τάσεων. Η ακρίβεια της ταξινόμησης αποκλίνει από αυτήν μίας κλασσικής software υλοποίησης του SVM μόνο κατά 1%. Η προτεινόμενη αρχιτεκτονική υλοποιήθηκε σε τεχνολογία TSMC 90 nm CMOS process και προσομοιώθηκε χρησιμοποιώντας το Cadence IC Suite.

Λέξεις Κλειδιά: Support Vector Machine, analog hardware architecture, on-chip learning, on-chip classification, Ultra-low power design, Gaussian function circuit, subthreshold region, Fully tunable implementation, analog multipier circuit, Winner-Take-All circuit.

Abstract

This work presents an ultra low power, 0.6V power supply analog integrated circuit architecture for the implementation of a Support Vector Machine algorithm with on-chip learning capability. The system architecture and its basic building blocks are discussed, with novel circuit architectures being proposed for the implementation of multivariate Radial Basis Function Kernels. This is an ultra low power implementation, with all transistors operating in the subthreshold region. The proposed architecture performs both learning and classification in an exclusively analog and massively parallel way. The efficiency and accuracy of the system is validated through performing SVM learning and classification with a real dataset. The inputs of the system vectors of 13 dimensions in the form of analog voltages. The classification accuracy diverges from a classic software SVM implementation by only 1%. The presented architecture was realized in TSMC 90 nm CMOS process and simulated using the Cadence IC Suite.

Keywords:Support Vector Machine, analog hardware architecture, on-chip learning, on-chip classification, Ultra-low power design, Gaussian function circuit, subthreshold region, Fully tunable implementation, analog multipier circuit, Winner-Take-All circuit.

Ευχαριστίες

Η ολοκλήρωση της παρούσας Διπλωματικής Εργασίας σηματοδοτεί και την ολοκλήρωση των προπτυχιακών μου σπουδών. Στο σημείο αυτό θα ήθελα να ευχαριστήσω τους ανθρώπους που στάθηκαν δίπλα μου σε αυτό το κομμάτι της ζωής μου.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Παύλο Πέτρο Σωτηριάδη για την πολύτιμη καθοδηγησή του, τις συμβουλές του καθώς και το πραγματικό ενδιαφέρον που επέδειξε για μένα ως φοιτητή και για το αντικείμενο της Διπλωματικής μου Εργασίας. Μου δώθηκε έτσι η πολύτιμη ευκαιρία να ασχοληθώ με ένα ιδιαίτερα ενδιαφέρον ερευνητικό θέμα σε ένα πολύ υποστηρικτικό περιβάλλον.

Στη συνέχεια, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Βασίλειο Αλιμήση για την ανεκτίμητη συνεισφορά του. Η Διπλωματική αυτή Εργασία όπως και οι σχετικές δημοσιεύσεις σε μεγάλο μέρος τους οφείλονται σε αυτόν διότι αποτελούν το αποτέλεσμα της καθημερινής μας συνεργασίας. Οπότε όσες ευχαριστίες και να του γράψω θα είναι πραγματικά λίγες. Θέλω εξίσου να ευχαριστήσω τον υποψήφιο διδάκτορα Χρήστο Δήμα και τον διπλωματικό φοιτητή Γεώργιο Γέννη. Μαζί τους συγκροτήθηκε μια ξεχωριστή ομάδα με άριστη συνεργασία η οποία συνέβαλε καθοριστικά στην ολοκλήρωση της Εργασίας αυτής.

Τέλος, θα ήθελα πάνω από όλα να ευχαριστήσω την οικογένειά μου, τους δικούς μου ανθρώπους και τους φίλους μου για την ατελείωτη αγάπη και την στήριξή τους όλα αυτά τα χρόνια.

Γουρδουπάρης Μάριος,
Ιούνιος 2021

Περιεχόμενα

Ευρετήριο Εικόνων

Κατάλογος Πινάκων

Εκτεταμένη Ελληνική Περίληψη

Σε αυτό το κεφάλαιο θα εξεταστεί συνοπτικά η παρούσα διπλωματική εργασία μέσω μίας εκτεταμένης περίληψης στην ελληνική γλώσσα. Η θεωρητική θεμελίωση, η μαθηματική ανάλυση, τα αναλυτικά αποτελέσματα καθώς και τεχνικές λεπτομέρειες θα αναλυθούν εκτενώς στο αγγλικό κείμενο που ακολουθεί. Η παρούσα ελληνική περίληψη θα επικεντρωθεί στην ουσία των αποτελεσμάτων και την επιστημονική συνεισφορά της διπλωματικής αυτής εργασίας.

Εισαγωγή

Η κατασκευή υπολογιστικών μηχανών τα οποία θα μπορούν να αναπτύξουν την δικιά τους νοημοσύνη αποτελεί ένα διαρκές πανανθρώπινο όνειρο και έναν από τους κυρίαρχους επιστημονικούς και τεχνολογικούς στόχους του τελευταίου αιώνα. Η μηχανική μάθηση (ML) ορίζεται ως η μελέτη αλγορίθμων και στατιστικών μοντέλων που χρησιμοποιούνται για την επιτυχή εκτέλεση εργασιών χωρίς να έχουν προγραμματιστεί ρητά να το κάνουν. Η προσέγγιση της μηχανικής μάθησης επιτρέπει στα προγράμματα υπολογιστών να παράγουν νέες γνώσεις χωρίς ένα συγκεκριμένο σύνολο οδηγιών, αλλά χρησιμοποιώντας ένα σύνολο δειγμάτων δεδομένων και εξάγοντας χρήσιμα μοτίβα από αυτό. Αυτή η γενίκευση γνώσης πραγματοποιείται χωρίς ανθρώπινη παρέμβαση και οδηγεί σε αποτελεσματική πρόβλεψη ή ταξινόμηση νέων πληροφοριών από το σύστημα μηχανικής μάθησης. Η μηχανική εκμάθηση (ML) χρησιμοποιείται ως βασικό εργαλείο σε πληθώρα εφαρμογών στον σύγχρονο κόσμο, από βιοϊατρικές εφαρμογές έως αναγνώριση ομιλίας, αυτόνομη οδήγηση και χρηματιστήριο.

Οι αλγόριθμοι μηχανικής μάθησης υλοποιούνται παραδοσιακά εξ ολοκλήρου στο λογισμικό. Ωστόσο, τα δεδομένα που απαιτούνται από τις εφαρμογές μηχανικής μάθησης αυξάνονται σταθερά τα τελευταία χρόνια. Καθώς οι εργασίες μηχανικής μάθησης περιλαμβάνουν ένα τεράστιο ποσό υπολογισμών, η

εφαρμογή αλγορίθμων (ML) σε ολοένα και μεγαλύτερης κλίμακας πραγματικά προβλήματα γίνεται ιδαιτέρως απαιτητική όσον αφορά τη μνήμη και τους υπολογιστικούς πόρους. Στις παραδοσιακές αρχιτεκτονικές (CPU), οι λειτουργικοί στόχοι σχετικά με την χαμηλή κατανάλωση ενέργειας και την ταχύτητα λειτουργίας ζημιώνονται σε μεγάλο βαθμό από την ανάγκη για συνεχή μεταφορά δεδομένων μεταξύ μνήμης και επεξεργαστή.

Καθώς οι κλασικές αρχιτεκτονικές υπολογιστών αγωνίζονται να ανταποκριθούν αποτελεσματικά στις συνεχώς αυξανόμενες απαιτήσεις των σύγχρονων εφαρμογών μηχανικής μάθησης, υπάρχει μια τάση ανάπτυξης εξειδικευμένων αρχιτεκτονικών υλικού για υλοποιήσεις αλγορίθμων ΜΛ που ανταποκρίνονται σε αυτούς τους περιορισμούς. Τέτοιοι επιταχυντές υλικού(hardware accelerators) θα μπορούσαν επίσης να πληρούν τις απαιτήσεις των εφαρμογών Internet of Things (IOT), στις οποίες η συλλογή, επεξεργασία και αποθήκευση δεδομένων υψηλής ταχύτητας και χαμηλής ισχύος (μακριά από κέντρα δεδομένων) είναι υψίστης σημασίας. Η επεξεργασία πληροφοριών on-chip χωρίς την ανάγκη μεταφοράς δεδομένων ενισχύει σημαντικά την ανάπτυξη έξυπνων συστημάτων αισθητήρων χαμηλής καθυστέρησης και μεγάλης διάρκειας ζωής μπαταρίας.

Ψηφιακές αρχιτεκτονικές βασιζόμενες σε Field-Programmable Gate Arrays (FPGAs) αναπτύσσονται με μεγάλη προόδο και επιτυχία για την επιτάχυνση Βαθιών Νευρωνικών Δικτύων. Εκτός από αυτές τις ευρέως διαδεδομένες ψηφιακές αρχιτεκτονικές, ερευνούνται εναλλακτικές προσεγγίσεις για hardware ML accelerators που περιλαμβάνουν τη χρήση αναλογικών και μικτού σήματος ολοκληρωμένων κυκλωμάτων(IC). Οι αρχιτεκτονικές αναλογικών ολοκληρωμένων κυκλωμάτων έχουν το πλεονέκτημα της χαμηλής κατανάλωσης ενέργειας, μαζικά παράλληλου υπολογισμού και φθηνής υλοποίησης σε μικρό κυκλωματικό χώρο (area efficient). Η ακριβής υλοποίηση μαθηματικών συναρτήσεων και συνεπώς υπολογισμών επιτυγχάνεται χάρη στις φυσικές ιδιότητες των τρανζίστορ, με την πληροφορία να επεξεργάζεται και να μεταδίδεται σε αναλογική μορφή αντιστοιχώντας σε τάση ή ρεύμα στο κύκλωμα. Επιπλέον, οι αναλογικές αρχιτεκτονικές έχουν την δυνατότητα επεξεργασίας του σήματος απευθείας από τις εξόδους των αισθητήρων, χωρίς να απαιτούνται μετατροπείς αναλογικού σε ψηφιακό σήμα, οι οποίοι καταλαμβάνουν κυκλωματικό χώρο και καταναλώνουν μεγάλα ποσά ενέργειας. Ακολουθώντας αυτήν την προσέγγιση, αυτή η εργασία προτείνει μια εναλλακτική, χαμηλής ισχύος, πλήρως αναλογική και παράλληλη αρχιτεκτονική για την κυκλωματική υλοποίηση του αλγορίθμου Support Vector Machine με δυνατότητα για on-chip learning.

Ο αλγόριθμος SVM μπορεί να τροποποιηθεί για την περίπτωση πολλαπλών κλάσεων, αλλά στην ουσία του είναι ένας αλγόριθμος δυαδικής ταξινόμησης. Συγκεκριμένα, ο στόχος του SVM είναι να ταξινομήσει τα δεδομένα σε κατηγορίες καθορίζοντας το βέλτιστο υπερεπίπεδο, το οποίο είναι το όριο απόφασης

μεταξύ των δύο κλάσεων. Το βέλτιστο υπερεπίπεδο ορίζεται ως το υπερεπίπεδο που έχει το μέγιστο περιθώριο(απόσταση) από τα πλησιέστερα σε αυτό σημεία και των δύο κλάσεων. Στην περίπτωση μη γραμμικά διαχωρίσιμων δεδομένων, ο αλγόριθμος SVM χρησιμοποιεί το τέχνασμα του πυρήνα(Kernel trick) για τη μοντελοποίηση μη γραμμικών ορίων αποφάσεων. Το τέχνασμα του πυρήνα επιτρέπει τη λειτουργία σε χώρο πολλών διαστάσεων. Αυτό επιτυγχάνεται με τη χρήση Kernels. Οι συναρτήσεις ακτινικής βάσης (RBF) είναι από τα πιο συχνά χρησιμοποιούμενα Kernels σε υλοποιήσεις του SVM και αποτελούν επίσης μέρος αυτής της εργασίας.

Ο αλγόριθμος μάθησης του SVM μπορεί να τροποποιηθεί και να μετατραπεί σε μία πιο hardware-friendly εκδοχή του, η οποία μπορεί να υλοποιηθεί με πλήρως αναλογικά κυκλώματα(Εξ.2.10). Η μορφή αυτή και η διαδικασία εξαγωγής της περιγράφεται αναλυτικά στο εκτενές αγγλικό κείμενο.

Η πλειοψηφία των ηαρδωαρε υλοποιήσεων του SVM αποτελούν ψηφιακές αρχιτεκτονικές βασιζόμενες σε FPGA, ενώ έχουν υπάρξει προσεγγίσεις χρησιμοποιώντας και αναλογικές και μεικτού σήματος αρχιτεκτονικές. Από τις υπάρχουσες πλήρως αναλογικές υλοποιήσεις, η κάθε μία παρουσιάζει συγκεκριμένους περιορισμούς. Σε μία απο τις υλοποιήσεις τα βασικα κυκλωματικά μέρη της αρχιτεκτονικής έχουν τρανζίστορ τα οποία λειτουργούν στην περιοχή ορθής πόλωσης οπότε καταναλώνουν μεγαλύτερη ισχύ, ενώ σε άλλη απαιτούνται μετατροπείς δεδομένων από αναλογικό σε ψηφιακό σήμα και τα διανύσματα εισόδου είναι μόνο πλήθους 2 διαστάσεων. Σε μία τρίτη υλοποίηση επιτυγχάνεται αναλογικά μόνο ταξινόμηση στο chip αλλά στη διαδικασία της μάθησης συμμετέχει υπολογιστής ενώ τέλος σε μία άλλη ο SVM υλοποιείται πλήρως αναλογικά αλλά μόνο σε επίπεδο θεωρητικής προσομοίωσης στο Matlab, χωρις αναλογική VLSI σχεδίαση στο Cadence.

Αρχιτεκτονική Συστήματος

Σε αυτήν την εργασία παρουσιάζεται μια πλήρως παράλληλη και πλήρως αναλογική ηαρδωαρε υλοποιήση χαμηλής κατανάλωσης του αλγορίθμου SVM με δυνατότητα εκμάθησης on-chip. Η πρωτοτυπία αυτού του έργου έγκειται στο γεγονός ότι αποτελείται από καινοτόμα κυκλώματα εξαιρετικά χαμηλής ισχύος ως δομικά στοιχεία για την υλοποίηση πολυδιάστατων RBF συναρτήσεων και όλα τα τρανζίστορ της αρχιτεκτονικής λειτουργούν στην περιοχή υποκατωφλίου(subthreshold region). Επιπλέον, μια τροποποιημένη έκδοση του κλασικού κυκλώματος Winner Take All (WTA) χρησιμοποιείται για βελτιωμένη ακρίβεια της διαδικασίας ταξινόμησης. Η προτεινόμενη αρχιτεκτονική δοκιμάστηκε για μάθηση και ταξινόμηση χρησιμοποιώντας 8 learning samples και

δεδομένα εισόδου προς ταξινόμηση 13 διαστάσεων.

Η προτεινόμενη αρχιτεκτονική του συστήματος αποτελείται από δύο βασικά δομικά στοιχεία, τη μονάδα εκμάθησης και τη μονάδα ταξινόμησης. Σε επίπεδο συστήματος, το μπλοκ εκμάθησης έχει σχεδιαστεί με στόχο την υλοποίηση του hardware-friendly κανόνα εκμάθησης του SVM και απεικονίζεται στο Σχ.2.2. Επομένως, υπάρχει ανάγκη για κυκλώματα που υλοποιούν RBF Kernels, κυκλώματα που ενσωματώνουν τα labels, κυκλώματα που να πολλαπλασιάζουν με την τιμή του ai και κυκλώματα που θα πραγματοποιούν τις απαραίτητες επαναλήψεις. Οι τιμές των πολλαπλασιαστών Lagrange και των RBF Kernels αντιστοιχούν σε ρεύματα τρανζίστορ, ενώ τα labels yi = + -1 αντιστοιχούν στις θετικές και αρνητικές τάσεις τροφοδοσίας αντίστοιχα.

Το μπλοκ εκμάθησης αποτελείται από ένα array M^2 RBF, όπου το M είναι ο αριθμός των δειγμάτων που εμπλέκονται στη διαδικασία εκμάθησης. Τα κελιά RBF λαμβάνουν τις εισόδους (δείγματα εκμάθησης) του συστήματος. Κάθε κελί RBF εφαρμόζει μία πολυμεταβλητή συνάρτηση RBF N διαστάσεων με ρυθμιζόμενο ύψος, ώστε να ικανοποιεί την ανάγκη πολλαπλασιασμού με ai. Τα labels περιλαμβάνονται μέσω της χρήσης $M(M-1)$ διακοπτών . Η έξοδος κάθε $X_{i,j}$ κελιού RBF για $i\ neq j$ από τον πίνακα $X_{M\ cdot M}$ των κυττάρων RBF έχει το αντίστοιχο label που επισυνάπτεται από τους διακόπτες. Τα ρεύματα εξόδου των διακοπτών σύμφωνα με τα αντίστοιχα labels αθροίζονται και συνδέονται με τα adjuster κυκλώματα.

Υπάρχουν M adjuster κυκλώματα που ουσιαστικά εφαρμόζουν τις μη γραμμικές λειτουργίες ελαχιστοποίησης και μεγιστοποίησης του hardware-friendly κανόνα εκμάθησης του SVM. Κάθε adjuster κύκλωμα λαμβάνει έπειτα από άθροιση τα ρεύματα εξόδου των διακοπτών μιας σειράς του πίνακα $X_{M\ cdot M}$ των κυττάρων RBF και παράγει ρεύματα εξόδου τα οποία στη συνέχεια τροφοδοτούνται σε μία στήλη του πίνακα των RBF κυττάρων. Έτσι, διαμορφώνεται ένας βρόχος ανάδρασης και το κύκλωμα εκμάθησης συγκλίνει πλήρως παράλληλα και ασύγχρονα, χωρίς τη χρήση εξωτερικού ρολογιού. Η διαδικασία μάθησης ολοκληρώνεται προσδιορίζοντας τις σωστές τιμές για τα ρεύματα εξόδου των adjusters που αντιπροσωπεύουν τις παραμέτρους μάθησης του αλγορίθμου SVM.

Το μπλοκ ταξινόμησης έχει σχεδιαστεί με σκοπό την εφαρμογή του κανόνα απόφασης του SVM σε hardware και απεικονίζεται στο Σχ.2.3. Τα testing δεδομένα, (διανύσματα N διαστάσεων) τροφοδοτούνται στο μπλοκ ταξινόμησης σύμφωνα με ένα εξωτερικό ρολόι. Κατά τη διάρκεια κάθε κύκλου ρολογιού, καθένα από τα M κελιά RBF υπολογίζει την τιμή συνάρτησης RBF μεταξύ του διανύσματος testing του συγκεκριμένου κύκλου και του δείγματος εκμάθησης που αντιστοιχεί στο συγκεκριμένο κελί. Τα κελιά RBF πολώνονται με αντίγραφα των ρευμάτων εξόδου των adjusters που καθορίζονται κατά τη διάρκεια της

φάσης εκμάθησης του συστήματος.

Στη συνέχεια, πρέπει να καθοριστεί το πρόσημο του αθροίσματος στην έκφραση του κανόνα απόφασης του SVM. Αυτό θα μπορούσε να επιτευχθεί συγκρίνοντας τις θετικές τιμές αυτού του αθροίσματος με τις αρνητικές. Στην αρχιτεκτονική της υλοποίησής μας, αυτές οι θετικές τιμές είναι τα ρεύματα εξόδου των κυττάρων RBF που έχουν δείγματα εκμάθησης εισόδου με θετικό label, ενώ οι αρνητικές τιμές είναι τα ρεύματα εξόδου των κυττάρων RBF που έχουν δείγματα εκμάθησης εισόδου με αρνητικό label. Χρησιμοποιώντας διακόπτες, όλες οι θετικές τιμές αθροίζονται μαζί, όπως όλες οι αρνητικές. Στη συνέχεια, ένα κύκλωμα που ονομάζεται Winner-Take-All (WTA) χρησιμοποιείται για τη σύγκριση των αρνητικών και των θετικών τιμών με αποτέλεσμα την ταξινόμηση του άγνωστου διανύσματος εισόδου. Χρησιμοποιείται ένα κύκλωμα WTA αντί ενός συγκριτή λόγω του γεγονότος ότι η επεξεργασία της πληροφορίας στο σύστημα πραγματοποιείται κυρίως σε μέσω ρεύματος, όπως συμβαίνει και στα WTA κυκλώματα.

Η πλήρης ανάλυση των αναλογικών κυκλωμάτων για τις πολυδιάστατες RBF συναρτήσεις, των διακοπτών που ενσωματώνουν τα labels, των adjusters και των WTA κυκλωμάτων πραγματοποιείται στο εκτενές αγγλικό κείμενο παρακάτω.

Αρχιτεκτονικές Κυκλωμάτων

Τα βασικά δομικά στοιχεία της αρχιτεκτονικής του προτεινόμενου συστήματος είναι αναλογικά κυκλώματα που αποτελούνται από τρανζίστορ MOS που λειτουργούν στην περιοχή υποκατωφλίου. Αυτή η επιλογή σχεδιασμού βασίζεται σε δύο κύρια χαρακτηριστικά της λειτουργίας της περιοχής subthreshold που την καθιστά μια ελκυστική τεχνική για το σχεδιασμό αναλογικών αρχιτεκτονικών για επεξεργασία πληροφορίας. Πρώτον, στην περιοχή κάτω του κατωφλίου το ρεύμα του τρανζίστορ σχετίζεται εκθετικά με την τάση μεταξύ πύλης και πηγής. Αυτό επιτρέπει την υλοποίηση χρήσιμων μαθηματικών εκφράσεων σε κυκλώματα όπου το ρεύμα εκφράζει την επιθυμητή πληροφορία. Επιπλέον, η ροή ρεύματος στα τρανζίστορ που λειτουργούν στη subthreshold είναι πολύ χαμηλότερη από τη ροή ρεύματος σε περιοχή ισχυρής αναστροφής. Αυτό επιτρέπει τη σωστή λειτουργία των σχεδιασμένων κυκλωμάτων με χαμηλό ρεύμα λίγων nA ή ακόμη και pA, με αποτέλεσμα εξαιρετικά χαμηλή κατανάλωση ισχύος. Ολόκληρο το προτεινόμενο σύστημα λειτουργεί με πολύ χαμηλή τάση τροφοδοσίας 0.6V.

Παρουσιάζονται και αναλύονται λεπτομερώς 4 νέες κυκλωματικές αρχιτεκτονικές για την υλοποίηση πολυδιάστατων RBF συναρτήσεων. Τα προτει-

νόμενα κυκλώματα έχουν ανεξάρτητο και ηλεκτρονικό έλεγχο στο ύψος, τη μέση τιμή και την διασπορά του ρεύματος εξόδου το οποίο είναι μία πολυδιάστατη Γκαουσιανή συνάρτηση των τάσεων εισόδου των κυκλωμάτων αυτών. Και οι 4 προτεινόμενες τοπολογίες χρησιμοποιούν έλεγχο του bulk των τρανζίστορ ώστε να επιτύχουν την επιθυμητή ελεγξιμότητα στο εύρος της Γκαουσιανής καμπύλης εξόδου. Τα κυκλώματα αυτά έχουν πιο χαμηλή τάση τροφοδοσίας σε σχέση με την βιβλιογραφία, όσο και αρκετά πιο χαμηλή κατανάλωση. Χαρακτηριστικά αναφέρεται ότι η 1η από τις προτεινόμενες τοπολογίες(Σχ.3.3) επιτυγχάνει σωστή λειτουργία με κατανάλωση ενέργειας 3,9nW για την υλοποίηση μιας μονοδιάστατης Γκαουσσιανής, με την μείωση στην κατανάλωση ισχύος σε σχέση με την χαμηλότερη τιμή της βιβλιογραφίας (13,5nW να είναι είναι κατά 71,1% . Παράλληλα, οι κυκλωματικές αυτές αρχιτεκτονικές απαιτούν μικρό αριθμό τρανζίστορ για την υλοποίηση Γκαουσιανών μίας διάστασης και έχουν την ικανότητα επέκτασης της αρχιτεκτονικής για την υλοποίηση πολυδιάστατων RBF συναρτήσεων.

Με σκοπό την βελτίωση της γραμμικότητας στον έλεγχο του ύψους των πολυδιάστατων RBF συναρτήσεων, στην έξοδο κάθε cascaded Gaussian Function circuit συνδέεται ένας αναλογικός translinear πολλαπλασιαστής(Σχ.3.62) ο οποίος επιτυγχάνει την γραμμικότητα αυτή. Στη συνέχεια, διακοπτικά κυκλώματα τα οποία υλοποιούν XOR πύλες με λιγότερα τρανζίστορ από ότι οι παραδοσιακή CMOS static λογική, ενσωματώνουν στην έξοδο των RBF cells την πληροφορία που εμπεριέχεται στα labels(Σχ.3.64). Στη συνέχεια σχεδιάζονται οι adjusters(Σχ.3.65) οι οποίοι είναι κυκλώματα βασισμένα σε καθρεύτες ρεύματος και υλοποιούν τις μη γραμμικές λειτουργείες του μεγίστου και του ελαχίστου που εμπεριέχονται στην έκφραση εκμάθησης του Σ΄Μ. Οι adjusters λαμβάνουν ως εισόδους τα ρεύματα που έρχονται από τους διακόπτες ενώ τα ρεύματα εξόδου των adjusters πολώνουν τα RBF cells. Έτσι δημιουργείται ο βρόγχος ανάδρασης που επιτρέπει στην κυκλωματική αρχιτεκτονική να συγκλίνει στις σωστές τιμές των ρευμάτων εξόδου των adjusters που αναπαριστούν τους πολλαπλασιαστές Lagrange της διαδικασίας μάθησης του αλγορίθμου.

Για την ταξινόμηση των άγνωστων δεδομένων εισόδου και την υλοποίηση του κανόνα απόφασης του SVM χρησιμοποιούνται κυκλώματα Winner-Take-All (WTA) (Σχ.3.69) τα οποία δέχονται ως εισόδους ρεύματα και στην έξοδο τους μη μηδενικό είναι το μόνο ρεύμα εκείνο που αντιστοιχεί στο ρεύμα εισόδου με την μεγαλύτερη τιμή. Ουσιαστικά, τα κυκλώματα αυτά εκτελούν σύγκριση μεταξύ των ρευμάτων εισόδου τους και επιτυγχάνεται έτσι η ταξινόμηση. Στην προτεινόμενη αρχιτεκτονική αντί για ένα απλό WTA κύκλωμα, χρησιμοποιείται ένα τριπλό στη σειρά, με αποτέλεσμα να αυξάνεται η ακρίβεια στην σύγκριση των ρευμάτων και συνεπώς στην ταξινόμηση.

Εφαρμογή του Συστήματος και Αποτελέσματα

Η ορθή λειτουργία και αποτελεσματικότητα της προτεινόμενης αρχιτεκτονικής του συστήματος για μάθηση και ταξινόμηση SVM δοκιμάζεται και επαληθεύεται χρησιμοποιώντας ένα πραγματικό σύνολο δεδομένων. Προκειμένου να χρησιμοποιηθούν αυτά τα δεδομένα για μάθηση και ταξινόμηση SVM, επιλέγονται 2 από τις συνολικές κλάσεις του συνόλου δεδομένων για κάθε δοκιμή. Μετά την επεξεργασία των δεδομένων από Python, εξάγονται 13 χαρακτηριστικά. Για την περίπτωση ταξινόμησης μεταξύ της πρώτης και της δεύτερης κλάσης του συνόλου δεδομένων, το hardware προτεινόμενο σύστημα πέτυχε validation accuracy 83 %, ενώ το validation accuracy της Python ήταν 84 %. Για την περίπτωση ταξινόμησης μεταξύ της πρώτης και της τρίτης κλάσης του συνόλου δεδομένων, η αρχιτεκτονική της υλοποίησης μας πέτυχε validation accuracy 94 %, ενώ το validation accuracy της Python ήταν 93 %. Τα σφάλματα μεταξύ λογισμικού και της προτεινόμενης hardware υλοποίησης είναι σχεδόν αμελητέα, γεγονός που αποδεικνύει τη σωστή λειτουργία και την αποτελεσματικότητα της παρουσιαζόμενης κυκλωματικής αρχιτεκτονικής.

Επίλογος και Μελλοντική Επέκταση

Σε αυτήν την εργασία, παρουσιάστηκε μια πλήρως αναλογική, εξαιρετικά χαμηλής ισχύος και μαζικά παράλληλη αναλογική κυκλωματική αρχιτεκτονική για την υλοποίηση του SVM με δυνατότητα μάθησης on-chip. Η αρχιτεκτονική του συστήματος αναλύθηκε και τα βασικά δομικά στοιχεία της συζητήθηκαν λεπτομερώς σε επίπεδο τρανζίστορ. Παρουσιάστηκαν τέσσερις διαφορετικές αρχιτεκτονικές κυκλωμάτων για την αναλογική υλοποίηση των πυρήνων ΡΒΦ, καθώς και πολλαπλασιαστές, διακόπτες, adjusterts και κυκλώματα WTA. Η προτεινόμενη αρχιτεκτονική εκτελεί μάθηση και ταξινόμηση με εντελώς αναλογικό τρόπο και περιλαμβάνει δομικά κυκλωματικά στοιχεία πολύ χαμηλής κατανάλωσης ισχύος. Αποδίδει αποτελεσματικά σε προβλήματα δυαδικής ταξινόμησης, καθώς δοκιμάστηκε με ένα πραγματικό σύνολο δεδομένων και παρουσίασε σφάλματα όχι περισσότερο από 1 % σε σύγκριση με μια παραδοσιακή υλοποίηση του SVM με λογισμικό. Μελλοντικές εργασίες που σχετίζονται με αυτήν την αρχιτεκτονική θα μπορούσαν να περιλαμβάνουν layout και κατασκευή chip και μετρήσεις του προτεινόμενου κυκλώματος. Επιπλέον, θα μπορούσαν να σχεδιαστούν αναλογικά και χαμηλής ισχύος κυκλώματα μνήμης συνδεόμενα με την προτεινόμενη αρχιτεκτονική, με σκοπό την αποθήκευση τιμών παραμέτρων. Αυτή η αρχιτεκτονική θα μπορούσε επίσης να κλιμακωθεί προκειμένου να εξυπηρετεί διανύσματα εισόδου υψηλότερης διάστασης από 13 και με περισσότερα

διαθέσιμα δείγματα μάθησης από 8, όπως συνέβαινε σε αυτό το έργο. Εκτός από την εφαρμογή SVM, τα βασικά δομικά στοιχεία αυτής της αρχιτεκτονικής θα μπορούσαν να βελτιστοποιηθούν περαιτέρω και να χρησιμοποιηθούν σε άλλες εφαρμογές hardware acceleration αλγορίθμων μηχανικής μάθησης.

Κεφάλαιο 1

Introduction

1.1 Machine Learning in general

The design of machines that could develop their own intelligence has been one of the biggest dreams of humankind and one of the greatest scientific and technological challenges of the last century. Machine learning is defined as the study of algorithms and statistical models that are used to successfully perform tasks without having been explicitly programmed to do so[1, 2]. The machine learning approach enables computer programs to generate new knowledge without a specific instruction set but by using a set of sample data and extracting useful patterns from it. This knowledge generalization is performed without human intervention and results in effective prediction or classification of new information by the machine learning system. Machine learning(ML) is used as a fundamental tool in an abundance of applications in the modern world, ranging from biomedical applications[3] to speech recognition[4], autonomous driving[5] and stock market trading[6].

1.2 Benefits of hardware Machine Learning architectures

Machine learning algorithms are traditionally implemented entirely in software. However, the amount of data required by machine learning applications has been steadily increasing in recent years. As machine learning tasks involve a massive amount of computations, ML algorithms become increasingly demanding in memory and computational resources [7, 8]. In traditional CPU architectures, goals regarding power consumption and speed of operation are greatly compromised by the need for constant data transfer be-

tween memory and processor. As classic computer architectures struggle to keep up efficiently with the ever increasing demands of modern machine learning applications there is a trend towards developing specialised hardware architectures for implementations of ML algorithms that meet those constraints[9].

Such hardware accelerators could also meet the requirements of Internet of Things (IOT) applications, in which high speed and low power embedded data collection, processing and storage (away from data centers) is of paramount importance. On-chip information processing without the need to transfer data greatly boosts the development of low latency and long battery life smart sensor systems[10, 11] .

1.3 Hardware Machine Learning architectures

Digital architectures based on Field-Programmable Gate Arrays (FPGAs) are becoming increasingly popular when it comes to hardware acceleration of Deep Learning networks[12].

Apart from digital architectures, alternative approaches are being researched involving the use of analog and mixed-signal integrated circuits (IC). Analog IC architectures have the advantage of low power consumption, low area and massively parallel computation.Accurate emulation of mathematical functions is achieved thanks to the physical properties of transistors[13, 14, 15].

Extensive research has been conducted in the past regarding analog hardware implementation of Artificial Neural Networks[16, 17, 18, 19]. Traditional analog VLSI building blocks have been used to implement neurons and synapses while learning algorithms like back propagation[20, 21, 22] and weight perturbation [23, 24, 25, 26] have been realized in hardware. However, such architectures generally could not achieve similar efficiency as software approaches.

An emerging approach which demonstrates promising results in hardware Neural Networks Implementations are neuromorphic systems[27, 28]. They are composed of analog spike based circuits that implement neurons and synapses imitating the function of the human brain along with digital Address Event Representation (AER) communication protocols[29, 30, 31, 32]. Neuromorphic architectures implement Spike-Timing-Dependent-Plasticity Rules (STDP) and realize Spiking Neural Networks(SNNs) in hardware[33]. Neuromorphic systems often incorporate arrays of memristors, non-linear and non-volatile analog programmable memories that can

store the network's weights[34, 35, 36, 37]. Large scale neuromorphic chips have already been fabricated and tested[38, 39, 40].

Apart from architectures implementing general purpose neural networks' structures, dedicated hardware architectures have also been proposed for executing specific tasks. Mixed-mode processors have been implemented for object recognition[41, 42] as well as analog architectures for K-means clustering[43, 44], Radial Basis Function(RBF) neural networks[45], RBF classifiers[46], Support Vector Machine algorithm[47, 48, 49, 50, 51], Support Vector Domain Description algorithm[52] and Support Vector Regression for approximate computing[53].

Following this approach, this work proposes an alternative low power, fully analog and parallel architecture for hardware implementation of Support Vector Machine algorithm with on-chip learning capability.

Κεφάλαιο 2

SVM hardware system architecture

2.1 Support Vector Machine Theory

Support Vector Machines are algorithms that belong in the category of Supervised Machine Learning, one of the core branches of Machine Learning and Artificial Intelligence theory. Supervised machine learning systems are used either for regression or classification problems. Supervised ML uses a training dataset, which is data with known labels, to perform the learning procedure by minimizing a loss function with algorithms like gradient descent. In the case of classification, after the learning procedure has been completed and the system's parameters have been updated accordingly, new data with unknown labels serves as the input of the system that attempts to categorize it.[1, 2]

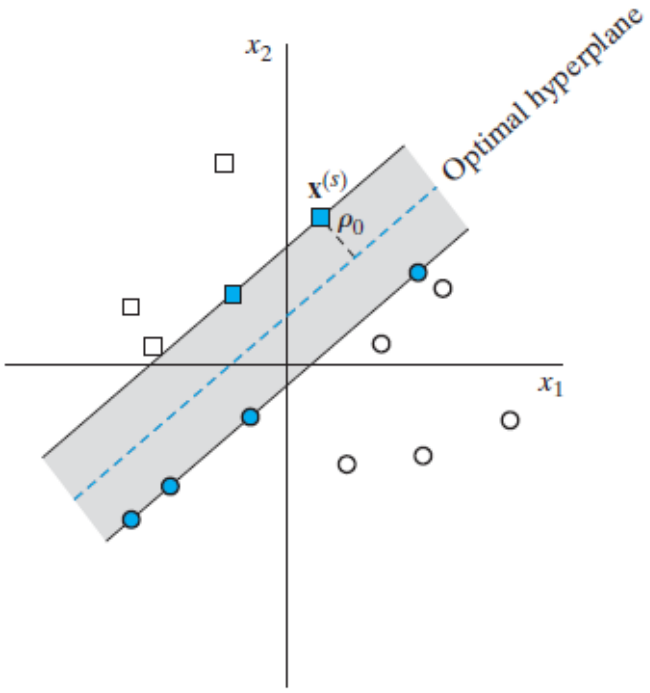
SVM algorithm can be modified to accommodate multiple classes but in its essence it is a binary classification algorithm. In particular, the goal of SVM is to categorize data into classes by determining the optimal hyperplane, which is the decision boundary between the two classes. The optimal hyperplane is defined as the hyperplane that has the maximum margin from the closest points of both classes(Fig.2.1). If w_o, b_o are the optimal values of the weight vector and the bias vector respectively, Xi is the N-dimensional training sample with label yi equal to 1 or -1 , the optimal hyperplane as a mutlidimensional linear decision boundary is defined as:

$$w_o^T X + b_o = 0 \tag{2.1}$$

The goal is to determine the values w_o and b_o for the optimal hyperplane,

given a specific learning set Xi,yi. This pair of values should satisfy the following expression:

$$yi(w_o^T Xi + b_o) \geq 1 \text{for i-1,2,...N} \tag{2.2}$$



Σχήμα 2.1: SVM's optimal hyperplane

Apart from satisfying the above expression, the optimal values of w and b should be such that maximize the margin of separation between the two classes. Maximizing this margin is equivalent to minimizing the norm $\|X\|$ Thus, the following cost function should be minimized:

$$F(w) = \frac{w^T w}{2} \tag{2.3}$$

The minimization of this cost function with the specified linear constraints is a convex optimization problem and is solved by a transformation to its dual problem and the use of Lagrange multipliers.

In the case of non-linearly separable data, the SVM algorithm makes use of the kernel trick to model non-linear decision boundaries. The kernel

trick allows operation in high dimensional feature space. This is achieved by using kernel functions. Radial Basis Functions(RBF) are among the most commonly used Kernels in SVM implementations and they are also part of this work. The RBF Kernel on two vector samples X and X' and parameter σ is defined as:

$$K(X, X') = e^{-\frac{\|X-X'\|^2}{2s^2}} \tag{2.4}$$

For the case of non-linearly separable dataset the dual optimization problem is defined as: Given a training dataset (xi,yi) for i=1,2...N the Lagrange multiplier values ai for i=1,2...N should be determined so as to maximize the following function:

$$Q(a) = \sum_{i=1}^{N} a_i - \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j K(x_i, x_j)}{2} \tag{2.5}$$

under the following constraints:

$$\sum_{i=1}^{N} a_i y_i = 0 \tag{2.6}$$

and

$$0 \leq a_i \leq C \tag{2.7}$$

for i=1,2....N, where C is a regularization parameter

This is SVM's learning rule. With the values of the Lagrange multipliers that correspond to the optimal hyperplane having been determined by the learning rule, the classification rule of SVM is the following:

$$f(x) = sign[\sum_{i=1}^{N} a_i y_i K(x, x_i) + b] \tag{2.8}$$

for input test vector x and a training set xi,yi for i=1,2...N

2.1.1 Hardware-Friendly SVM Algorithm

The values of the Lagrange Multipliers are derived by solving the constrained quadratic programming problem defined by equations (2.5), (2.6) and (2.7). This SVM learning rule can be modified to be more compatible with analog hardware architectures[48, 49].To achieve this, we choose a type of gradient-descent algorithm to update the values of ai in the following way:

$$a_i \leftarrow a_i - \frac{\partial W(a,b)}{\partial a_i} n_i \tag{2.9}$$

where ni is the learning rate.For choosing $n_i = \frac{1}{K(x_i,x_i)}$ and for K being a self normalized kernel like the Gaussian Kernel ($K(x_i, x_i) = 1$) the hardware-friendly SVM update rule is defined as follows:

$$a_i \leftarrow min(C, max(0, 1 - y_i \sum\nolimits_{i \neq m} y_m a_m K(x_i, xm))) \tag{2.10}$$

In this update rule the bias value b is set to 0. The characteristics of the Gaussian Kernel, which maps the input vectors to a space of infinite dimensions, makes the omission of a single bias value b possible as its effect on the total result can be considered negligible.

The derived SVM update rule of the last equation is more suitable for hardware implementation, thanks to specific properties it demonstrates. Firstly, there is no need for extra memory to store previous ai values as they do not appear in the right hand side of the update rule. Furthermore, the form of the update rule resembles the one of the classification rule Eq.(2.8), meaning that the same hardware blocks could be used for both tasks. This would simplify the system architecture and make it more compact and area efficient.

2.2 Related Work in Hardware SVM implementation

SVM hardware implementations usually involve digital FPGA based architectures [54, 55, 56, 57, 58]. There have also been several mixed signal[50] and analog[51, 47, 48, 49] architectures for hardware implementation of SVM.

The mixed-signal architecture in [50] incorporates analog circuits for computations along with a digital interface including delta-sigma Analog to Digital converters and DRAM storage,performing both learning and classification on chip.

In [51] an array of analog translinear circuits with floating gate transistors operating in the subthreshold region is used. Low power computation provided by translinear and subthreshold techniques is combined with analog non volatile memory storage due to the existence of floating gate transistors. The specific implementation achieves very low power consumption in a very large scale experimental setup, performing multiclass SVM for 24 classes, with input vectors of 14 dimensions and as many as 720 support vectors.

However, the learning is not performed on-chip and in an entirely analog way. The learning procedure is called PC-in-loop, as a computer is connected to the system that performs the update of learning parameters of SVM in software. Then, these parameters are downloaded to the analog floating gate array.

On the contrary, the circuit architectures in [47, 48, 49] perform on chip learning of SVM along with classification.

In [47] a fully analog implementation of SVM is presented, using floating gate transistors operating in the subthreshold region. To implement the learning procedure, projection neural networks adapted for SVM are proposed and the constrained quadratic problem is solved by a set of ordinary differential equations. This fully analog approach however has only been realized with Matlab and Spice simulations, without actual analog vlsi design taking place. Analog circuit design and tapeout of such an architecture would be complicated due to the presence of floating gate transistors.

In[48] a row parallel architecture is presented, with transistors operating in the subthreshold region. It utilizes the hardware-friendly version of SVM algorithm of ().The proposed implementation is area efficient and achieves low power consumption. However, the proof of concept chip fabricated in the specific work classifies input vectors of only 2 dimensions. Furthermore, Analog to Digital converters and a digital block in a feedback loop configuration realizing a binary search algorithm are necessary to implement the training mode of SVM.

In [49] a fully analog and parallel architecture is presented. The basic circuit components of this architecture are such that enable an area efficient implementation of analog Kernels as well as a more robust design compared to other works, suitable for the implementation of high dimensional Kernels (accommodating inputs of 64 dimensions). This architecture also makes use of the hardware-friendly version of SVM algorithm but in contrast to [48] realizes it with fully analog circuitry. The analog circuits are self-converging, determining the proper Lagrange multiplier values for SVM learning without the presence of an external digital clock. For the realization of multivariate RBF Kernels, this architecture uses circuits with transistors operating in saturation region. On the one hand this design choice increases the speed of operation and the robustness of the architecture against process variations. On the other hand it leads to higher power consumption compared to implementations using transistors operating in the subthreshold region.

2.3 Proposed System Level Architecture of Hardware SVM implementation

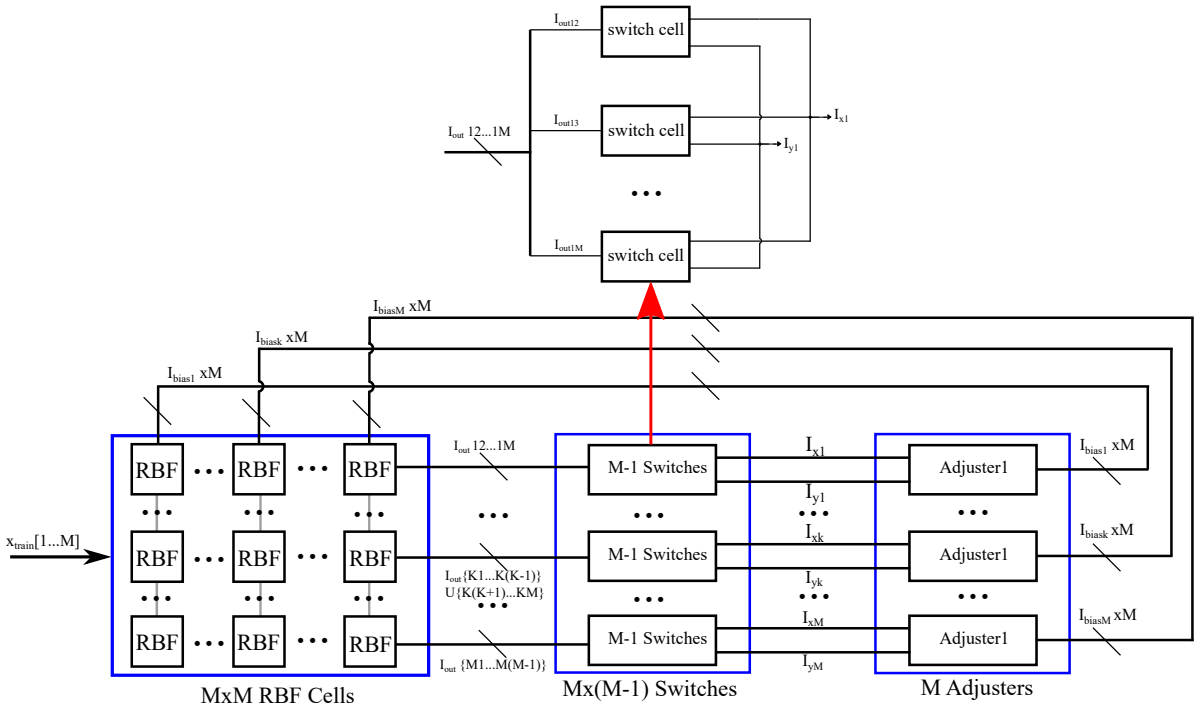
In this work a fully parallel and fully analog ultra low power hardware implementation of SVM algorithm with on-chip learning capability is presented. The system architecture and the learning circuitry are inspired by [49] as the same hardware-friendly version of SVM is realized exclusively with analog circuits.

The proposed system architecture consists of two main building blocks, the learning and the classification block. In a system level perspective, the learning block is designed with the goal of realizing the update rule of the hardware-friendly SVM and is depicted in Fig.2.2 Thus, there is a need for circuits implementing Kernels, incorporating labels, multiplying by the value of ai and performing the necessary iterations. The lagrange multipliers' and kernel function's values are realized with transistor currents while the labels yi= +-1 correspond to the positive and negative supply voltages respectively.

The learning block consists of an array of M^2 RBF cells, where M is the number of samples involved in the learning procedure. The RBF cells receive the inputs(learning samples) of the system. Each RBF cell implements a multivariate RBF Kernel of N dimensions with tunable height so as to satisfy the need for multiplication by ai.Labels are included through the use of $M(M-1)$ switches. The output of every $X_{i,j}$ RBF cell for $i \neq j$ from the matrix $X_{M \cdot M}$ of RBF cells has its corresponding label attached to it by the switches. The output currents of the switches according to the corresponding labels are summed and connected to the adjuster circuits.

There are M adjuster circuits which essentially implement the non linear minimum and maximum operations of the hardware-friendly update rule. Each adjuster circuit receives the summed output currents of the switches of one row of the matrix $X_{M \cdot M}$ of RBF cells and produces output currents which are then fed back to one column of the maxtrix of RBF cells. Thus, a feedback loop configuration is formed and the learning circuitry self-converges without the use of an external clock. The learning process is completed in a fully parallel and autonomous fashion, determining the right values for the adjusters' output currents which represent the learning parameters of SVM algorithm.

The classifiaction block is designed with the aim of implementing the SVM decision rule(Eq.2.8) in hardware and is depicted in Fig.2.3 It consists of M RBF cells, M switches and a Winner Take All circuit for comparison.

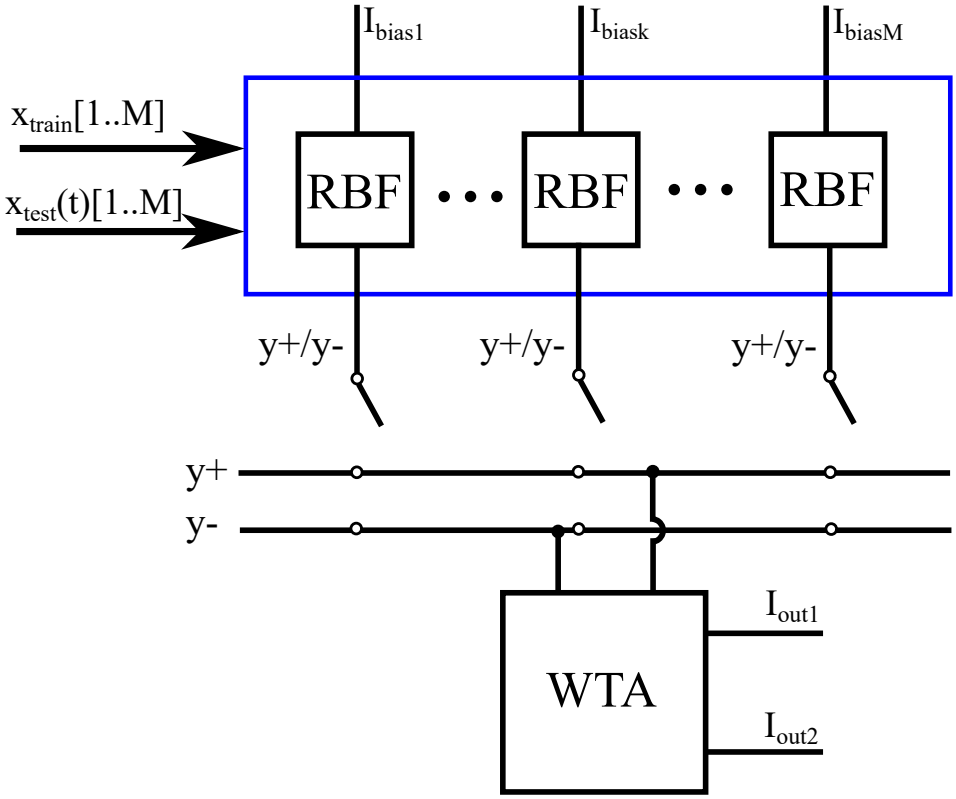


Σχήμα 2.2: Learning Block.

Test samples(vectors of N dimensions) are synchronously (based on an external clock) fed to the classification block. During every clock cycle, each of the M RBF cells computes the RBF Kernel Function of the test vector of the specific cycle and the learning sample corresponding to the specific cell. The RBF cells are biased with copies of the adjusters' output currents that are determined during the learning phase of the system.

Next, the sign of the sum in the expression of SVM's decision rule has be to determined. This could be achieved by comparing the positive values of this sum to the negative ones. In our hardware implementation, these positive values are the output currents of the RBF cells that have input learning samples with positive label while the negative values are the output currents of the RBF cells that have input learning samples with negative label. By using switches, all the positive values are summed together, same as all the negative ones. Then, a current-mode circuit called Winner-Take-All(WTA) circuit is used for comparing the negative to the positive values and classifying the test vector. A WTA circuit is used instead of a comparator due to the fact that information processing in the system is performed mainly in

current-mode.



Σχήμα 2.3: Classification Block.

The novelty of this work lies in the fact that it consists of novel ultra low power circuits as building blocks for implementing multivariate Kernels and all transistors of the architecture operate in the subthreshold region. Furthermore, a modified version of the classic Winner Take All (WTA) circuit is used for enhanced accuracy of the classification procedure. The proposed architecture was tested for learning and classification using 8 learning samples and classification instances of 13 dimensions.

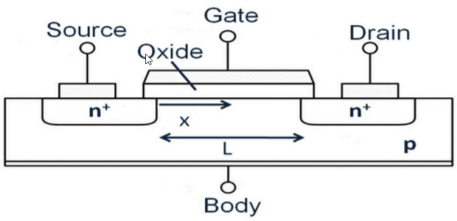
The transistor level analog implementation of the RBF cells, the switch cells, the adjuster circuits and the Winner Take All circuit will be discussed in detail below.

Κεφάλαιο 3

Proposed Circuit architectures

3.1 MOS Transistors in Subthreshold Region Operation

In Fig.3.1 the structure of an NMOS transistor is depicted. The body or substrate of a NMOS is p-type so in the majority carriers are holes. In the n-type regions of the source and the drain, which are symmetrical in a mosfet structure, the majority carriers are electrons. The channel is the area between the source and the drain, underneath the gate and has length L and width W. The gate terminal of the transistor is made of polysilicon and is insulated from the channel by using silicon dioxide. The voltages at the gate (V_G), the source (V_S) and the drain (V_D) of the transistor are referenced to the substrate voltage. V_t is the threshold voltage, a value above which a conducting channel is formed and the transistor operates in linear and strong inversion region.For $V_{GS} < V_t$ the transistor operates in the so-called subthreshold or weak inversion region.



Σχήμα 3.1: NMOS structure

It is often assumed that for V_{GS} values lower than the threshold voltage the transistor's current between drain and source is zero. However, in the subthreshold region the mosfet's current is an exponential function of its gate to source voltage. In this region of operation, the electric charge of the inversion region is much smaller than the charge of the depletion region, due to the fact that the gate to source voltage applied is not sufficiently large to form a conducting channel. The subthreshold current is essentially diffusion current caused by th concentration of minority carriers. By using the sub-threshold region MOS model in [59], the currents for for the PMOS and NMOS respectively are the following:

$$I_{pmos} = I_{o_p} e^{\kappa_p (V_w - V_G)/V_T} \left(e^{(V_S - V_w)/V_T} - e^{(V_D - V_w)/V_T} \right) \tag{3.1}$$

$$I_{nmos} = I_{o_n} e^{\kappa_n (V_G - V_w)/V_T} \left(e^{(V_w - V_S)/V_T} - e^{(V_w - V_D)/V_T} \right) \tag{3.2}$$

where: κ_p and κ_n are the slope factors for PMOS and NMOS transistors respectively, V_G, V_S, V_D and V_w are the gate voltage, source voltage, drain voltage and bulk voltage respectively, V_T is the thermal voltage and I_{o_p} and I_{o_n} are the characteristic currents (pre-exponential currents) for PMOS and NMOS transistors, respectively.

The subthreshold slope factor κ is defined as:

$$\kappa = \frac{C_{ox}}{C_{ox} + C_d} \tag{3.3}$$

where C_{ox} is the capacitance of the gate oxide per unit area and C_d the incremental capacitance of the depletion region per unit area. When V_{gb} increases The width of the depletion layer increases as well and this leads t a slight decrease of C_d and a slight increase of κ. The values of κ vary between 0.5 and 0.9 depending on the fabrication process and as κ is very slightly affected by a change in the gate voltage, it can be considered a constant value when performing an approximate analysis of the circuit in subthreshold.

The basic building blocks of the proposed system's architecture are analog circuits composed of MOS transistors operating in the subthreshold region. This design choice is motivated by two main characteristics of subthreshold region operation that makes it an attractive technique for designing analog information processing hardware architectures. Firstly, in subthreshold region the transistor's current is exponentially related to its gate to source voltage. This enables the implementation of useful mathematical

operations in current-mode circuits. Furthermore, the current flow in transistors operating in subthreshold is much lower than current flow in strong inversion region. This allows proper operation of the designed circuits with current down to several nA or even pA, resulting in ultra low power consumption.

3.2 Proposed Gaussian Function Circuit Architectures

3.2.1 Gaussian Function Circuit Related Work

RBF kernels are suitable for analog hardware implementation. This is achieved via specific analog circuits which operate in sub-threshold. They are known in the literature as Gaussian function circuits or Bump circuits. The original Bump circuit was proposed by Delbruck [60] and it is a compact structure with only 8 transistors. It consists of two subcircuits, a differential pair and a current correlator. The current correlator provides a measure of similarity between the two currents of the differential pair and its output is a Gaussian function with respect to the input voltages of the differential pair [60].

The design of Gaussian function circuits generally aims to independent tunability of the Gaussian curve's three characteristics, which are height, center (mean value) and width [60, 61]. In addition, such circuits should be ultra-low power and low-area due to the fact that in system level implementations of multivariate kernels or RBF NN with many neurons in each hidden layer, many Bump cells have to operate in parallel fashion.

Delbruck's Bump circuit [60], shown in Fig. 3.53, achieves tunability in height and center, while width is determined by the effective W/L ratio of the transistors, which also affects the height. There have been several implementations of Gaussian function circuits, with transistors operating in sub-threshold, many of them inspired by Delbruck's Bump circuit architecture. There are realizations in which the Gaussian curve's width is not electronically tunable. Instead, the width is tuned by setting a ratio of transistor W/L which necessitates the use of different Bump cells to generate different widths of the Bump [62, 59]. An example of such an architecture with ultra-low power consumption (only $13.5nW$) is analyzed in [62].

The most compact Gaussian function circuit consisting of only four transistors, but without width tunability is presented in [63]. Moreover, there are compact Bump circuits using back-gate control voltage to achieve width

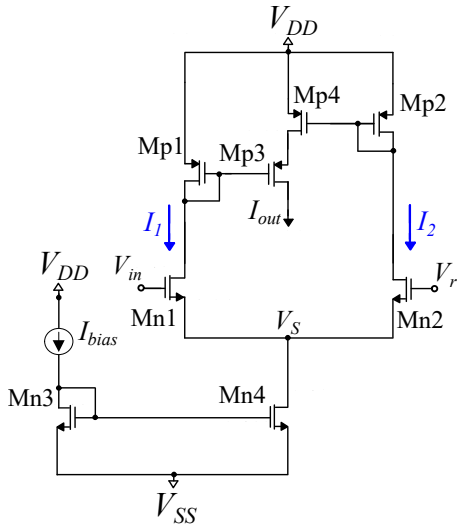
tunability [64, 65]. In contrast, there is a variety of complicated architectures for the realization of fully-tunable Gaussian functions. Examples of such cirucits are synthesized using the translinear principle and the exponential characteristics of subthreshold MOS, with increased number of transistors [66, 67, 68]. Also, an architecture based on the translinear principle and BJT transistors is presented in [69].

Other complicated architectures use extra stages to achieve tunability of the Gaussian curve's characteristics. These extra stages may be implemented with pseudo-differential transconductors [61], operational transconductance amplifiers [70], Digital to Analog Converters (DAC) [48], different values of series-connected resistances [71] or prescaling circuits with floating gate transistors [46, 72]. E.g. a complicated Gaussian function architecture is presented in [73], consisting of DAC, operational amplifier, floating resistor, multiplier and expontiator. Also, binary switches are incoporated in the Bump circuit of [74], which exhibits compensation on temperature and V_{DD} variations and has peak output currents exceeding $1\mu A$. It is a modification of the architecture presented in [75].

Apart from the previous implementations in which transistors operate in sub-threshold, there are architectures in which transistors operate in strong inversion. There is a trade-off between low power consumption in sub-threshold and higher speed operation in strong inversion. Two Gaussian generation circuits consisting of transistors operating in both regions are presented in [72, 49]. An implementation which consists of a symmetric current correlator and a differential pair with extra current sources is presented in [76], with the ability to operate both in strong and weak inversion regions.

There are also other architectures that consist of transistors operating exclusively in strong inversion region. There are also Gaussian circuits without width tunability [77, 78]. More specifically, a CMOS and a BiCMOS RBF circuit [77] and a Bump circuit which consists of a voltage correlator [78] are presented. The appropriate tunability in Gaussian curve's characteristics is achieved through complicated implementations. A complicated and accurate implementation of a Gaussian function circuit using a 4_{th} -order approximation (based on Taylor mathematical series) with current mode squaring circuits operating in saturation is presented in [79]. A Gaussian membership function architecture is implemented in [80] based on current mode squaring and exponential approximation circuits and in [81] based on a differential coupled amplifier (using two differential pairs). Differential pairs and a minimum value circuit are used for the realization of a Gaussian function output [82]. A modified current rectifier is proposed in [83] for

the realization of the Gaussian function. Moreover, a different architecture based on current conveyors is proposed in [84].



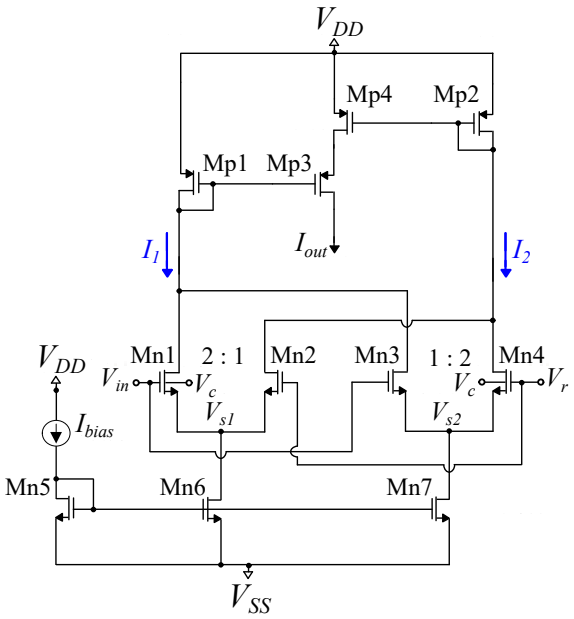
Σχήμα 3.2: Delbruk's Bump circuit.

Gaussian function circuits and Bump circuits are useful in a variety of applications. In modern neuromorphic systems there is a need for tunable Gaussian function circuits to implement weight update mechanisms [85] and adaptive "stop learning" procedures [65, 86]. Tunable Gaussian function circuits have also been used in architectures implementing SVM [48, 49, 47], support vector domain description algorithm [52], k-means clustering algorithm [44], RBF NN [62, 45], RBF classifiers [46] and Gaussian Kernels used in support vector regression implementations for approximate computing [53]. Memristive RBF NN architectures are also being investigated using hybrid CMOS-memristor Bump circuits [87]. Bump circuits have also been used in various sensor applications such as image edge detection [88] and unsupervised anomaly detection [71].

3.3 Proposed Circuit Architecture

The architecture of the original Bump circuit [60] in Fig.3.2, can be modified to achieve electronically and adjustable width, independent of the other

Bump parameters. This is achieved by the proposed architecture[89] depicted in Fig.3.3, consisting of a differential difference pair ($M_{n1} - M_{n4}$) and a modified current correlator ($M_{p1} - M_{p4}$).



Σχήμα 3.3: Proposed Gaussian function circuit.

3.3.1 Differential Difference Pair

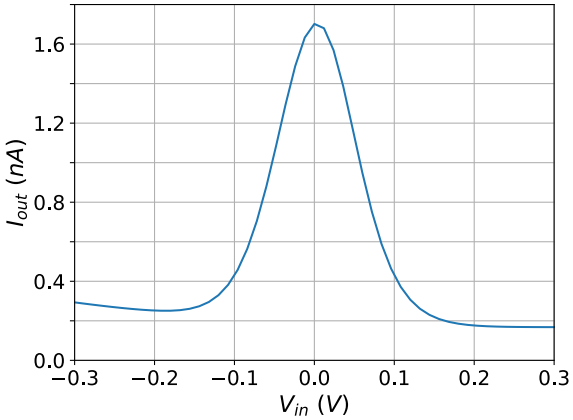
The differential difference pair consists of two differential pairs which produce currents with sigmodial shaped curves of adjustable slopes. The differential difference pair's transistors M_{n1} and M_{n3} have their bulks connected to a control voltage V_c. Input voltage V_{in} is connected to the gates of M_{n1} and M_{n3} while a parameter voltage V_r is applied to the gates of M_{n2} and M_{n4}. To increase the linearity of the differential difference pairs block, the ratio of the transistors' M_{n1}-M_{n2} and M_{n3}-M_{n4} sizes is set to 2 instead of 1 [90, 91, 92].

Transistors' dimensions are summarized in Table 3.1. The transistor sizes are fixed and selected so that they result in a good balance of the circuit's performance. It is the topology and control nodes that offer the desirable tunability. The Bump circuit structure is biased with current I_{bias}. In our implementation, we set the power supply rails at $V_{DD} = -V_{SS} = 0.3V$, and all transistors operate in the sub-threshold region. The proposed

architecture provides a Gaussian function output as shown in Fig 3.4 (for $V_r = 0$, $V_c = -300mV$ and $I_{bias} = 1nA$).

Πίνακας 3.1: MOS Transistors Dimensions.

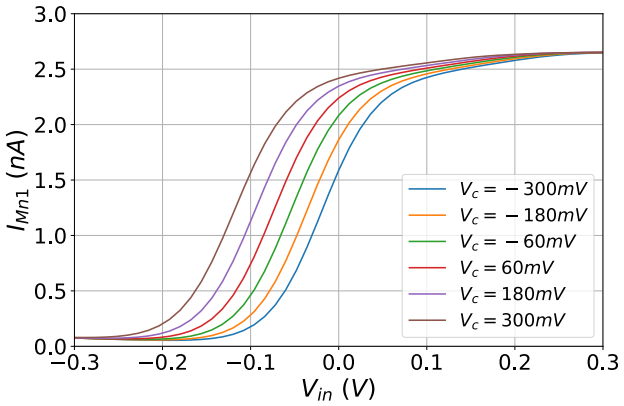
Block	**W/L** (μ**m**)	**Current Correlator**	**W/L** (μ**m**)
M_{n1},M_{n4}	1.6/0.4	M_{p1},M_{p3}	0.4/1.6
M_{n2},M_{n3}	0.8/0.4	M_{p2}	0.4/1.6
M_{n5}	0.8/1.6	M_{p4}	0.6/1.6
M_{n6},M_{n7}	1.2/1.6	-	-



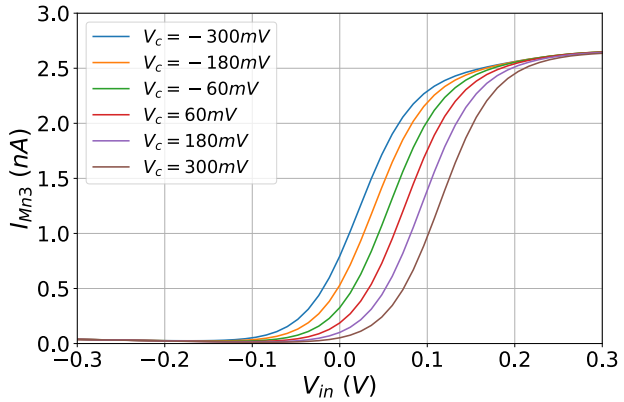
Σχήμα 3.4: Output current of the Gaussian function circuit for $V_r = 0$, $V_c = -300mV$ and $I_{bias} = 1nA$ (post-layout simulation).

In each differential pair, connecting the bulk of one of the two transistors to a control voltage V_c results in a shift of the differential currents along the V_{in} axis by altering V_c. The currents of the first differential pair (M_{n1} and M_{n2}) are shifted in a symmetric way about the origin ($V_{in} = 0$) relative to the currents of the second differential pair (M_{n3} and M_{n4}). As shown in Fig 3.5 an increase in V_c results in a shift of the current's (I_{Mn1}) transfer curve to the left. The same change is applied to current I_{Mn2}. For the second differential pair, an increase in V_c results in a shift of the currents' transfer curve to the right, as shown in Fig3.6 for the case of I_{Mn3}.

Bulk-controlled implementations of Bump circuits are also presented in [64, 65]. In contrast to those designs, in the proposed architecture there is

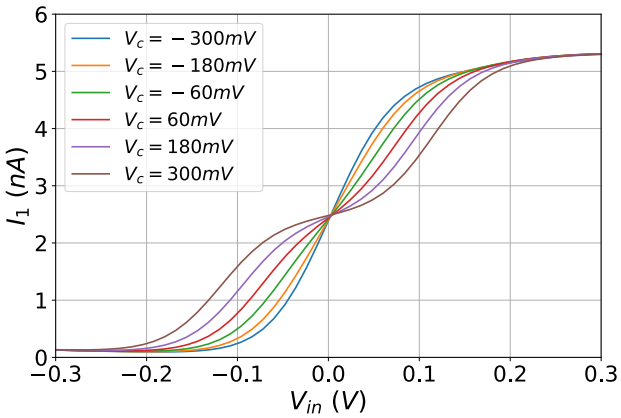


Σχήμα 3.5: Displacement of current I_{Mn1} ια παραμετερ ολταγε V_c, φορ $I_{bias} = 1nA$ ανδ $V_r = 0mV$ (post-layout simulation).

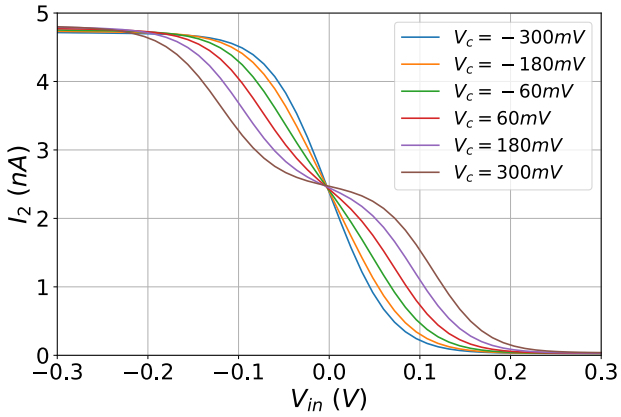


Σχήμα 3.6: Displacement of current I_{Mn3}via parameter voltage V_c, for $I_{bias} = 1nA$ and $V_r = 0mV$ (post-layout simulation).

summation of currents from the two differential pairs. Adding the currents flowing through M_{n1} and M_{n3} achieves the desired variability in the slope of current I_1, as shown in Fig 3.7. Similarly for M_{n2} ανδ M_{n4}, currents I_{Mn2} and I_{Mn4} are summed, thus tunability in the slope of current I_2 is achieved , as shown in Fig 3.8. The fact that the current correlator's input currents I_1 and I_2 have adjustable slopes leads to the desired tunability in the Gaussian output curve's width.



Σχήμα 3.7: Tuning of current I_1 via parameter voltage V_c, φορ $I_{bias} = 1nA$ and $V_r = 0mV$ (post-layout simulation).

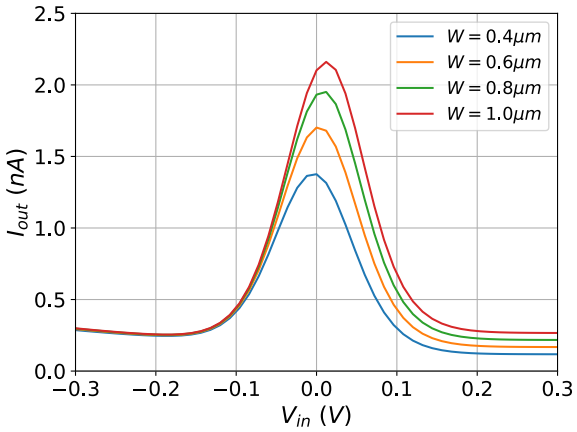


Σχήμα 3.8: Tuning of current I_2 via parameter voltage V_c, for $I_{bias} = 1nA$ and $V_r = 0mV$ (post-layout simulation).

3.3.2 Modified Current Correlator

In the proposed implementation a modified current correlator has been used, in order to tackle the inherent asymmetries in the standard current correlator topology. In this work, the transistors' dimensions are not all equal (non-symmetric implementation). While transistors M_{p1}, M_{p2} and M_{p3} have $\frac{W}{L} = \frac{0.4\mu m}{1.6\mu m}$, the transistor's M_{p4} value of W/L has been changed. This modification enables the elimination of small dc offsets of the Bump's transfer curve along the V_{in} axis, with $\frac{W}{L} = \frac{0.6\mu m}{1.6\mu m}$ proving to be the optimal value for this purpose, as shown in Fig. 3.9.

In our implementation, we want to achieve more symmetrical results in the Gaussian function curve for the minimum bias current of $1nA$. The available values of width are multiples of a single transistor with $W = 0.2\mu m$. Through parametric simulations for different values of transistor's width (M_{p4}) demonstrated in Fig. 3.9, it can be observed that for values of W greater than $W = 0.6\mu m$ there is a considerable offset of the Gaussian curve's center along the V_{in} axis. Furthermore, comparing the two Gaussian curves for $W = 0.6\mu m$ and $W = 0.4\mu m$, the one with $W = 0.6\mu m$ is slightly more symmetrical across the whole range of V_{in} values. Thus, we choose $W = 0.6\mu m$ as the optimal value for our design (layout).



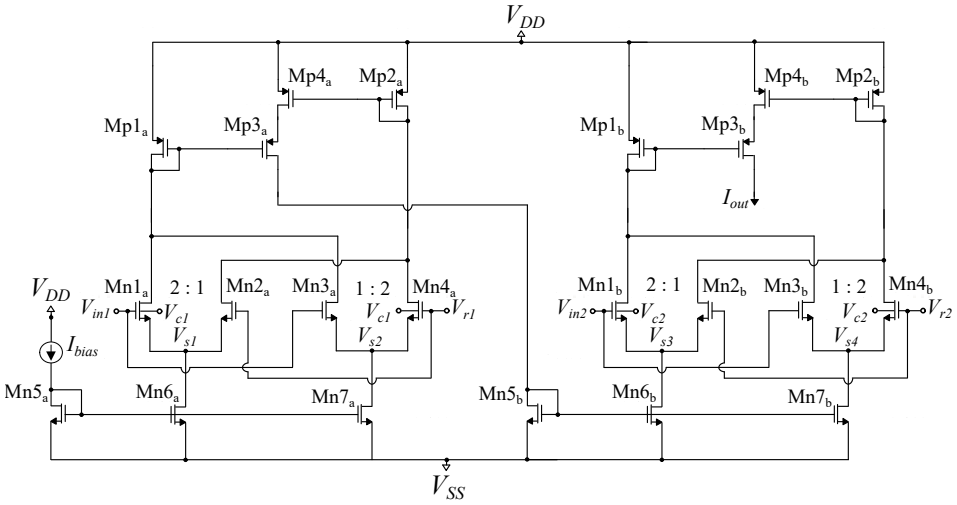
Σχήμα 3.9: Selection of the optimal W value of transistor M_{p4}, for $I_{bias} = 1nA$, $V_c = -300mV$ and $V_r = 0mV$ (schematic simulation).

The height, width and center of the produced Gaussian function are electronically tuned via three circuit's parameters (I_{bias}, V_c and V_r respectively). A single current source I_{bias} provides the bias current for both differential pairs and controls the height of the Bump while as explained above a control voltage V_c at the bulks of M_{n1} and M_{n4} alters the width. The center of the Gaussian function is set by the parameter voltage V_r. The output current of the Bump reaches its maximum value when the input voltage matches the parameter voltage ($V_{in} = V_r$).

3.3.3 2-D Implementation

A typical characteristic of Bump circuit architectures is the dimensional scalability. It is realized with two or more cascaded Bump circuits, in which the output current of one Bump circuit is used as bias current for the next

identical Bump cell. Each Bump cell has each own input voltage V_{in} and parameter voltages V_c and V_r, while only the first Bump cell is biased with a current I_{bias}. A cascaded $2-D$ implementation using the proposed Gaussian function circuit as a basic component, is shown in Fig. 3.10. The scalability of the proposed Bump circuit makes it an interesing candidate for implementations of multivariate RBFs.



Σχήμα 3.10: Proposed 2-D implementation.

3.4 Circuit Theoretical Analysis

In this Section, a mathematical analysis of the proposed Gaussian circuit is presented. All transistors operate in the sub-threshold region and we use the MOS model in [59], i.e. for the PMOS and NMOS are respectively:

$$I_{pmos} = I_{o_p} e^{\kappa_p (V_w - V_G)/V_T} \left(e^{(V_S - V_w)/V_T} - e^{(V_D - V_w)/V_T} \right) \tag{3.4}$$

$$I_{nmos} = I_{o_n} e^{\kappa_n (V_G - V_w)/V_T} \left(e^{(V_w - V_S)/V_T} - e^{(V_w - V_D)/V_T} \right) \tag{3.5}$$

In this work, I_{o_p} and I_{o_n} are the pre-exponential currents of transistors M_{p1} and M_{n2} respectively. For every transistor we consider, we use a scaling factor (m), i.e. mI_{o_p} or mI_{o_n}, to capture relative W/L value according to Table 3.1.

3.4.1 Differential Difference Pair Analysis

In this work two transistors, M_{n1} and M_{n4}, of the differential difference pair are bulk-controlled and their bulks are connected to the parameter voltage V_c, as shown in Fig.3.2 All four transistors M_{n1} to M_{n4} operate in saturation region with ολταγες $V_D >> V_S$. Thus, we are led to a simplified version of Eq.3.5. More specifically for transistors M_{n1} and M_{n4} their saturation currents are given by the following expressions (using the appropriate aspect ratio W/L):

$$I_{M_{n1}} = 2\ I_{o_n} e^{(\kappa_n V_{in} - V_{S1} + (1-\kappa_n)V_c)/V_T} \tag{3.6}$$

$$I_{M_{n4}} = 2\ I_{o_n} e^{(\kappa_n V_r - V_{S2} + (1-\kappa_n)V_c)/V_T} \tag{3.7}$$

Transistors M_{n2} and M_{n3} have their bulks connected to V_{SS}. Their saturation currents are given by the following expressions (using the appropriate aspect ratio W/L):

$$I_{M_{n2}} = I_{o_n} e^{(\kappa_n V_r - V_{S1} + (1-\kappa_n)V_{SS})/V_T} \tag{3.8}$$

$$I_{M_{n3}} = I_{o_n} e^{(\kappa_n V_{in} - V_{S2} + (1-\kappa_n)V_{SS})/V_T} \tag{3.9}$$

Transistors $M_{n5} - M_{n7}$ operate in saturation as current mirrors. According to their relative W/L values, the bias current I_{bias} is equal to:

$$I_{bias} = 2\frac{I_{M_{n1}} + I_{M_{n2}}}{3} \tag{3.10}$$

$$I_{bias} = 2\frac{I_{M_{n3}} + I_{M_{n4}}}{3} \tag{3.11}$$

for the two differential pairs. Combining Eq.3.6,3.8 and 3.10 we conclude:

$$I_{M_{n1}} = \frac{3I_{bias}}{2 + e^{(\kappa_n(V_r - V_{in}) + (\kappa_n - 1)(V_c - V_{SS}))/V_T}} \tag{3.12}$$

In order to simplify the expression, we set $\Delta V = V_r - V_{in}$ and $V_{c1} = V_c - V_{SS}$ and the previous equation is transformed in the following way:

$$I_{M_{n1}} = \frac{3I_{bias}}{2 + e^{(\kappa_n \Delta V + (\kappa_n - 1)V_{c1})/V_T}} \tag{3.13}$$

In this step we combine Eq.3.7,3.9 and 3.11. The drain current of M_{n3} is given by:

$$I_{M_{n3}} = \frac{3I_{bias}}{2 + 4e^{(\kappa_n \Delta V + (1-\kappa_n)V_{c1})/V_T}} \tag{3.14}$$

The current I_1 is equal to the sum of $I_{M_{n1}}$ and $I_{M_{n3}}$, as shown in Fig.3.2 The total expression of I_1 is:

$$I_1 = \frac{3I_{bias}}{2 + e^{(\kappa_n \Delta V + (\kappa_n - 1)V_{c1})/V_T}} + \frac{3I_{bias}}{2 + 4e^{(\kappa_n \Delta V + (1-\kappa_n)V_{c1})/V_T}} \tag{3.15}$$

The current I_2 is equal to the sum of $I_{M_{n2}}$ and $I_{M_{n4}}$, as shown in Fig.3.2. By using the same methodology we can calculate the total expression of I_2 which is given by:

$$I_2 = \frac{3I_{bias}}{2 + e^{(-\kappa_n \Delta V + (\kappa_n - 1)V_{c1})/V_T}} + \frac{3I_{bias}}{2 + 4e^{(-\kappa_n \Delta V + (1-\kappa_n)V_{c1})/V_T}} \tag{3.16}$$

3.4.2 Modified Current Correlator Analysis

The current correlator's transistors' dimensions are shown in Table 3.1. Supposing that the output node's voltage (drain of transistor M_{p3}) is sufficiently low to ensure operation in saturation region, its drain current, using Eq.3.4, is given by the following expression:

$$I_{out} = I_{M_{p3}} = I_{o_p} e^{(V_{D_{Mp4}} - \kappa_p V_{D_{Mp1}} + (\kappa_p - 1)V_{DD})/V_T} \tag{3.17}$$

where $V_{D_{Mp1}}$ and $V_{D_{Mp4}}$ are drain voltage of transistors M_{p1} and M_{p4}. Transistor's M_{p1} drain current is given by:

$$I_1 = I_{M_{p1}} = I_{o_p}\ e^{\kappa_p (V_{DD} - V_{D_{Mp1}})/V_T} \tag{3.18}$$

Transistor's M_{p2} current is given by:

$$I_2 = I_{M_{p2}} = I_{o_p}\ e^{\kappa_p (V_{DD} - V_{D_{Mp2}})/V_T} \tag{3.19}$$

Transistor M_{p4} operates in triode region. The drain current of transistor M_{p4} is equal to the drain current of transistor M_{p3} and given by:

$$I_{M_{p4}} = \frac{3}{2} I_{o_p} e^{\kappa_p (V_{DD} - V_{G_{Mp4}})/V_T} \left(e^{(V_{DD} - V_{DD})/V_T} - e^{(V_{D_{Mp4}} - V_{DD})/V_T} \right) \tag{3.20}$$

$$I_{out} = \frac{3}{2} I_{o_p} e^{\kappa_p (V_{DD} - V_{D_{Mp2}})/V_T} \left(1 - e^{(V_{D_{Mp4}} - V_{DD})/V_T} \right) \tag{3.21}$$

where $V_{D_{Mp2}} = V_{G_{Mp4}}$ and $I_{out} = I_{M_{p3}} = I_{M_{p4}}$, as shown in Fig. 3.2.

Solving Eq. 3.17-3.19 for $V_{D_{Mp1}}$, $V_{D_{Mp2}}$ and $V_{D_{Mp4}}$, and replacing them into Eq.3.21 we get:

$$I_{out} = \frac{3}{2} I_2 \left(1 - \frac{I_{out}}{I_1}\right) \tag{3.22}$$

The expression of current correlator for the proposed topology is expressed in the following way:

$$I_{out} = \frac{\frac{3}{2} I_1 I_2}{I_1 + \frac{3}{2} I_2} \tag{3.23}$$

3.4.3 Bump Circuit Analysis

In order to simplify the expression of Eq.3.15 and Eq.3.16 we set $x = \kappa_n \Delta V / V_T$ and
$y = (\kappa_n - 1) V_{c1} / V_T$:

$$I_1 = \frac{3 I_{bias}}{2} \left(\frac{4 + e^x e^y + 4 e^x e^{-y}}{2 + e^x e^y + 4 e^x e^{-y} + 2 e^{2x}}\right) \tag{3.24}$$

By using the same methodology the simplified expression of I_2 which is given by:

$$I_2 = \frac{3 I_{bias}}{2} \left(\frac{4 e^{2x} + e^x e^y + 4 e^x e^{-y}}{2 + e^x e^y + 4 e^x e^{-y} + 2 e^{2x}}\right) \tag{3.25}$$

Combining Eq.3.23-3.25:

$$I_{out} = \frac{3 I_{bias}}{2} \frac{(6 e^{-x} + 3M)(2 e^x + M)}{(e^x + e^{-x} + M)(6 e^x + 4 e^{-x} + 5M)} \tag{3.26}$$

Finally, the output current of the fully tunable Bump circuit is expressed using hyperbolic cosine equation as:

$$I_{out} = \frac{3 I_{bias}}{2} \frac{12 + 3M^2 + 12M \cosh x}{(2 \cosh x + M)(6 e^x + 4 e^{-x} + 5M)} \tag{3.27}$$

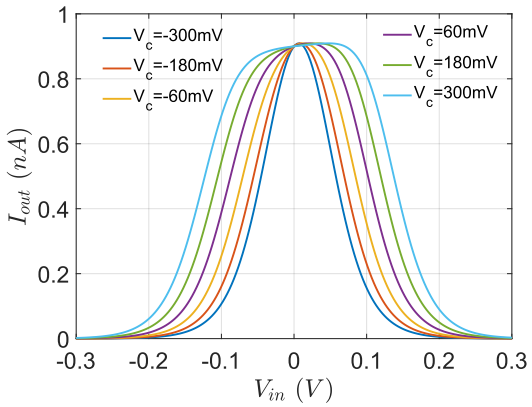
where intermediate variable M is given by:

$$M = 2 e^{-y} + \frac{e^y}{2} = 2 e^{-(\kappa_n - 1)(V_c - V_{SS})/V_T} + \frac{e^{(\kappa_n - 1)(V_c - V_{SS})/V_T}}{2} \tag{3.28}$$

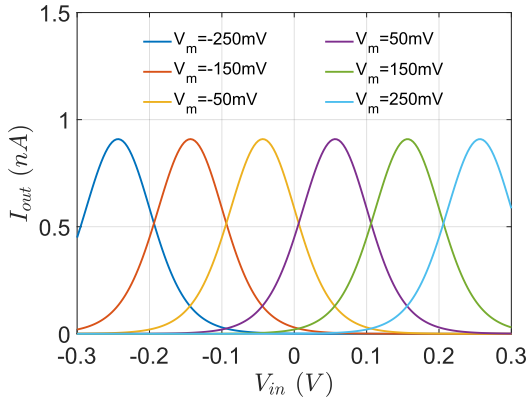
and x has been defined as:

$$x = \frac{\kappa_n (V_r - V_{in})}{V_T} \tag{3.29}$$

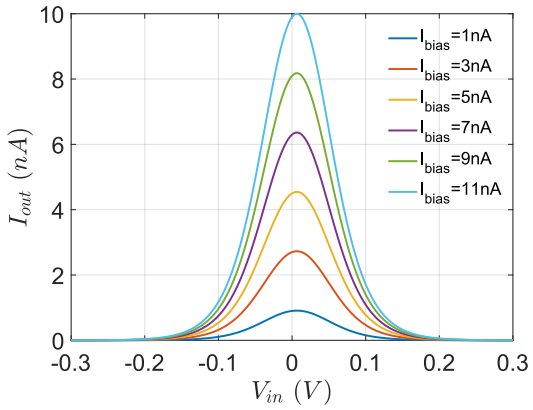
The Gaussian circuit's current I_{out} depends on the input voltage V_{in}, and parameters I_{bias}, V_r and V_c. The theoretical output current of the Gaussian circuit, according to Eq.3.27 is presented in Figs.3.11,3.12,3.13 Each parameter (V_c, V_r and I_{bias}) independently tunes a characteristic of the Gaussian curve. We alter the value of one parameter while the others are kept constant. Parameter V_c adjusts the Gaussian curve's width as shown in Fig.3.11 Parameter V_r sets the mean value (center) of the Gaussian function's output, as shown in Fig.3.12 Parameter I_{bias} scales the height of the Gaussian curve, as shown in Fig.3.13 The results of the theoretical analysis illustrate the correct operation of the proposed Gaussian function circuit.



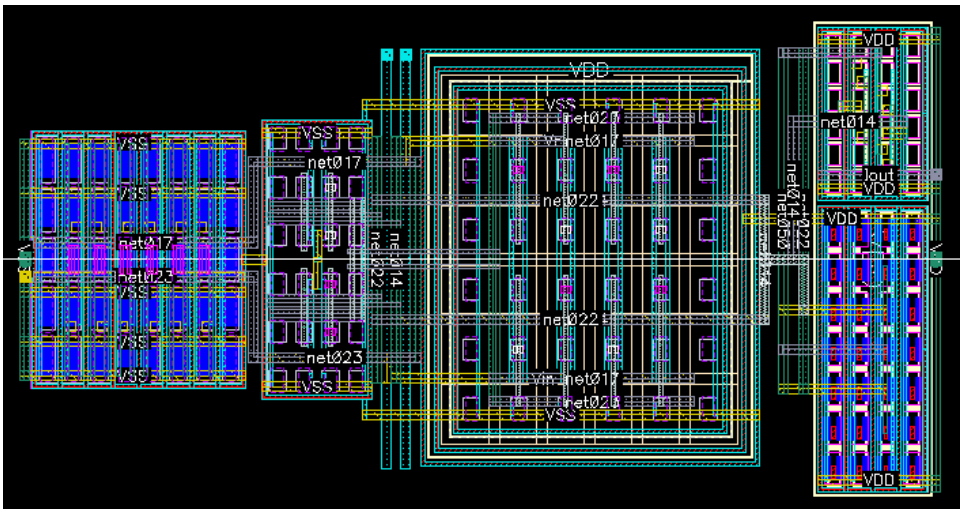
Σχήμα 3.11: Width tuning in Theoretical output function of the Bump circuit, for $I_{bias} = 1nA$ and $V_r = 0mV$ (MATLAB simulation).



Σχήμα 3.12: Center adjustment in Theoretical output function of the Bump circuit, for $I_{bias} = 1nA$ and $V_c = -300mV$ (MATLAB simulation).



Σχήμα 3.13: Height scaling in Theoretical output function of the Bump circuit, for $V_c = -300mV$ and $V_r = 0mV$ (MATLAB simulation).



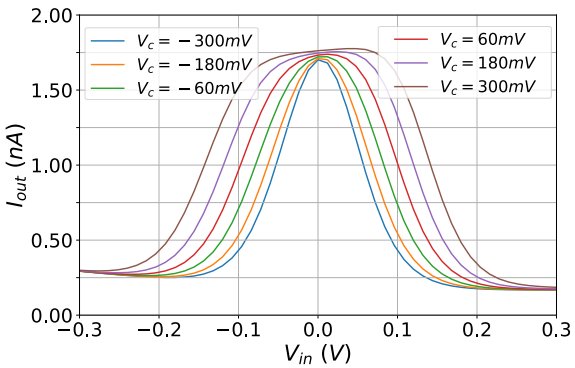
Σχήμα 3.14: Layout of the implemented Gaussian function circuit.

3.5 Simulation Results

The proposed ultra-low power, low-voltage, fully tunable, bulk-controlled Gaussian function circuit has been designed in TSMC $90nm$ CMOS process, using the Cadence IC design suite. The power supply rails are $V_{DD} = -V_{SS} = 0.3V$, and all transistors operate in the sub-threshold region. The Gaussian circuit's simulation results are from post-layout simulations. The

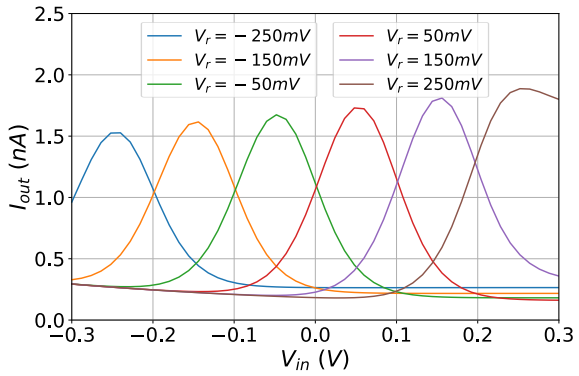
layout of the proposed 1-D Gaussian function architecture (Fig.3.3 is shown in Fig.3.14, where the area is $50.45\mu m$ Ξ $25.85\mu m$. In order to deal with manufacturing considerations and mismatches, dummy transistors were used and transistor matching is applied only for transistors with the same length (L) based on the common-centroid technique.

The width tunability of the proposed Gaussian function circuit, via parameter voltage V_c, is shown in Fig.3.15, for constant values of $I_{bias} = 1nA$ and $V_r = 0$. An increase in parameter voltage V_c leads to an increase in the Gaussian curve's width. The mean value of the derived Gaussian function is determined by voltage V_r, as illustrated in Fig.3.16, for constant values of $I_{bias} = 1nA$ and $V_c = -300mV$. Proper operation is achieved for a wide parameter voltage V_r range. Its values are between $V_{r_{min}} = -250mV$ and $V_{r_{max}} = 250mV$ (83.3% of the power supply range). The height of the Bump output current is set by the bias current I_{bias} as shown in Fig.3.17, for constant $V_r = 0$ and $V_c = -300mV$. The tunability of the Gaussian function's characteristics derived from post-layout simulations and shown in Figs.3.15-3.17 matches the expected behavior depicted in Figs.3.11-3.13.

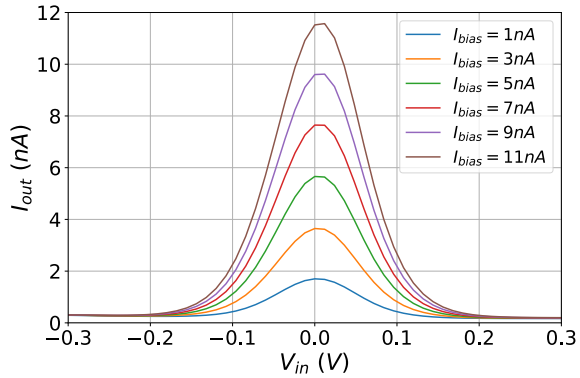


Σχήμα 3.15: Width tuning of the output current with voltage V_c, for $I_{bias} = 1nA$ and $V_r = 0mV$ (post-layout simulation).

The output current of two cascaded Bump cells is represented in $3-D$ space and is depicted in Figs.3.18-3.21. The independent tunability of the Gaussian curve's characteristics (width, height, center) is also achieved for $2-D$ RBFs. The first of the two cascaded Bump circuits of the architecture in Fig.3.10, is biased with $I_{bias} = 2nA$. Setting control parameters $V_{c1} = V_{c2} = 300mV$ and $V_{r1} = V_{r2} = 0V$, maximum width is achieved as shown in Fig.3.18. By keeping I_{bias}, V_{r1} and V_{r2} values constant while altering $V_{c1} = V_{c2} = -300mV$, the Gaussian curve's width is independently adjusted,



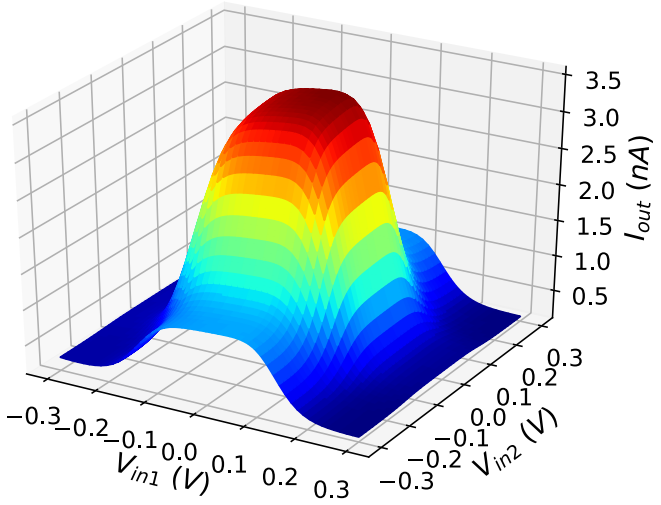
Σχήμα 3.16: Center adjustment of the output current with voltage V_r, for $I_{bias} = 1nA$ and $V_c = -300mV$ (post-layout simulation).



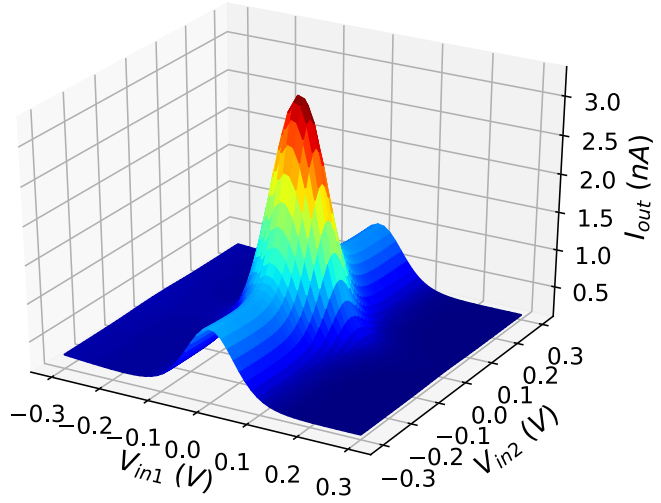
Σχήμα 3.17: Height scaling of the output current with bias current I_{bias}, for $V_c = -300mV$ and $V_r = 0mV$ (post-layout simulation).

as shown in Fig.3.19. Keeping the parameter voltages constant at $V_{c1} = V_{c2} = -300mV$ and $V_{r1} = V_{r2} = 0V$ and increasing $I_{bias} = 4nA$, height scaling of the $2-D$ output is achieved, as shown in Fig.3.20. The tunability of the RBF's center is presented in Fig.3.21, by setting parameter voltages $V_{r1} = V_{r2} = 100mV$ while the other parameters are the same as in Fig.3.20. The $2-D$ implementation's results confirm the desirable scalability of the proposed circuit.

The sensitivity behavior has been evaluated using the Monte-Carlo analysis tool for $N = 100$ points. The corresponding histogram for the Bump circuit's center of voltage is shown in Fig.3.22. The center of the voltage is $V_{mean} = 1.7mV$, and the standard deviation is $\sigma_V = 4.3mV$. It confirms



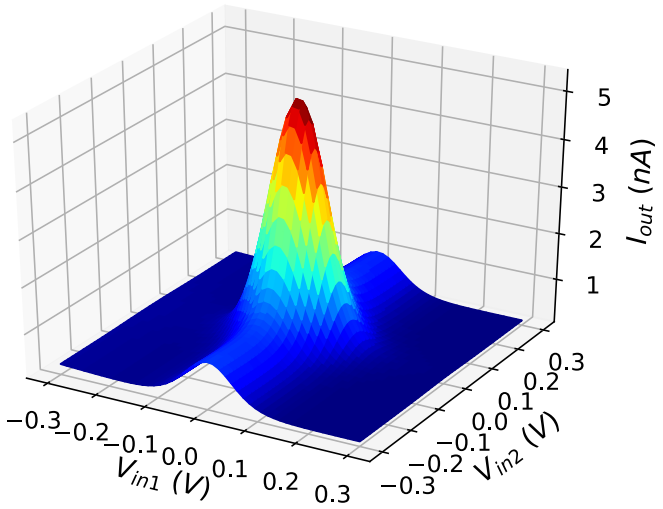
Σχήμα 3.18: A $2-D$ Gaussian Function with bias current $I_{bias} = 2nA$, $V_r = 0V$ and $V_c = 300mV$ (post-layout simulation).



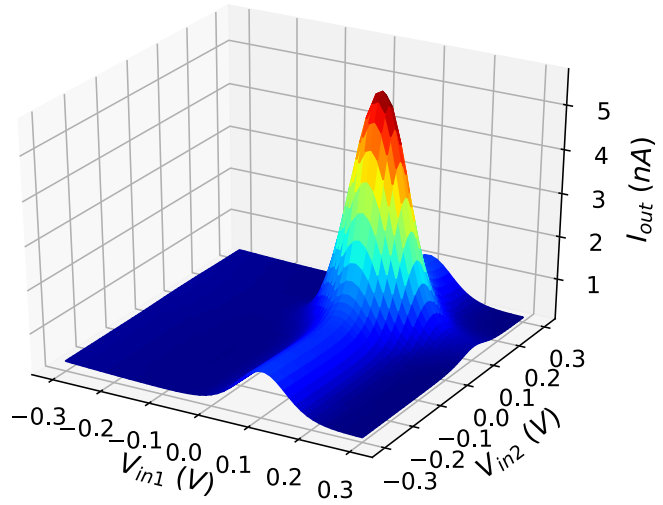
Σχήμα 3.19: A $2-D$ Gaussian Function with bias current $I_{bias} = 2nA$, $V_r = 0V$ and $V_c = -300mV$ (post-layout simulation).

the correct performance and accuracy of the proposed circuit.

The simulation results confirm the proper operation and performance of the proposed Gaussian function circuit, in accordance with the theoretical

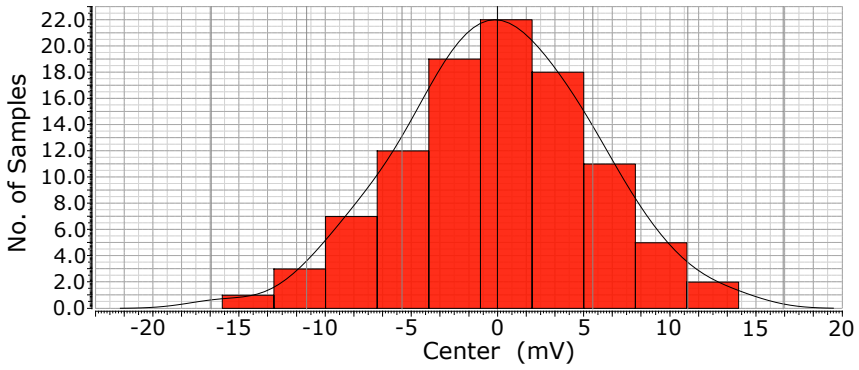


Σχήμα 3.20: A $2-D$ Gaussian Function with bias current $I_{bias} = 4nA$, $V_r = 0V$ and $V_c = -300mV$ (post-layout simulation).



Σχήμα 3.21: A $2-D$ Gaussian Function with bias current $I_{bias} = 4nA$, $V_r = 100mV$ and $V_c = -300mV$ (post-layout simulation).

analysis. There are specific factors resulting in slight variations between the theoretical results as shown in Figs.3.11-3.13 and simulation results as shown in Figs.3.15-3.17. In theoretical analysis the sub-threshold slope factors



Σχήμα 3.22: Center value sensitivity via Monte-Carlo simulation.

κ_n and κ_p are considered to be constant, which is typical in approximate analysis of sub-threshold circuits. However, an increase in gate-to-bulk voltage results in a small increase in the slope factors' values [59]. This approximation justifies small variations in the form of the Gaussian function output. For transistors $M_{n5} - M_{n7}$ (current mirrors) the current mirroring ratio for small currents (1νA-5νA) diverges from the aspect ratio ($\frac{3}{2}$). This is illustrated in the height difference of the Gaussian function output between theoretical and simulation results. Thus, in simulation results the output current exceeds $1nA$ for bias current $I_{bias} = 1nA$. This is vital for the scalability of the proposed architecture without having degradation of the output current. In Figs.3.11-3.13 and Figs.3.15-3.17 there is a small dc offset of up to $10mV$ for increased bias current (more than $5nA$). This is a result of the non-symmetric current correlator topology, which achieves the appropriate symmetry in small currents ($I_{bias} = 1nA$) and sufficient current scaling.

3.6 Comparison Study and Discussion

The proposed Gaussian function circuit is compared with recent literature works in terms of performance and design characteristics and the results are summarized in Table 2. All the implementations are in CMOS process except of [87] which is designed in LT-SPICE with PTM transistor models and MS memristor model and [64] which uses quad MOS transistor arrays ALD1106 and ALD1107, manufactured by Advanced Linear Devices (ALD).

Low power consumption is a vital characteristic of Bump circuits that are used for the implementation of RBF NNs. The previous works opera-

Πίνακας 3.2: Performance Summary and Comparison.
*Add extra stages in order to achieve width tunability.

	Technology	**Power Consumption**	**Power Supply**	**Minimum I_{bias}**	**No of Transistors**	**Ind. Param. Tunability**
This work	$90nm$	$3.9nW$	$0.6V$	$1nA$	11	YES
[61]	$130nm$	$18.9nW$	$3V$	$1nA$	14	YES
[62]	$180nm$	$13.5nW$	$0.9V$	$40nA$	$*9$	YES
[87]	*Discrete*	$4.1\mu W$	$0.7V$	$1\mu A$	22	YES
[45]	$130nm$	$10.5\mu W$	$1.2V$	-	$*13$	YES
[63]	$180nm$	-	$1.3-2V$	-	4	NO
[64]	*Discrete*	-	$5V$	$2nA$	10	YES
[66]	$350nm$	$650nW$	$1.3V$	$50nA$	17	YES
[67]	$180nm$	$350nW$	$0.7V$	$50nA$	31	YES
[48]	$180nm$	-	$1.8V$	$100nA$	$*11$	YES
[46]	$500nm$	$90\mu W$	$3.3V$	-	19	YES
[72]	$180nm$	$160nW$	$0.75V$	$35nA$	8	YES
[49]	$180nm$	-	$1.8V$	$50nA$	14	YES
[79]	$180nm$	$100\mu W$	$1V$	$10\mu A$	30	YES
[80]	$350nm$	-	$3.3V$	$10\mu A$	45	YES
[81]	$350nm$	$220\mu W$	$3.3V$	$9\mu A$	$*14$	YES
[82]	$180nm$	$27\mu W$	$1.8V$	$2\mu A$	15	YES
[83]	$180nm$	$23.7\mu W$	$2V$	$5\mu A$	32	YES

ting with ultra-low power consumption are [61] at $18.9nW$ and [62] at only $13.5nW$. Our work, provides a significant improvement in power consumption compared to literature, operating at only $3.9nW$ (reduction of power consumption by 71.1% compared to $13.5nW$).

Power supply voltages used in literature implementations range from $5V$ down to $0.7V$. The minimum power supply among the previous works is achieved in [87, 67] ($0.7V$) and [72] ($0.75V$). In the proposed architecture, power supply voltage is further reduced at only $0.6V$. Our implementation is biased with minimum $I_{bias} = 1nA$, which is smaller than $2nA$ [64] and equal to $1nA$ [61].

The implementation with the minimum number of transistors (only 4) is presented in [63], but it lacks tunability in width. The proposed Gaussian function circuit, is a compact and low area implementation, consisting of

only 11 transistors. Architectures with similar number of transistors are [72] (8 transistors) [64] (10 transistors) and [61, 49] (14 transistors). The compact Bump circuit in [62] (9 transistors) uses extra logic and multiplexer circuits to update the output curve's width, while for the same purpose in [48] (11 transistors) an extra DAC is used. Also, in [45] (13 transistors) extra switches and multiplexer circuits are used while in [81] (14 transistors) an extra stage for the controlling current implementation is used in order to achieve the desirable behavior. The area of the proposed circuit (layout) is $1304\mu m^2$ and only [61] has a smaller layout area ($988\mu m^2$), while in other implementations [62, 66, 67, 46, 83] it ranges from $2475\mu m^2$ up to $13054\mu m^2$.

3.7 2nd Proposed Gaussian Function Circuit Architecture

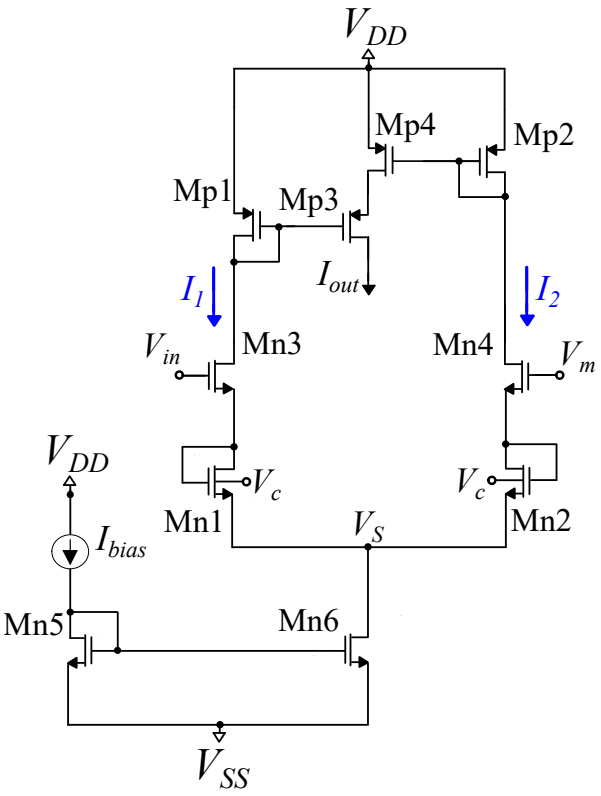
3.7.1 Proposed Architecture and Analysis

In this section, the proposed architecture and its mathematical analysis are presented. The implemented bump circuit is an alteration of the original Delbruck's SB [60]. This modification achieves ultra low power consumption ($4nW$), low voltage power supply ($V_{DD} = -V_{SS} = 0.3V$) and electronically controllable height, width and center of the produced Gaussian curve.

The proposed bump circuit is shown in Fig.3.23 and its output current is a Gaussian function in relation to the input voltage (V_{in}) as shown in Fig.3.24. Our architecture is composed of two subcircuits. In this work a modified current correlator (non-symmetric implementation) is used, and the differential pair of Delbruck's SB is substituted by a NMOS bulk-controlled block ($M_{n1} - M_{n4}$ and M_{n6}) [60]. Tunability is achieved with the use of three circuit's parameters.

Parameter voltage V_c, which is connected to the bulks of transistors $M_{n1} - M_{n2}$, adjusts the width of the Gaussian curve, while parameter voltage V_m tunes its center. All other NMOS have their bulks connected to V_{SS} whereas the bulks of all PMOS are connected to V_{DD}. Furthermore, bias current I_{bias} sets the height of the Gaussian function. The tunability of these three characteristics (width, height, center) is further explained via Mathematical Analysis and Simulation Results. All transistors operate in the sub-threshold region and their dimensions are summarized in Table3.3 .

The mathematical analysis of the proposed bump circuit is presented below. The MOS model in [**?**] is used to describe the operation of the circuit's transistors in the sub-threshold region according to the following



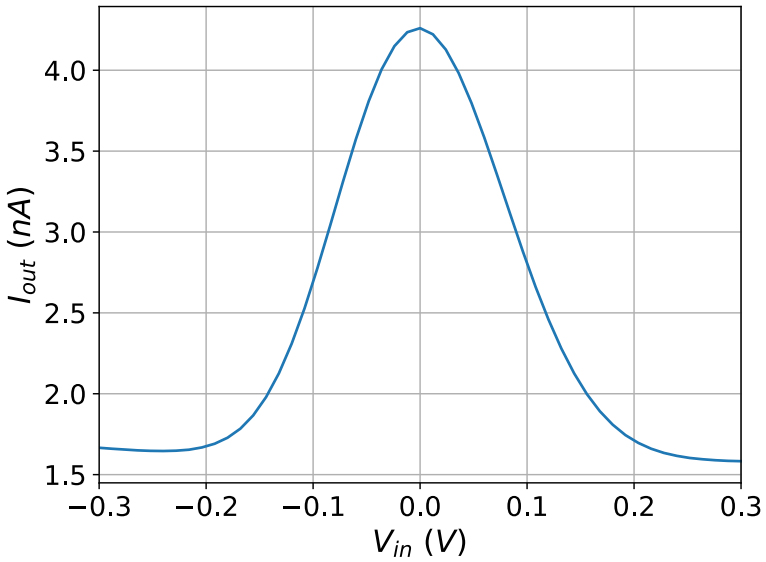
Σχήμα 3.23: 2nd Proposed Bump circuit.

Πίνακας 3.3: MOS Transistors Dimensions.

Block	**W/L** (μ**m**)	**Current Correlator**	**W/L** (μ**m**)
M_{n1}-M_{n2}	1.0/0.1	M_{p1},M_{p3}	0.8/0.1
M_{n3},M_{n4}	1.0/1.6	M_{p2}	0.8/0.4
M_{n5},M_{n6}	0.2/1.6	M_{p4}	2.4/0.4

equations for PMOS and NMOS respectively:

$$I_{pmos} = I_{o_p} e^{\kappa_p (V_w - V_G)/V_T} \left(e^{(V_S - V_w)/V_T} - e^{(V_D - V_w)/V_T} \right) \tag{3.30}$$



Σχήμα 3.24: Output current of 2nd Bump circuit for $I_{bias} = 5nA$, $V_c = -300mV$ and $V_m = 0mV$.

$$I_{nmos} = I_{o_n} e^{\kappa_n (V_G - V_w)/V_T} \left(e^{(V_w - V_S)/V_T} - e^{(V_w - V_D)/V_T} \right) \tag{3.31}$$

We consider the I_{o_p} and I_{o_n} values of reference for PMOS and NMOS transistors respectively. For every transistor we consider, we use a scaling factor (m), i.e. mI_{o_p} or mI_{o_n}, to capture the relative W/L value according to Table3.3.

3.7.2 Modified Current Correlator Analysis

The modified current correlator is the same circuit block that was analysed in a previous section of the 1st proposed bump circuit so we have:

$$I_{out} = \frac{3I_1 I_2}{I_1 + 3I_2} \tag{3.32}$$

3.7.3 NMOS Bulk-Controlled Block Analysis

$M_{n1} - M_{n4}$ and M_{n6} are the NMOS bulk-controlled block's transistors. Transistors M_{n1} and M_{n2} have the same characteristics. They operate in subthreshold triode region and their bulks are connected to a voltage V_c. Their

triode currents are given by:

$$I_1 = 16I_{o_n} e^{\kappa_n (V_{S_{Mn3}} - V_c)/V_T} (e^{(V_c - V_S)/V_T} - e^{(V_c - V_{S_{Mn3}})/V_T}) \tag{3.33}$$

$$I_2 = 16I_{o_n} e^{\kappa_n (V_{S_{Mn4}} - V_c)/V_T} (e^{(V_c - V_S)/V_T} - e^{(V_c - V_{S_{Mn4}})/V_T}) \tag{3.34}$$

where $V_{S_{Mn3}}$ and $V_{S_{Mn4}}$ are the source voltages of transistors M_{n3} and M_{n4}. Transistors M_{n3} and M_{n4} operate in saturation region and their currents are given by:

$$I_1 = I_{o_n} e^{(\kappa_n V_{in} + (1-\kappa_n)V_{SS} - V_{S_{Mn3}})/V_T} \tag{3.35}$$

$$I_2 = I_{o_n} e^{(\kappa_n V_m + (1-\kappa_n)V_{SS} - V_{S_{Mn4}})/V_T} \tag{3.36}$$

In this step, we combine Eqs.3.33 and 3.35. The current I_1 is given by:

$$I_1^{(\kappa_n+1)} + I_{x1} \frac{I_1}{I_{o_n}} e^{((\kappa_n - 1)V_{SS} - \kappa_n V_{in})/V_T} = I_{x1} e^{-V_S/V_T} \tag{3.37}$$

We combine Eqs.3.34 and 3.36. The current I_2 is given by:

$$I_2^{(\kappa_n+1)} + I_{x2} \frac{I_2}{I_{o_n}} e^{((\kappa_n - 1)V_{SS} - \kappa_n V_m)/V_T} = I_{x2} e^{-V_S/V_T} \tag{3.38}$$

The parameter term I_{x1} which depends on the input voltage V_{in} and parameter voltage V_c ις given by the following expression:

$$I_{x1} = 16 I_{o_n}^{(\kappa_n+1)} e^{(\kappa_n^2 V_{in} + (1-\kappa_n)V_c + (\kappa_n - \kappa_n^2)V_{SS})/V_T} \tag{3.39}$$

The parameter term I_{x2} which depends on the parameter voltage V_m and parameter voltage V_c is given by the following expression:

$$I_{x2} = 16 I_{o_n}^{(\kappa_n+1)} e^{(\kappa_n^2 V_m + (1-\kappa_n)V_c + (\kappa_n - \kappa_n^2)V_{SS})/V_T} \tag{3.40}$$

With M_{n6} operating in triode, the voltage V_S in Eqs.3.37 and 3.48 is such that:

$$e^{-V_S/V_T} = \frac{-I_1 - I_2 + I_{bias}}{I_{bias} e^{V_{SS}/V_T}} \tag{3.41}$$

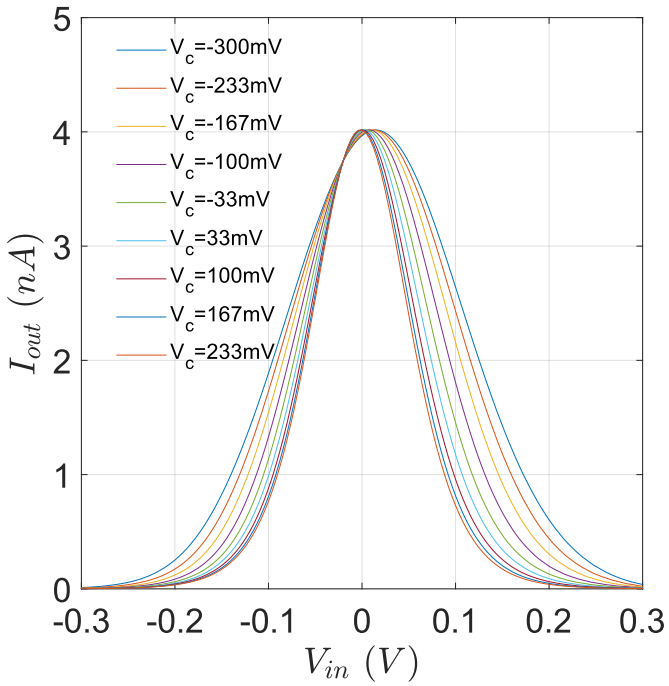
Substitution of Eq.3.41 into Eqs.3.37 and 3.38 leads to the following nonlinear system of equations with unknown I_1 and I_2:

$$\begin{aligned} I_1^{(\kappa_n+1)} + I_1 \left(\frac{I_{x1}}{I_{bias} e^{V_{SS}/V_T}} + \frac{I_{x1}}{I_{o_n}} e^{((\kappa_n - 1)V_{SS} - \kappa_n V_{in})/V_T} \right) \\ = I_{x1} \frac{I_{bias} - I_2}{I_{bias} e^{V_{SS}/V_T}} \end{aligned} \tag{3.42}$$

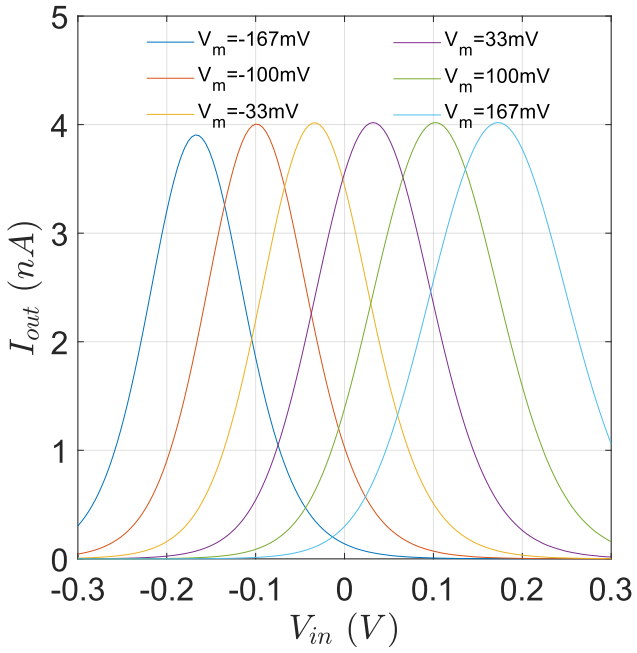
$$I_2^{(\kappa_n+1)} + I_2\left(\frac{I_{x2}}{I_{bias}e^{V_{SS}/V_T}} + \frac{I_{x2}}{I_{o_n}}e^{((\kappa_n-1)V_{SS}-\kappa_n V_m)/V_T}\right) = I_{x2}\frac{I_{bias}-I_1}{I_{bias}e^{V_{SS}/V_T}} \tag{3.43}$$

3.7.4 Theoretical Behavior of the Bump Circuit

The behavior of the proposed bump circuit is evaluated via the numerical solution of the non-linear system of Eqs.3.42 and 3.43. The solution is reached with the aid of MATLAB's fsolve function. The theoretical output current is depicted in Figs.3.25 and 3.26.. In Fig.3.25, all the circuit's parameters are constant except from parameter voltage V_c, which tunes the width of the Gaussian function. In Fig.3.26, parameter voltage V_m alters independently the center of the Gaussian curve. In both cases, the proposed circuit's theoretical analysis demonstrates its appropriate behavior and performance.



Σχήμα 3.25: Width tuning in Theoretical Output current of 2nd Bump circuit, for $I_{bias} = 5nA$ and $V_m = 0mV$.



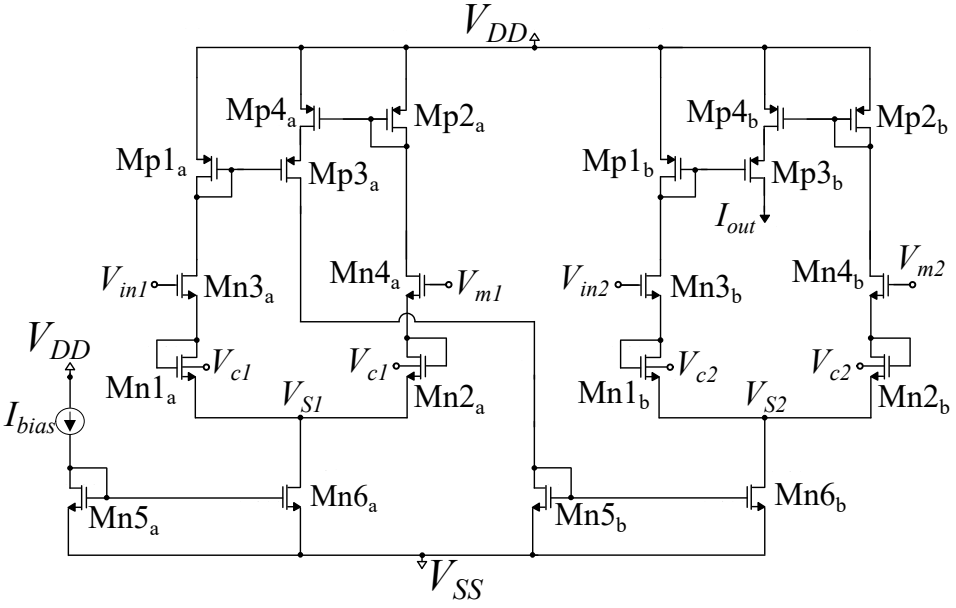
Σχήμα 3.26: Center adjustment in Theoretical Output current of 2nd Bump circuit, for $I_{bias} = 5nA$ and $V_c = -300mV$.

3.7.5 Cascaded 2-D Implemenation

The previous subsections analyze the $1-D$ bump circuit's architecture and mathematical analysis. However, the bump circuit's architecture is scalable, extending to more dimensions. Multivariate RBFs are implemented by cascaded bump circuits. These are formed by two or more bump circuits, in which the output current of each bump cell is used as the next bump cell's bias current. Different input voltage and circuit's parameters (V_m, V_c) are used in each cascaded bump circuit's stage. A $2-D$ implementation which uses two bump circuits is shown in Fig.3.27.

3.7.6 Simulations Results

The proposed ultra-low power $(4nW)$, low voltage power supply rails $(V_{DD} = -V_{SS} = 0.3V)$, fully-tunable bump circuit has been designed and evaluated in TSMC $90nm$ CMOS process, using the Cadence IC design suite. All transistors operate in the sub-threshold region and the minimum bias current is $I_{bias,min} = 3nA$. In theoretical analysis, we prove that width and center

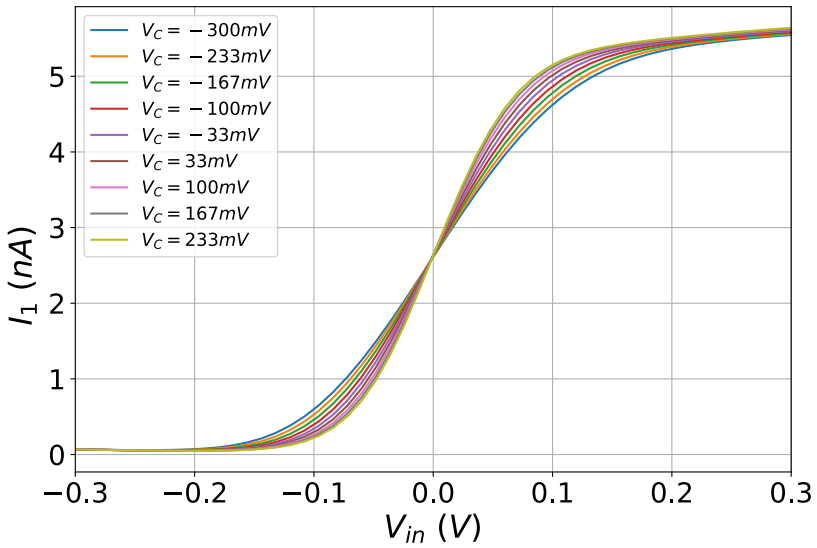


Σχήμα 3.27: $2-D$ Implementation schematic of 2nd Bump Circuit.

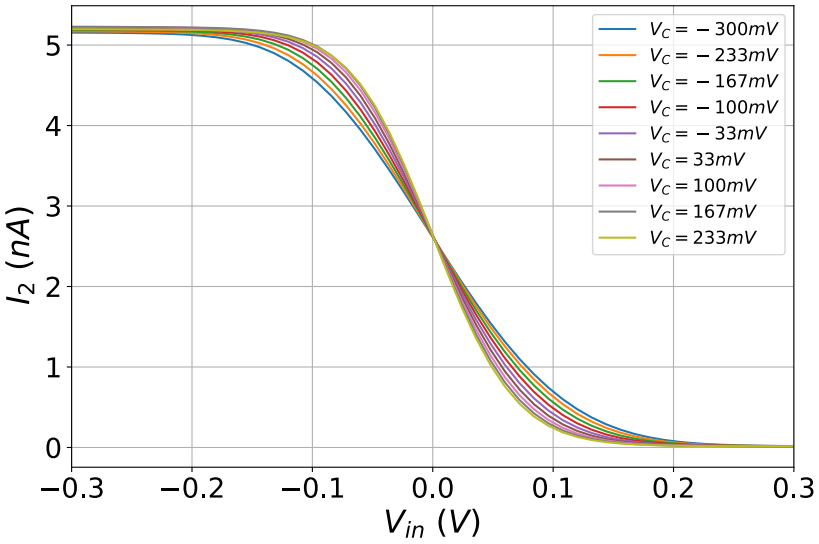
depend on parameter voltages V_c and V_m respectively. In this Section, simulations and results are presented and analyzed in order to confirm the expected theoretical behavior.

3.7.7 1-D Simulation Results

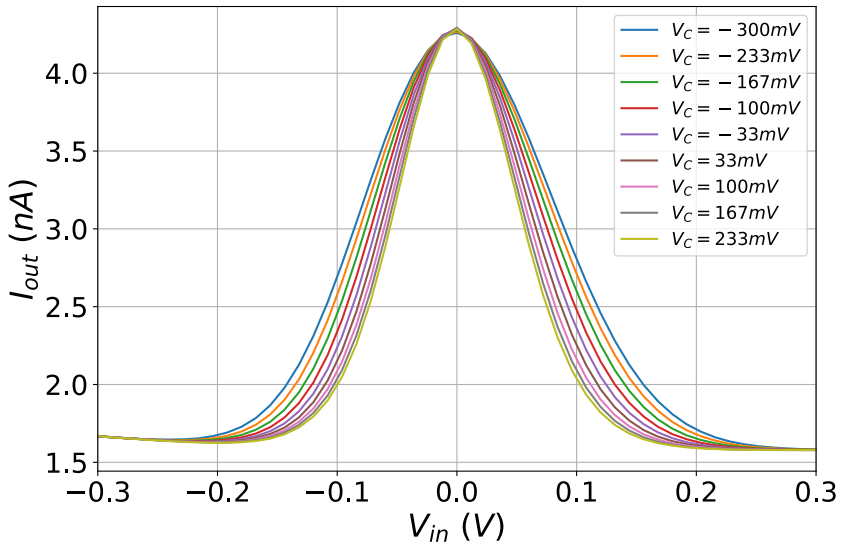
The Gaussian curve's width is controlled via parameter V_c. In Fig.3.23 the $M_{n1} - M_{n2}$ transistors' bulks (operating as diodes) are connected to the voltage V_c. The value of V_c affects the transistors' transcoductance g_m, thus altering the slope of the differential block's currents I_1 and I_2, as demonstrated in Figs.3.28 and 3.29. As a result, an independent tuning of width is achieved (without affecting the height and center) by altering the bulks' voltage V_c as shown in Fig.3.30. The scaling of the Gaussian Function's height is achieved by adjusting the bias current (I_{bias}) as shown in Fig.3.31. An increase in the bias current results in a growth in the Gaussian curve's height. Moreover, parameter voltage V_m adjusts the center of the Gaussian curve, as shown in Fig.3.32. The output current's value is maximized when the input voltage V_{in} of the bump circuit matches the parameter voltage V_m ($V_{in} = V_m$).



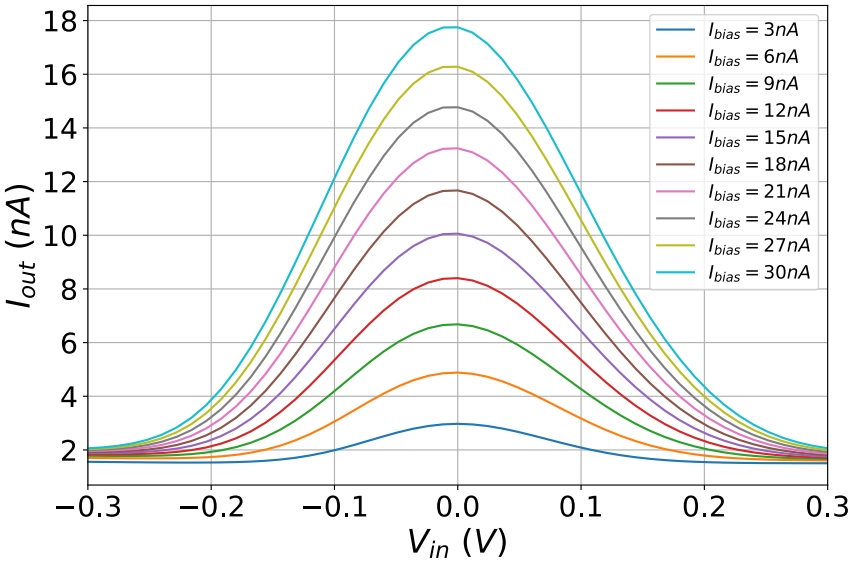
Σχήμα 3.28: Tuning of NMOS bulk-controlled block's current I_1 via parameter voltage V_c, for $I_{bias} = 5nA$ and $V_m = 0mV$.



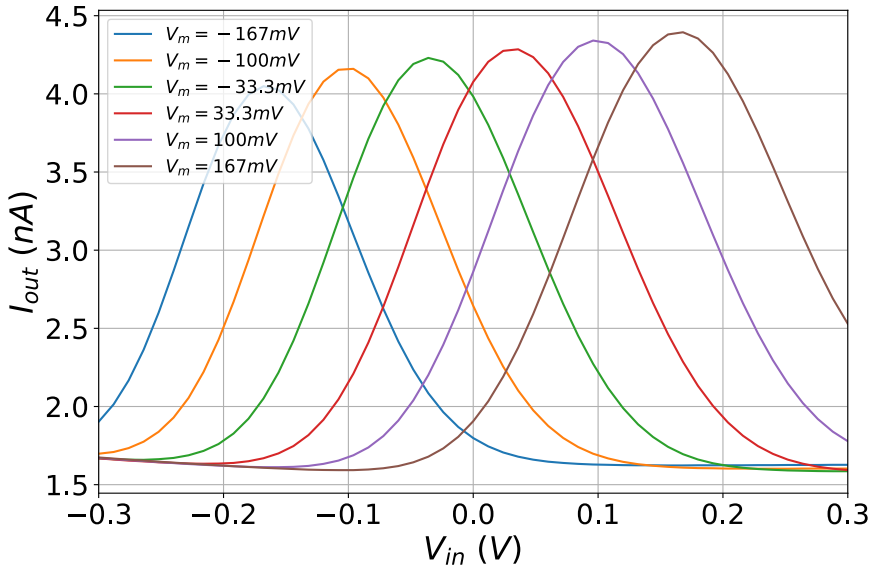
Σχήμα 3.29: Tuning of NMOS bulk-controlled block's current I_2 via parameter voltage V_c, for $I_{bias} = 5nA$ and $V_m = 0mV$.



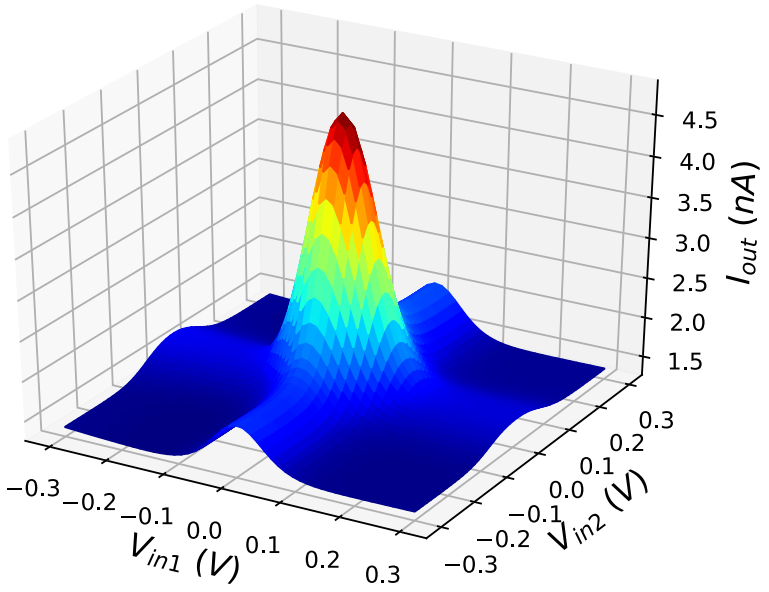
Σχήμα 3.30: Width tuning of the output current of 2nd Bump circuit with control voltage V_c, for $I_{bias} = 5nA$ and $V_m = 0mV$.



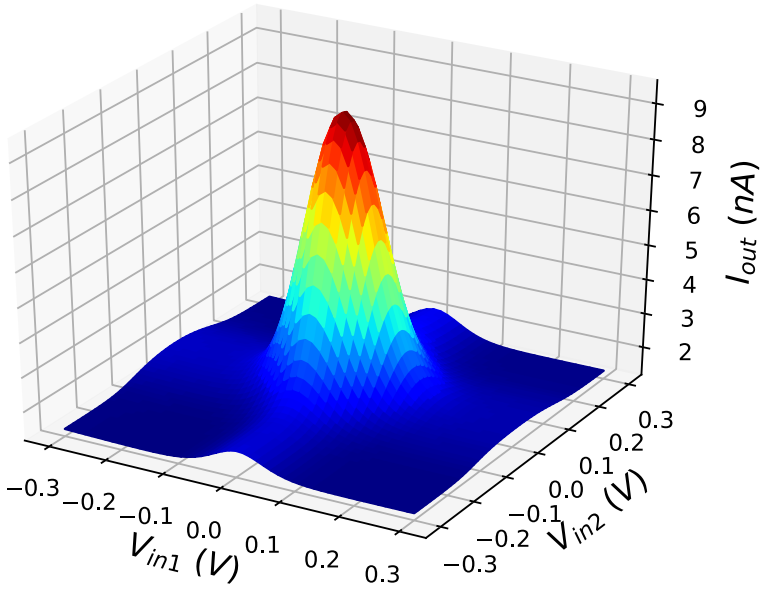
Σχήμα 3.31: Height scaling of 2nd Bump circuit with bias current I_{bias}, for $V_c = -300mV$ and $V_m = 0mV$.



Σχήμα 3.32: Center adjustment of 2nd Bump circuit with programmable voltage V_m, for $I_{bias} = 5nA$ and $V_c = -300mV$.



Σχήμα 3.33: A $2-D$ Gaussian Function of the 2nd Bump circuit with bias current $I_{bias} = 10nA$ and $V_c = 300mV$.

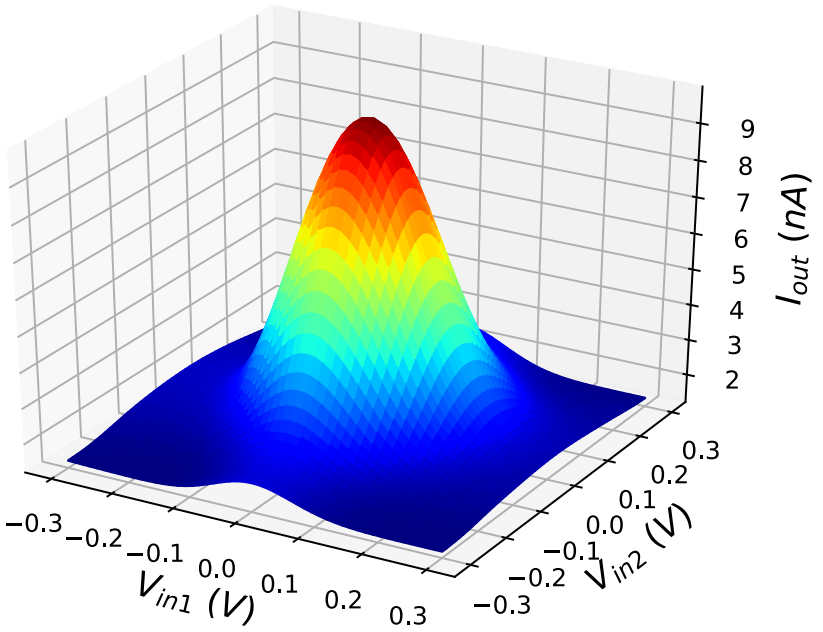


Σχήμα 3.34: A scaled height $2-D$ Gaussian Function of the 2nd Bump circuit with $I_{bias} = 30nA$ and $V_c = 300mV$.

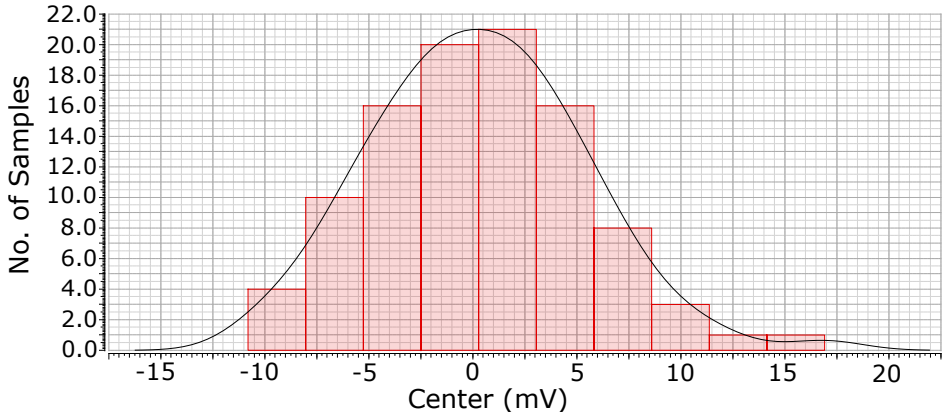
3.7.8 2-D Simulation Results

The previous figures analyze the behavior of $1-D$ bump circuit. In this step, $2-D$ Gaussian Functions are shown in three dimensional space. The output current of the cascaded bump implementation is shown in Fig.3.33. We bias the first bump circuit with $I_{bias} = 10nA$. Both Gaussian circuits' bulks are biased with $V_c = 300mV$ and parameter voltage V_m is kept constant at $0V$. In Fig.3.34, we increase the bias current's value ($I_{bias} = 30nA$), while the other parameters are kept constant. In Fig.3.35 we use the same bias current ($I_{bias} = 30nA$), but the bulks are biased with $V_c = -300mV$, which increases the Gaussian curve's width, in comparison with the results in Fig.3.34.

The sensitivity behavior has been evaluated using the Monte-Carlo analysis tool for $N = 100$ runs. The corresponding histogram for the bump circuit's center of voltage is shown in Fig.3.36. The mean value of the voltage is $V_{mean} = 2.3mV$, and the standard deviation is $\sigma_V = 5.6mV$. It confirms the correct performance and accuracy of the proposed circuit.



Σχήμα 3.35: A scaled height and width $2-D$ Gaussian Function of the 2nd Bump circuit with $I_{bias} = 30nA$ and $V_c = -300mV$.



Σχήμα 3.36: Center value sensitivity of the 2nd Bump circuit via Monte-Carlo simulation.

3.8 3rd Proposed Gaussian Function Circuit Architecture

3.8.1 Proposed Circuit Architecture

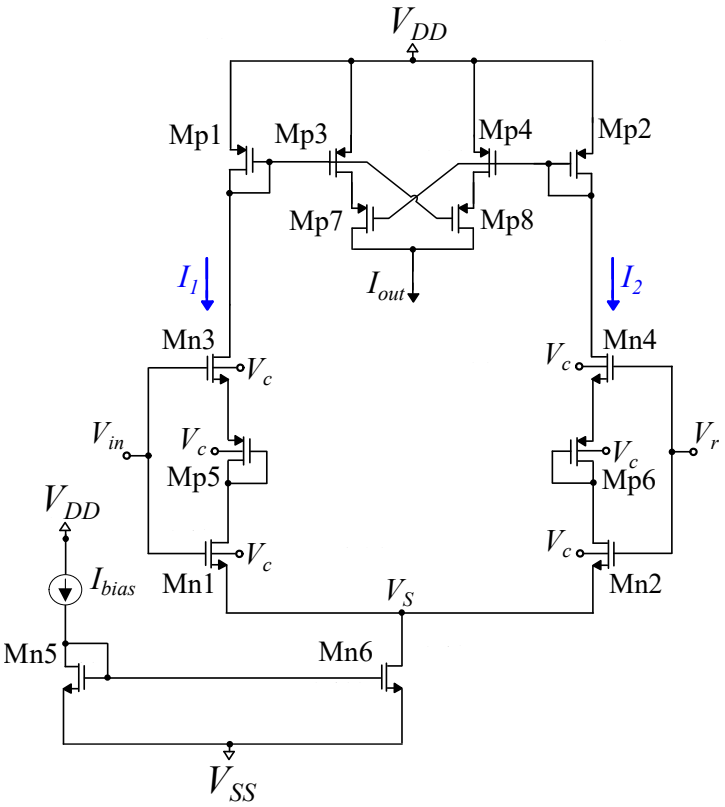
The proposed architecture is motivated by Delbruck's SB and is an ultra low power, fully tunable bulk controlled Gaussian circuit is presented. It is an ultra low power architecture because the power supply rails are $V_{DD} = -V_{SS} = 0.3V$ achieving a power consumption of only $3.3nW$. Moreover, the mean value, amplitude, and deviation of the Gaussian function are electronically programmable.

The proposed Gaussian circuit consists of two subcircuits, a modified current correlator ($M_{p1} - M_{p4}$ and $M_{p7} - M_{p8}$, which is a symmetric implementation) [61, 45] and a differential block ($M_{n1} - M_{n4}$, M_{p5}, M_{p6} and M_{n6}), which replaces the SB's differential pair, as shown in Fig.3.37. The symmetric current correlator architecture reduces the output current's offsets for input voltage V_{in} near the power supply rails. On the other hand, the differential block enables tunability in the Gaussian curve's deviation. The proposed circuit provides an output current which is a Gaussian function in relation to the input voltage V_{in}, as shown in Fig.3.39. The Gaussian curve's deviation, amplitude and mean value are controlled by three circuit's parameters (V_c, I_{bias} and V_r).

3.8.2 Simulation Results

The proposed ultra low power, fully tunable bulk controlled Gaussian circuit has been designed in TSMC $90nm$ CMOS process, using the Cadence IC design suite. All transistors operate in the sub-threshold region and the power supply rails are $V_{DD} = -V_{SS} = 0.3V$. The dimensions of the Gaussian circuit's transistors are shown in Table3.4. The Gaussian circuit's simulation results are from post-layout simulations. The layout of the proposed architecture is shown in Fig.3.38, where the area is $38.9\mu m \; \Xi \; 50.3\mu m$.

The parameter voltage V_c is connected to the bulks of the differential block's transistors ($M_{n1} - M_{n4}$, M_{p5}, M_{p6}). In contrast, the bulk voltages of the current correlator's PMOS transistors are biased in V_{DD} and the bulk voltages of the current mirror's NMOS transistors M_{n5} and M_{n6} are biased in V_{SS}. By increasing the bulk voltage V_c we achieve higher transistors' transcoductance g_m values. This results in more current flow for I_1 and I_2 as shown in Figs.3.40 and 3.41 respectively. Thus, the deviation of the Gaussian Function output is increased as shown in Fig.3.42, without affecting the

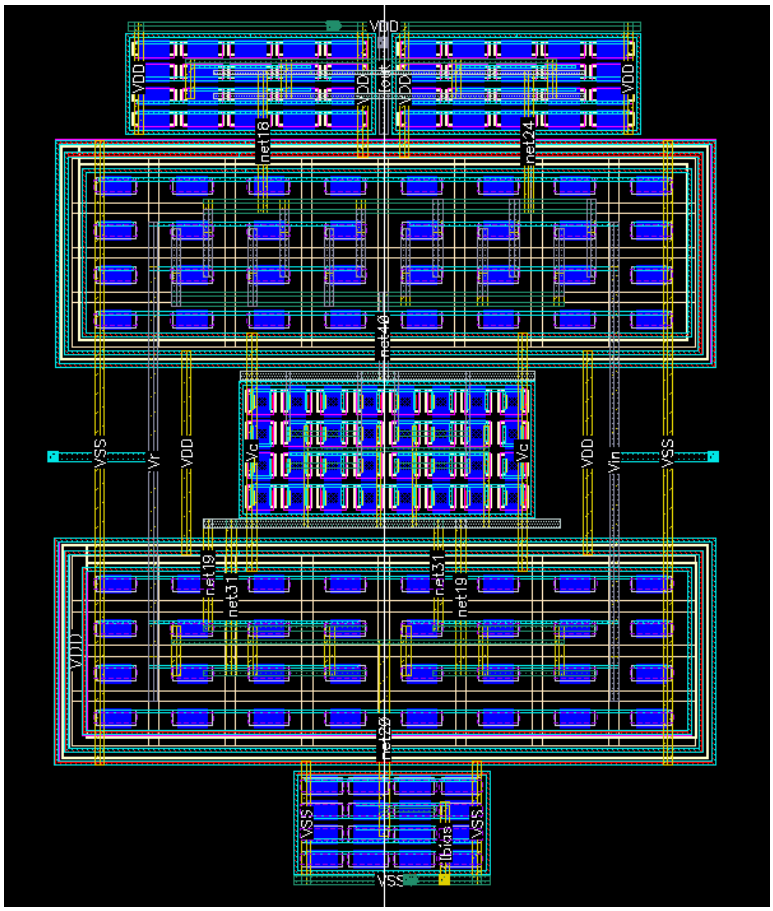


Σχήμα 3.37: 3rd Proposed Gaussian circuit.

amplitude and mean value. The bias current parameter (I_{bias}) affects the amplitude of the Gaussian curve, as shown in Fig.3.43. The increase in bias current, achieves an increase in the amplitude of the Gaussian curve. The mean value of the Gaussian curve is altered via parameter V_r, as shown in Fig.3.44. The maximum output current is achieved when the input voltage is equal to the parameter voltage ($V_{in} = V_r$).

The sensitivity behavior has been evaluated using the Monte-Carlo analysis tool for $N = 100$ points. The corresponding histograms for the bump circuit's center (mean value) and height are shown in Figs.3.45 and 3.46 respectively. The mean value of the center is $V_{mean} = 1.6mV$, and the standard deviation is $\sigma_{Vm} = 4.9mV$, while the mean value of the height is $V_{height} = 1.91nA$, and the standard deviation is $\sigma_{Vh} = 0.14nA$. Both confirm the correct performance and accuracy of the proposed circuit.

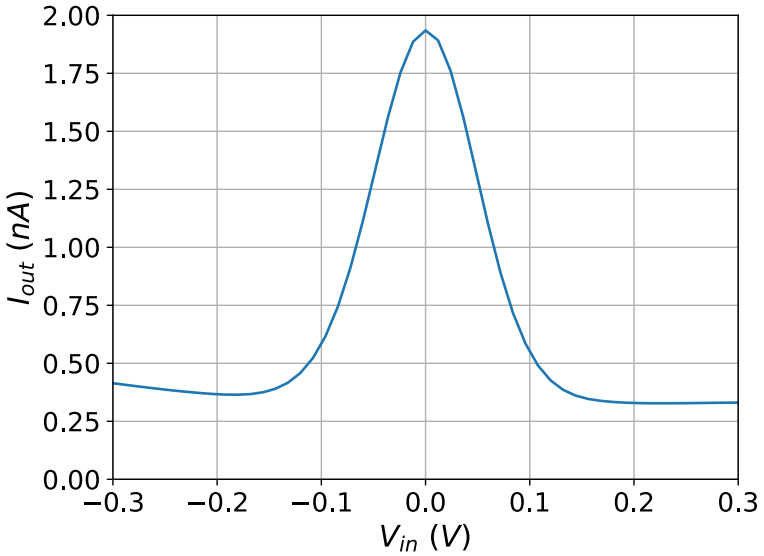
The proposed Gaussian circuit architecture demonstrates dimensional



Σχήμα 3.38: Layout of the 3rd implemented Gaussian circuit.

Πίνακας 3.4: MOS Transistors Dimensions.

Block	**W/L** (μ**m**)	**Current Correlator**	**W/L** (μ**m**)
M_{n1}-M_{n4}	1.2/1.6	M_{p1}-M_{p4}	0.4/1.6
M_{p5},M_{p6}	4.8/0.8	M_{p7},M_{p8}	0.4/1.6
M_{n5},M_{n6}	0.4/1.6	-	-

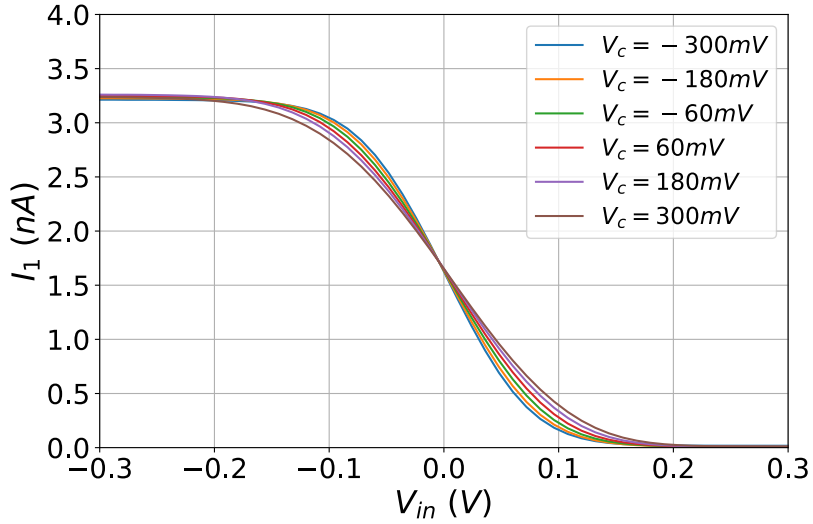


Σχήμα 3.39: Output current of the 3rd Gaussian circuit, for $I_{bias} = 3nA$, $V_c = -300mV$ and $V_r = 0mV$ (post-layout simulation).

scalability and can be used to implement multivariate Kernels. Such kernel architectures can be realized with two or more bump circuits connected in a cascaded fashion. Only the first bump circuit is biased with a current I_{bias} while the output current of each cell is used as the bias current for the next bump cell. Each Gaussian circuit cell has each own parameter voltages V_r, V_c and input voltage V_{in}. The post-layout simulation results of a $2-D$ Cascaded bump circuit implementation are shown in Figs.3.47-3.49 for different values of parameter voltages V_{c1}, V_{c2} and V_{r1}, V_{r2}. The $2-D$ implementation's simulation results confirm the desirable scalability of the proposed circuit.

3.8.3 Circuit Analysis

In this section a mathematical analysis of the ultra low power, fully tunable bulk controlled Gaussian circuit is presented. All transistors operate in the sub-threshold region and we use the MOS model in [**?**] same as in previous implementations.



Σχήμα 3.40: Tuning of current I_1 via parameter voltage V_c, for $I_{bias} = 3nA$ and $V_r = 0mV$ (post-layout simulation).

3.8.4 Symmetric Current Correlator Analysis

The symmetric current correlator consists of two classic current correlator sub-circuits ($M_{p1}, M_{p2}, M_{p3}, M_{p7}$ and $M_{p1}, M_{p2}, M_{p4}, M_{p8}$) [60] which produce two equal currents $I_{Mp7} = I_{Mp8}$. The drain current of transistor M_{p7}, which operates in saturation region, is given by the following expression:

$$I_{out} = I_{M_{p7}} = I_{o_p} e^{((\kappa_p - 1)V_{DD} - \kappa_p V_{D_{Mp2}} + V_{D_{Mp3}})/V_T} \tag{3.44}$$

where $V_{D_{Mp2}}$ and $V_{D_{Mp3}}$ are the drain voltages of transistors M_{p1} and M_{p3} respectively. Transistor's M_{p1} current is given by:

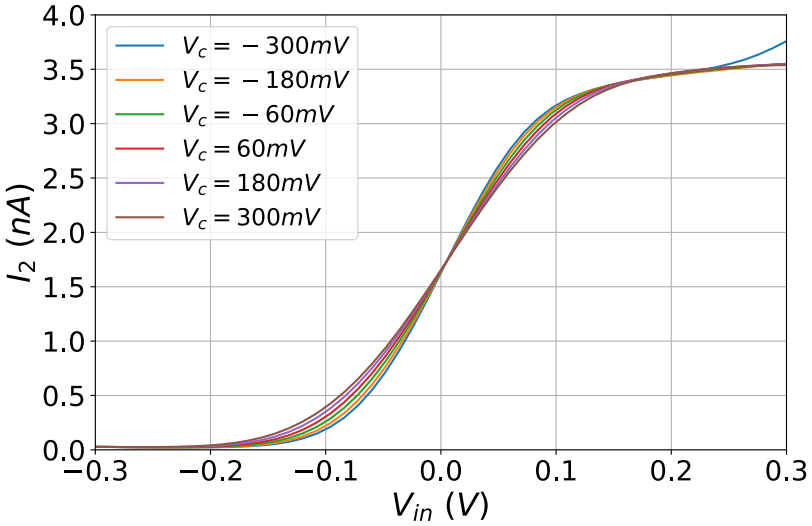
$$I_1 = I_{M_{p1}} = I_{o_p} e^{\kappa_p (V_{DD} - V_{D_{Mp1}})/V_T} \tag{3.45}$$

where $V_{D_{Mp1}}$ is the drain voltage of transistor M_{p1}. Transistor's M_{p2} current is given by:

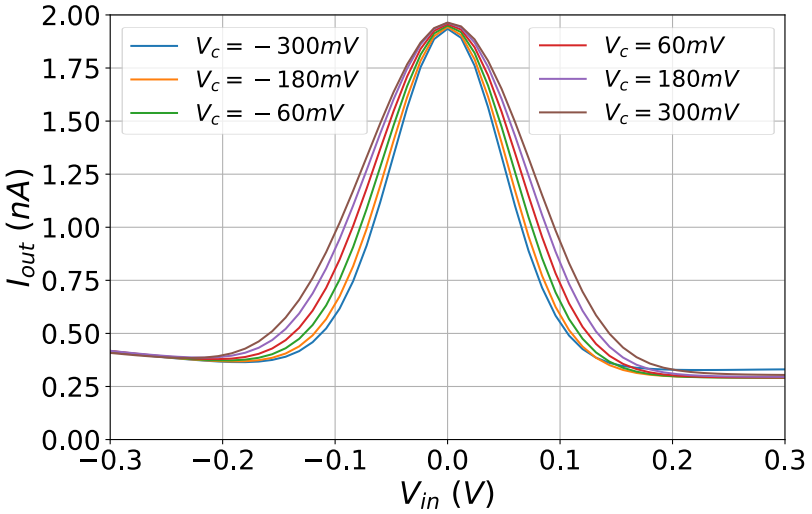
$$I_2 = I_{M_{p2}} = I_{o_p} e^{\kappa_p (V_{DD} - V_{D_{Mp2}})/V_T} \tag{3.46}$$

Both transistos M_{p3} and M_{p7} have the same current. Transistor M_{p3} operates in triode region. The drain current for transistor M_{p3} is given by:

$$I_{M_{p3}} = I_{o_p} e^{\kappa_p (V_{DD} - V_{D_{Mp1}})/V_T} \left(1 - e^{(V_{D_{Mp3}} - V_{DD})/V_T}\right) \tag{3.47}$$



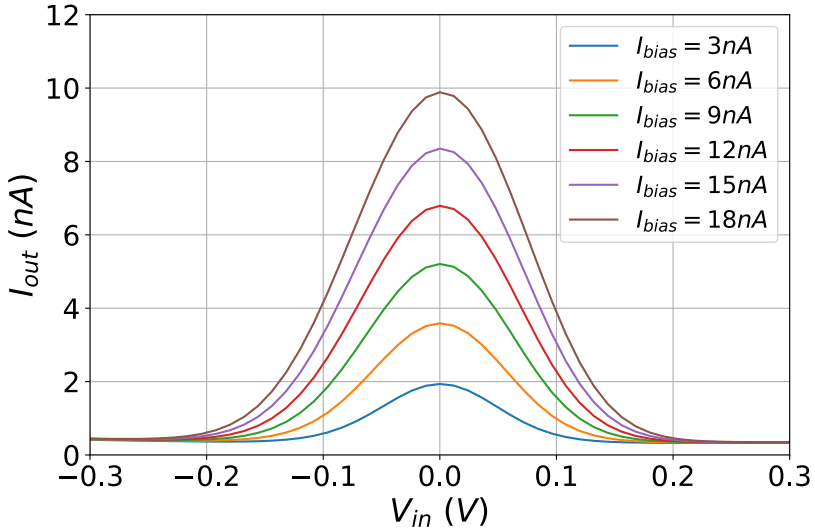
Σχήμα 3.41: Tuning of current I_2 via parameter voltage V_c, for $I_{bias} = 3nA$ and $V_r = 0mV$ (post-layout simulation).



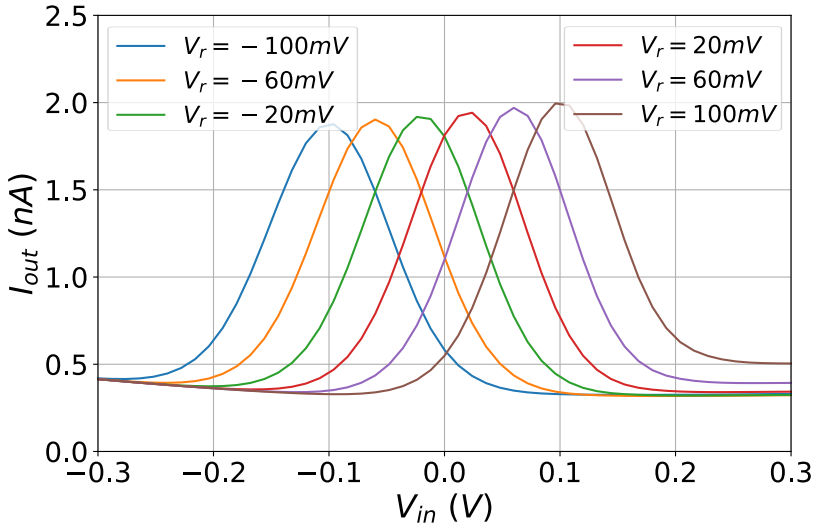
Σχήμα 3.42: Deviation tuning of the 3rd Bump circuit with control voltage V_c, for $I_{bias} = 3nA$ and $V_r = 0mV$ (post-layout simulation).

Combining Eqs.3.44-3.47 and using $I_{M_{p7}} = I_{M_{p8}}$ (symmetric current correlator implementation) we conclude:

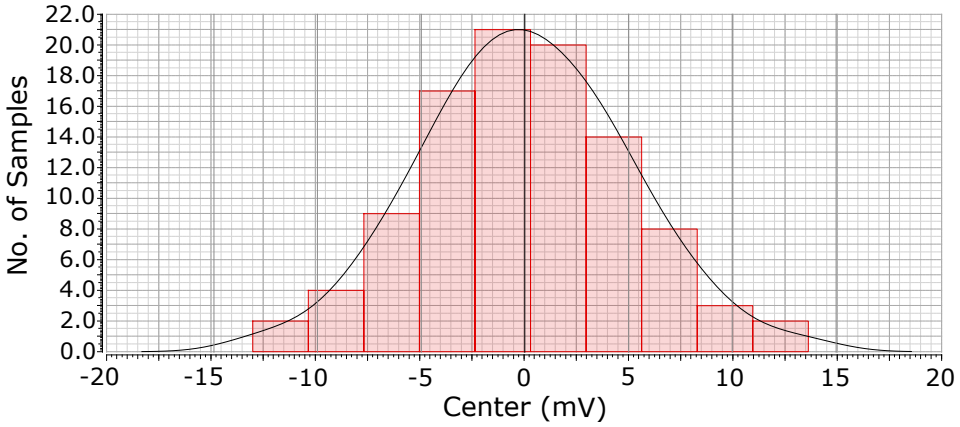
$$I_{out} = 2\frac{I_1 I_2}{I_1 + I_2} \tag{3.48}$$



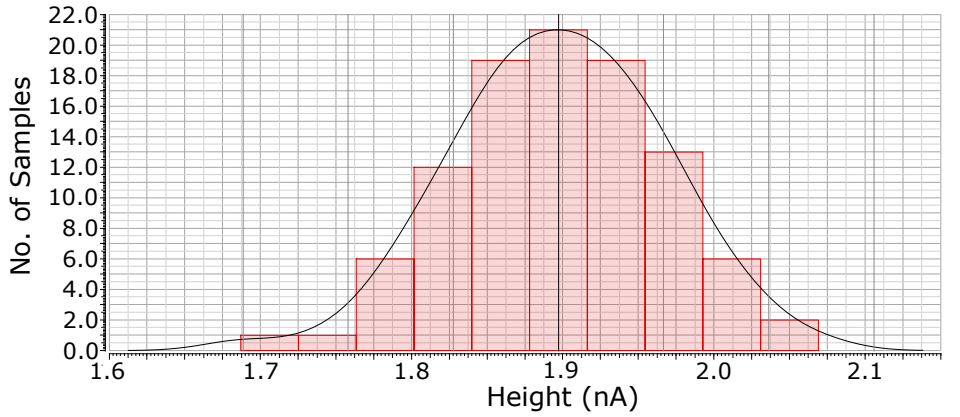
Σχήμα 3.43: Amplitude adjustment of the 3rd Bump circuit with bias current I_{bias}, for $V_c = -300mV$ and $V_r = 0mV$ (post-layout simulation).



Σχήμα 3.44: Mean value adjustment of the 3rd Bump circuit with voltage V_r, for $I_{bias} = 3nA$ and $V_c = -300mV$ (post-layout simulation).



Σχήμα 3.45: Center sensitivity of the 3rd Bump circuit via Monte-Carlo simulation (post-layout simulation).



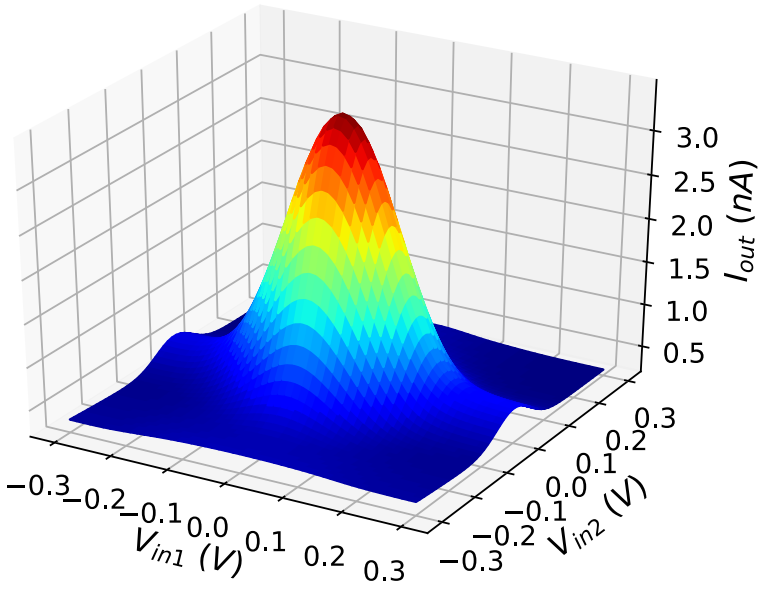
Σχήμα 3.46: Amplitude value sensitivity of the 3rd Bump circuit via Monte-Carlo simulation (post-layout simulation).

3.8.5 Differential Block Analysis

The differential block consists of seven transistors, with six of them $M_{n1} - M_{n4}$,$M_{p5} - M_{p6}$, having their bulks connected to a voltage source V_c. Transistors M_{n3} and M_{n4} operate in the saturation region and their saturation currents are given by:

$$I_1 = I_{o_n} e^{(\kappa_n V_{in} - V_{S_{Mn3}} + (1-\kappa_n)V_c)/V_T} \tag{3.49}$$

$$I_2 = I_{o_n} e^{(\kappa_n V_r - V_{S_{Mn4}} + (1-\kappa_n)V_c)/V_T} \tag{3.50}$$



Σχήμα 3.47: A $2-D$ Gaussian Function of the 3rd Bump circuit with bias current $I_{bias} = 10nA$, $V_{r1} = V_{r2} = 0V$ and $V_{c1} = V_{c2} = 300mV$ (post-layout simulation).

The weak inversion models of saturation current for transistors M_{p5} and M_{p6} are given by:

$$I_1 = 24I_{o_p} e^{(-\kappa_p V_{D_{Mn1}} + V_{S_{Mn3}} + (\kappa_p - 1)V_c)/V_T} \tag{3.51}$$

$$I_2 = 24I_{o_p} e^{(-\kappa_p V_{D_{Mn2}} + V_{S_{Mn4}} + (\kappa_p - 1)V_c)/V_T} \tag{3.52}$$

In this step we combine Eqs.3.49 and 3.51. The new expression of $V_{D_{Mn1}}$ and V_{in} is given by:

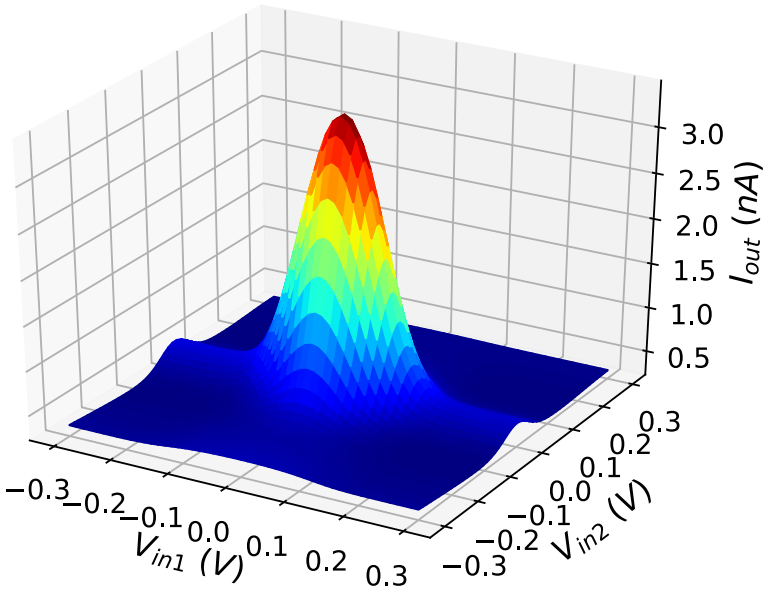
$$e^{-V_{D_{Mn1}}/V_T} = e^{\left(-\frac{\kappa_n}{\kappa_p} V_{in} + \frac{\kappa_n - \kappa_p}{\kappa_p} V_c\right)/V_T} \left(\frac{I_1^2}{24 I_o I_n}\right)^{\frac{1}{\kappa_p}} \tag{3.53}$$

Also, we combine Eqs. 3.50 and 3.52. The new expression of $V_{D_{Mn2}}$ and V_r is given by:

$$e^{-V_{D_{Mn2}}/V_T} = e^{\left(-\frac{\kappa_n}{\kappa_p} V_r + \frac{\kappa_n - \kappa_p}{\kappa_p} V_c\right)/V_T} \left(\frac{I_2^2}{24 I_o I_n}\right)^{\frac{1}{\kappa_p}} \tag{3.54}$$

Transistors M_{n1} and M_{n2} operate in the sub-threshold triode region and their currents are given by:

$$I_1 = I_{o_n} e^{\kappa_n (V_{in} - V_c)/V_T} (e^{(V_c - V_S)/V_T} - e^{(V_c - V_{D_{Mn1}})/V_T}) \tag{3.55}$$



Σχήμα 3.48: A $2-D$ Gaussian Function of the 3rd Bump circuit with bias current $I_{bias} = 10nA$, $V_{r1} = V_{r2} = 0V$ and $V_{c1} = V_{c2} = -300mV$ (post-layout simulation).

$$I_2 = I_{o_n} e^{\kappa_n (V_r - V_c)/V_T} (e^{(V_c - V_S)/V_T} - e^{(V_c - V_{D_{Mn2}})/V_T}) \tag{3.56}$$

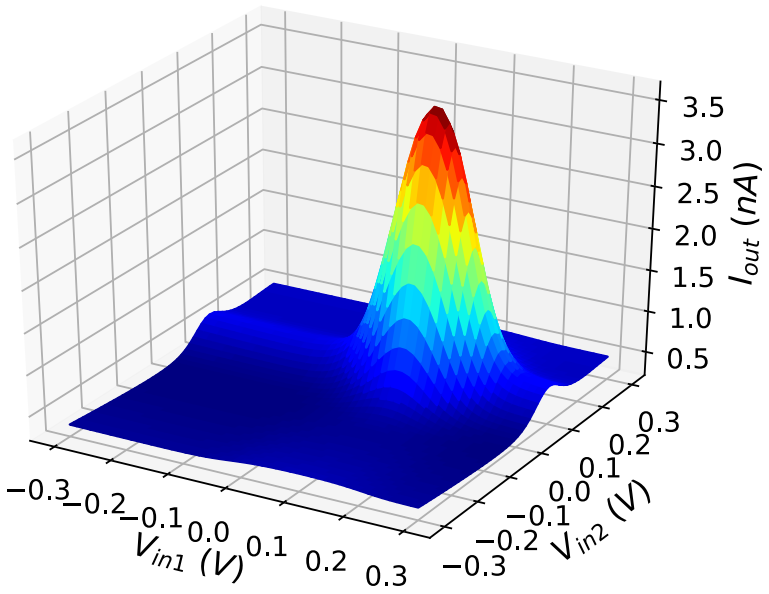
The equation of current I_1 and I_2 using the previous equations is given by:

$$\frac{I_x}{I_{o_n}} + \left(\frac{I_x^2}{24 I_{o_n} I_{o_p}}\right)^{\frac{1}{\kappa_p}} e^{(\frac{\kappa_p \kappa_n - \kappa_n}{\kappa_p}) V_x/V_T} e^{(\frac{2\kappa_p - \kappa_p \kappa_n - \kappa_n}{\kappa_p}) V_c/V_T} = e^{(\kappa_n V_x + (1-\kappa_n) V_c - V_S)/V_T} \tag{3.57}$$

where $I_x = I_1$ for $V_x = V_{in}$ and $I_x = I_2$ for $V_x = V_r$. The voltage V_S of the eqs.3.57 is given by:

$$e^{-V_S/V_T} = \frac{-I_1 - I_2 + I_{bias}}{I_{bias} e^{V_{SS}/V_T}} \tag{3.58}$$

Substitution of 3.58 to 3.57 leads to the following non-linear system



Σχήμα 3.49: A $2-D$ Gaussian Function of the 3rd Bump circuit with bias current $I_{bias} = 10nA$, $V_{r1} = V_{r2} = 100mV$ and $V_{c1} = V_{c2} = -300mV$ (post-layout simulation).

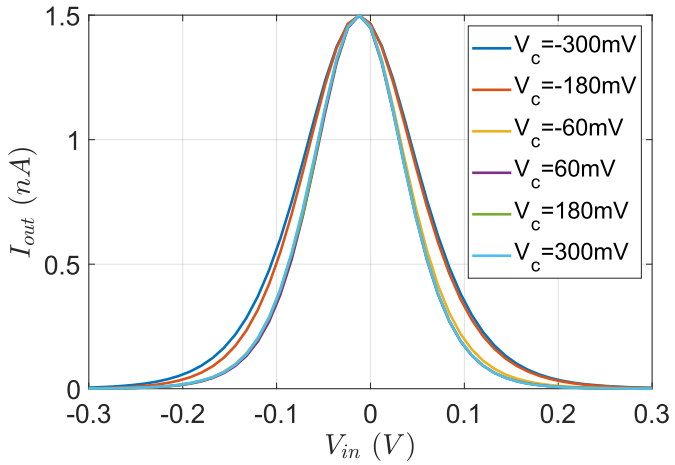
equation which involves I_1 and I_2:

$$\frac{I_1}{I_{o_n}}\left(1+\frac{I_{o_n}}{I_{bias}}e^{\frac{(1-\kappa_n)V_c+\kappa_n V_{in}-V_{SS}}{V_T}}\right) + \left(\frac{I_1^2}{24I_{o_n}I_{o_p}}\right)^{\frac{1}{\kappa_p}} e^{(\frac{\kappa_p\kappa_n-\kappa_n}{\kappa_p}V_{in}+\frac{2\kappa_p-\kappa_p\kappa_n-\kappa_n}{\kappa_p}V_c)/V_T} = \frac{I_{o_n}}{I_{bias}}\left(\frac{I_{bias}}{I_{o_n}}-\frac{I_2}{I_{o_n}}\right)e^{\frac{(1-\kappa_n)V_c+\kappa_n V_{in}-V_{SS}}{V_T}} \tag{3.59}$$

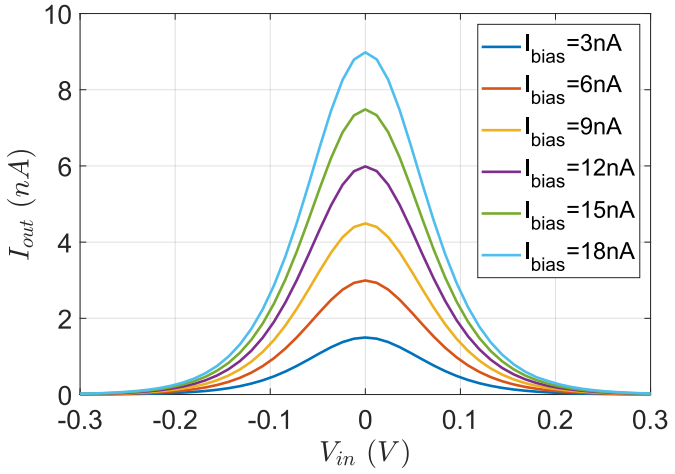
$$\frac{I_2}{I_{o_n}}\left(1+\frac{I_{o_n}}{I_{bias}}e^{\frac{(1-\kappa_n)V_c+\kappa_n V_r-V_{SS}}{V_T}}\right) + \left(\frac{I_2^2}{24I_{o_n}I_{o_p}}\right)^{\frac{1}{\kappa_p}} e^{(\frac{\kappa_p\kappa_n-\kappa_n}{\kappa_p}V_r+\frac{2\kappa_p-\kappa_p\kappa_n-\kappa_n}{\kappa_p}V_c)/V_T} = \frac{I_{o_n}}{I_{bias}}\left(\frac{I_{bias}}{I_{o_n}}-\frac{I_1}{I_{o_n}}\right)e^{\frac{(1-\kappa_n)V_c+\kappa_n V_r-V_{SS}}{V_T}} \tag{3.60}$$

Eqs.3.59 and 3.60 are numerically solved using the MATLAB's fsolve function using Eq.3.48. Their behavior is presented in Figs.3.50-3.52, which

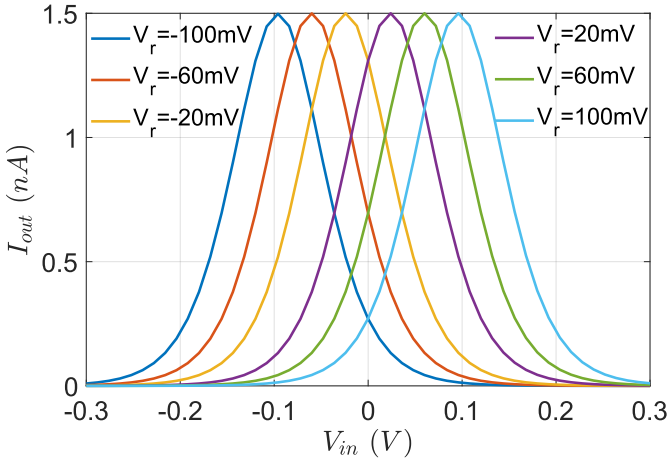
confirm the simulation results of Figs.3.42-3.44 and thus the proposed circuit's proper operation. There is only a small dc offset of $300pA$ between the theoretical and simulation results because the current correlator's output current does not reach zero for zero input currents (non-ideal circuit).



Σχήμα 3.50: Deviation tuning in Theoretical output function of the 3rd bump circuit, for $I_{bias} = 3nA$ and $V_r = 0mV$ (MATLAB simulation).



Σχήμα 3.51: Amplitude adjustment in Theoretical output function of 3rd the bump circuit, for $V_r = 0mV$ and $V_c = -300mV$ (MATLAB simulation).



Σχήμα 3.52: Mean value adjustment in Theoretical output function of the 3rd bump circuit, for $I_{bias} = 3nA$ and $V_c = -300mV$ (MATLAB simulation).

3.9 4th Proposed Gaussian Function Circuit Architecture And Analysis

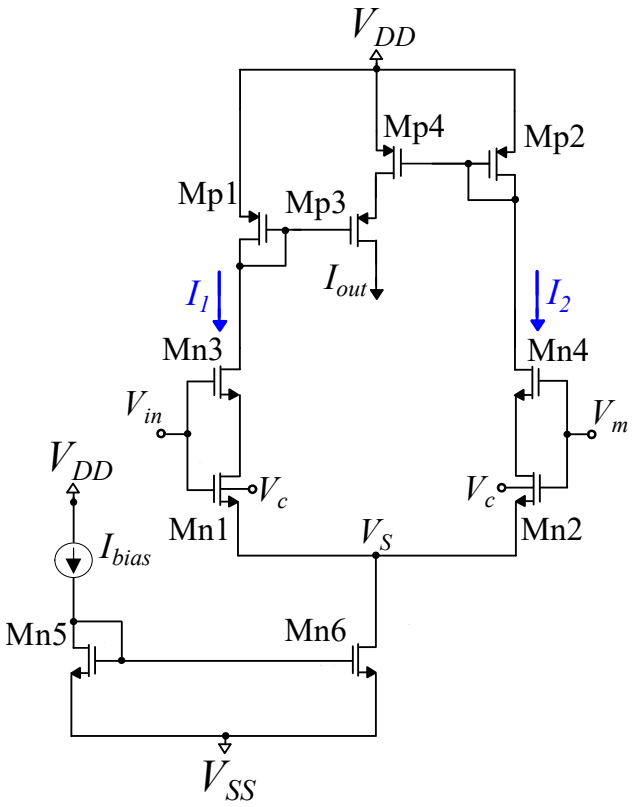
In this Section, the proposed architecture and mathematical analysis are presented. Our architecture, which is shown in Fig.3.53, consists of a modified, non symmetric current correlator ($M_{p1}-M_{p4}$ with different dimensions) and a NMOS differential block ($M_{n1}-M_{n4}$ and M_{n6}). The presented architecture provides a typical Gaussian Function, via output current, as shown in Fig.3.54, for $I_{bias} = 30nA$, $V_c = -300mV$ and $V_m = 0mV$. The Gaussian curve's height and center are tuned via bias current I_{bias} and parameter voltage V_m. In this work, we also have the ability to adjust the Gaussian curve's width, via controlled voltage V_c (parameter voltage) applied to M_{n1} and M_{n2} transistors' bulks (triple-nwell transistors), as shown in Fig.3.53.

A mathematical analysis of the proposed bump circuit is presented below. All transistors operate in the sub-threshold region and we use the MOS model in [**?**] same as previous implementations.

3.9.1 NMOS Differential Block Analysis

The NMOS differential block consists of five transistors $M_{n1}-M_{n4}$ and M_{n6}. Transistors M_{n3} and M_{n4} operate in saturation region and their drain currents are given by:

$$I_1 = 20I_{o_n}e^{(\kappa_n V_{in} - V_{S_{Mn3}} + (1-\kappa_n)V_{SS})/V_T} \tag{3.61}$$



Σχήμα 3.53: 4th Proposed Bump circuit.

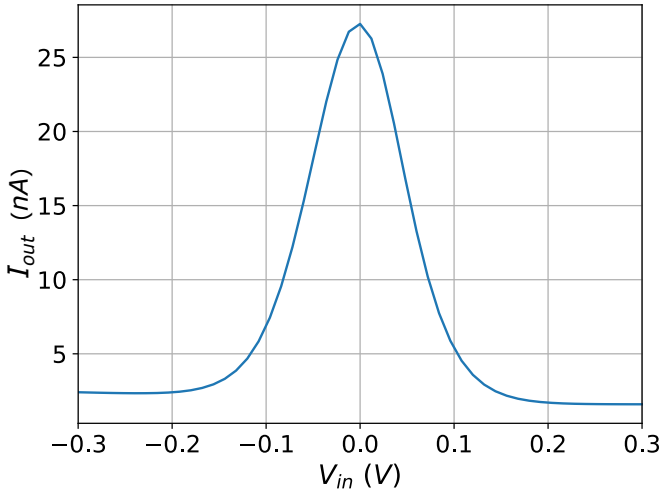
Πίνακας 3.5: MOS Transistors Dimensions.

NMOS Differential Block	**W/L** (μ**m**)	**Current Correlator**	**W/L** (μ**m**)
M_{n1},M_{n2}	2.0/1.0	M_{p1},M_{p3}	0.8/0.1
M_{n3},M_{n4}	2.0/0.1	M_{p2}	0.8/0.4
M_{n5},M_{n6}	0.2/1.6	M_{p4}	2.4/0.4

$$I_2 = 20 I_{o_n} e^{(\kappa_n V_m - V_{S_{Mn4}} + (1-\kappa_n) V_{SS})/V_T} \tag{3.62}$$

Transistors M_{n1} and M_{n2} operate in triode region and their currents are:

$$I_1 = 2 I_{o_n} e^{\kappa_n (V_{in} - V_c)/V_T} \left(e^{(V_c - V_S)/V_T} - e^{(V_c - V_{S_{Mn3}})/V_T} \right) \tag{3.63}$$



Σχήμα 3.54: Output current of the 4th Bump circuit (Gaussian Function) for $I_{bias} = 30nA$, $V_c = -300mV$ and $V_m = 0mV$.

$$I_2 = 2I_{o_n} e^{\kappa_n (V_m - V_c)/V_T} \left(e^{(V_c - V_S)/V_T} - e^{(V_c - V_{S_{Mn4}})/V_T} \right) \tag{3.64}$$

In this step we combine Eqs.3.61 and 3.63 . The new expression of I_1 is given by:

$$I_1 = 2I_{o_n} e^{(\kappa_n V_{in} - V_S + (1-\kappa_n)V_c)/V_T} - \frac{I_1}{10} e^{(\kappa_n - 1)(V_{SS} - V_c)/V_T} \tag{3.65}$$

Also, we combine Eqs.3.62 and 3.64. The new expression of I_2 is given by:

$$I_1 = 2I_{o_n} e^{(\kappa_n V_{in} - V_S + (1-\kappa_n)V_c)/V_T} - \frac{I_1}{10} e^{(\kappa_n - 1)(V_{SS} - V_c)/V_T} \tag{3.66}$$

The voltage V_S of the eqs.3.65 and 3.66 is given by:

$$e^{-V_S/V_T} = \frac{-I_1 - I_2 + I_{bias}}{I_{bias} e^{V_{SS}/V_T}} \tag{3.67}$$

Combining Eqs.3.65 and 3.67, the expression of I_1 is given by:

$$\begin{aligned} & I_1 e^{(\kappa_n - 1)V_c)/V_T} + \frac{I_1}{10} e^{(\kappa_n - 1)V_{SS}/V_T} \\ & = 2I_{o_n} \frac{I_{bias} - I_1 - I_2}{I_{bias}} \; e^{(\kappa_n V_{in} - V_{SS})/V_T} \end{aligned} \tag{3.68}$$

Combining Eqs.3.66 and 3.67 the expression of I_2 is given by:

$$\begin{aligned} & I_2 e^{(\kappa_n - 1)V_c)/V_T} + \frac{I_2}{10} e^{(\kappa_n - 1)V_{SS}/V_T} \\ & = 2I_{o_n} \frac{I_{bias} - I_1 - I_2}{I_{bias}} \; e^{(\kappa_n V_m - V_{SS})/V_T} \end{aligned} \tag{3.69}$$

We simplify the equations of current I_1 and I_2:

$$I_1 = \frac{I_{y1}}{I_{x1}}(I_{bias} - I_2) \tag{3.70}$$

$$I_2 = \frac{I_{y2}}{I_{x2}}(I_{bias} - I_1) \tag{3.71}$$

The eqs. of I_{x1}, I_{y1} and I_{x2}, I_{y2} are given by:

$$I_y = \frac{2I_{on}}{I_{bias}} e^{(\kappa_n V_x + (1-\kappa_n)V_c - V_{SS})/V_T} \tag{3.72}$$

$$I_x = 1 + \frac{1}{10} e^{(1-\kappa_n)(V_c - V_{SS})/V_T} + I_y \tag{3.73}$$

where $I_x = I_{x1}$ and $I_y = I_{y1}$ for $V_x = V_{in}$ and $I_x = I_{x2}$ and $I_y = I_{y2}$ for $V_x = V_m$. Combining Eqs.3.70 and 3.71, the final eqs. are:

$$I_1 = \frac{I_{y1} I_{bias}(I_{x2} - I_{y2})}{I_{x1}I_{x2} - I_{y1}I_{y2}} \tag{3.74}$$

$$I_2 = \frac{I_{y2} I_{bias}(I_{x1} - I_{y1})}{I_{x1}I_{x2} - I_{y1}I_{y2}} \tag{3.75}$$

3.9.2 Modified Current Correlator Analysis

The modified current correlator is the same circuit block that was analysed in a previous section of the 1st proposed bump circuit so we have:

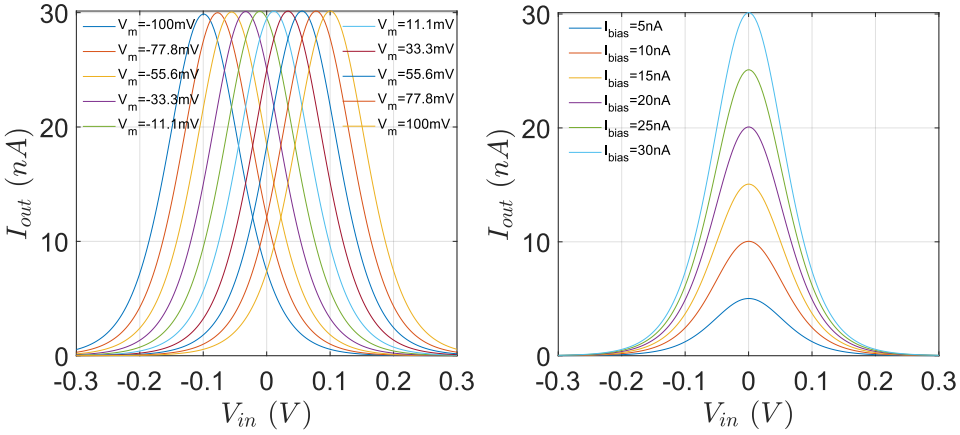
$$I_{out} = \frac{3I_1 I_2}{I_1 + 3I_2} \tag{3.76}$$

3.9.3 Proposed Bump Circuit Analysis

Substitution of Eqs.3.74 and 3.75 to 3.76 leads to the following equation which involves V_{in}, V_m, I_{bias} and V_c:

$$I_{out} = \frac{3I_{y1}I_{y2}I_{bias}(I_{x2} - I_{y2})(I_{x1} - I_{y1})}{(I_{x1}I_{x2} - I_{y1}I_{y2})(I_{y1}I_{x2} + 3I_{y2}I_{x1} - 4I_{y1}I_{y2})} \tag{3.77}$$

The proposed bump circuit's behavior is presented in Fig.3.55 using eq.3.77. The left subfigure shows the Gaussian function's behavior for constant I_{bias} and various V_m values, while the right subfigure shows changes caused by altering I_{bias}, with constant V_m. Both confirm the proper operation of the proposed topology.



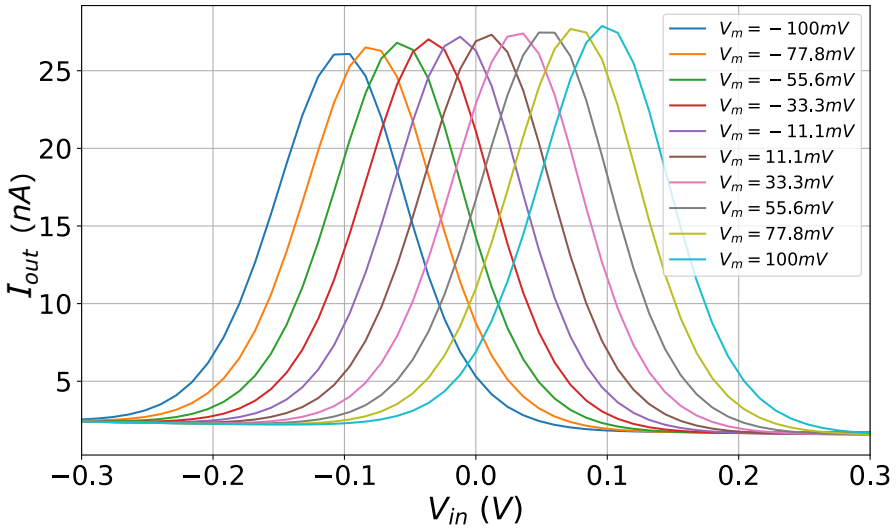
Σχήμα 3.55: Theoretical Output current of 4th Gaussian circuit

3.9.4 Simulation Results

In this work, an ultra low power, fully tunable bulk controlled Gaussian circuit is evaluated in TSMC 90nm CMOS process, usign Cadence IC design suite. It is an ultra low power architecture and the power supply rails are $V_{DD} = -V_{SS} = 0.3V$ achieving a power consumption of only 6.0nW and exhibits proper behavior with bias current down to $I_{bias} = 5nA$. Moreover, the center, height and width of the Gaussian function are electronically programmable.

The confirmation of theoretical analysis is achieved via simulation results. The output current peak is achieved when input voltage V_{in} is equal to parameter voltage V_m. As a result the Gaussian function's center is set by

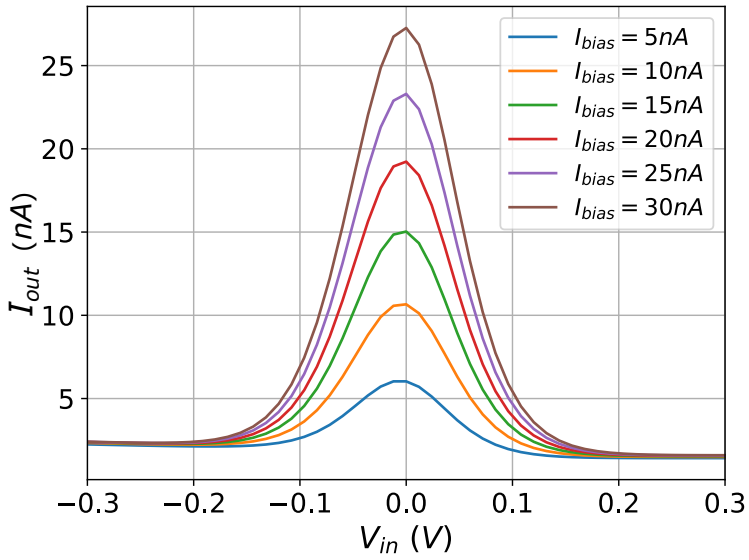
parameter voltage V_m, as shown in Fig.3.56. The Gaussian function's height is scaled through the use of the bias current I_{bias}, in a proportional way, as shown in Fig.3.57. An alteration in the parameter voltage V_c, achieves an adjustment in the output current's width. An increase in M_{n1} and M_{n2} transistors' body voltage leads to a width increase, as shown in Fig.3.58.



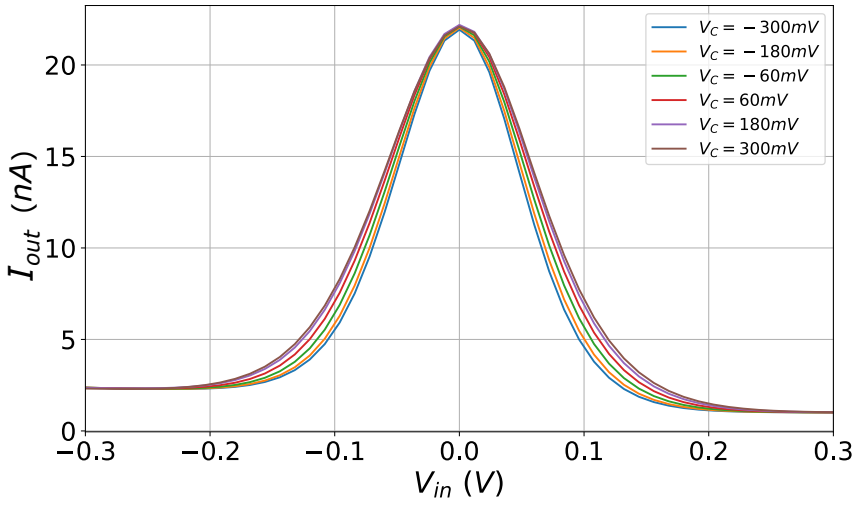
Σχήμα 3.56: Output current of 4th Bump circuit via programmable voltage V_m, for $I_{bias} = 30nA$ and $V_c = -300mV$.

The sensitivity behavior has been evaluated using the Monte-Carlo analysis tool for $N = 100$ runs. The corresponding histogram for the bump circuit's center of voltage is shown in Fig.3.59. The center of the voltage is $V_{mean} = -2.6mV$, and the standard deviation is $\sigma_V = 3.7mV$. It confirms the correct performance and accuracy of the proposed circuit.

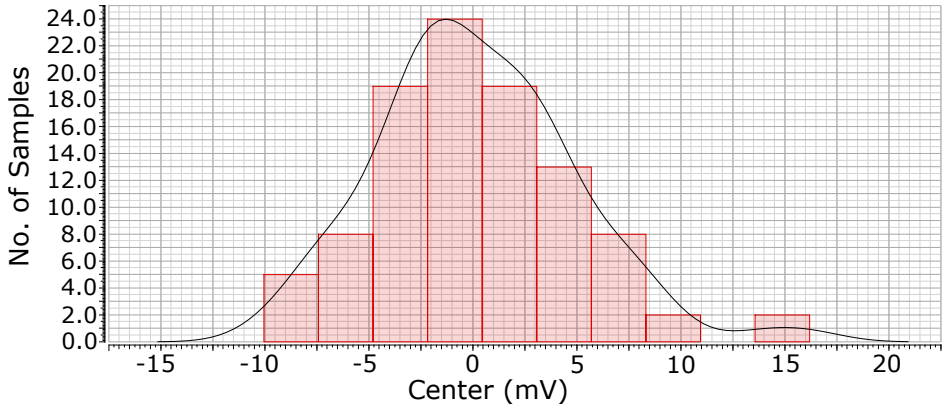
In total, 4 different novel circuit architectures for the implementation of Gaussian functions and multivariate RBF funtions were presented in detail. All 4 architectures operate with ultra low power consumption and rely on bulk-control techniques in order to achieve tunability in the Gaussian function's output width. For system level implementation, the architecture of the 1st presented bump circuit is chosen and modified so as to be used as a basic building block.



Σχήμα 3.57: Output current of 4th Bump circuit via bias current, for $V_c = -300mV$ and $V_m = 0mV$.



Σχήμα 3.58: Output current of 4th Bump circuit via programmable input voltage V_c, for $I_{bias} = 25nA$ and $V_m = 0mV$.



Σχήμα 3.59: Sensitivity performance of 4th Bump circuit using Monte-Carlo analysis.

3.10 RBF Cell

Each RBF cell in the proposed system architecture is composed of a multi-dimensional bump circuit which has its output current inserted to an analog multiplier. The multiplier's output current is the output of the RBF cell.

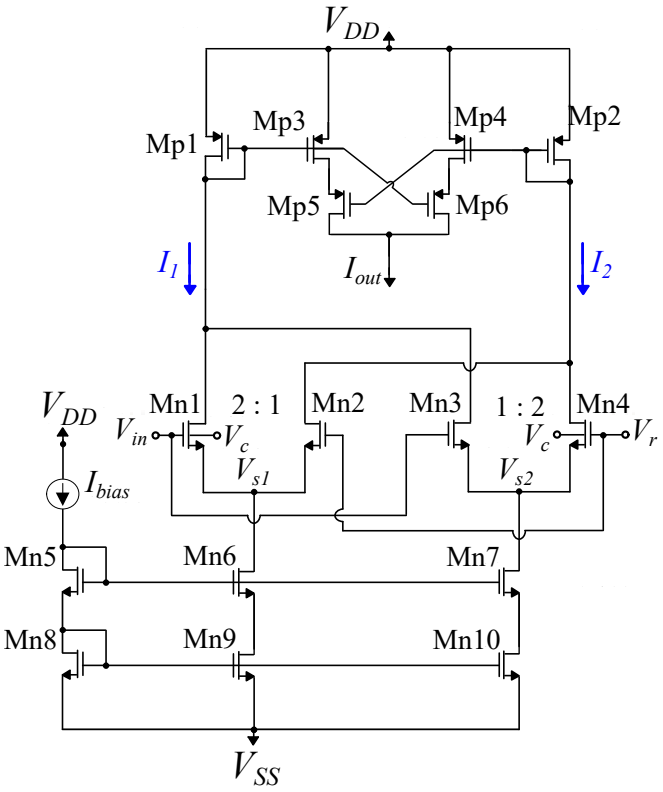
3.10.1 Modified Gaussian Function CIrcuit for System Level Implementation

The proposed Gaussian Function circuit architectures are ultra low power and compact implementations composed by a reduced number of transistors. However, in order to meet system level requirements with precision a modified version of the first proposed Gasussian Function circuit is presented in Fig.3.60. The transistor dimensions are summarized in Table 6.

Πίνακας 3.6: MOS Transistors Dimensions.

NMOS Differential Block	**W/L (μm)**	**Current Correlator**	**W/L (μm)**
M_{n1},M_{n4}	1.6/0.4	M_{p3},M_{p4}	0.4/1.6
M_{n2},M_{n3}	0.8/0.4	M_{p1},M_{p2}	1.2/1.6
M_{n5},M_{n6},M_{n7},M_{n8}	0.4/1.6		
M_{n9},M_{n10}	1.2/1.6	M_{p5},M_{p6}	0.4/1.6

Firstly,this topology instead of simple current mirrors uses cascode current mirrors for biasing of the differencial difference pair. This alteration provides accurate current mirroring even for very small currents (down to $1nA$). Furthermore, a symmetric current correlator is employed, which enchances the symmetry of the Gaussian Function output's curve. To the expense of circuit area and power consumption, a more robust circuit architecture to be used in high dimensional RBF Kernel applications is presented.

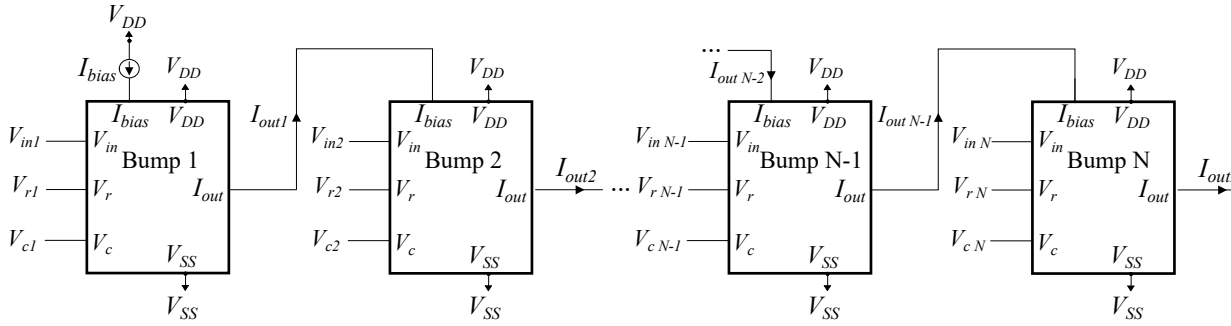


Σχήμα 3.60: Modified Gaussian Function circuit for system level implementation.

3.10.2 Multiplier Circuit

As mentioned is a previous section, cascaded bump circuits are formed by using the output current of one bump cell as bias current for the next bump cell, as shown for an N dimensional cascaded bump circuit in Fig.3.61. In this configuration, only the first bump cell is biased with a current *Ibias* which determines the peak of the mutivariate RBF Kernel. However, it

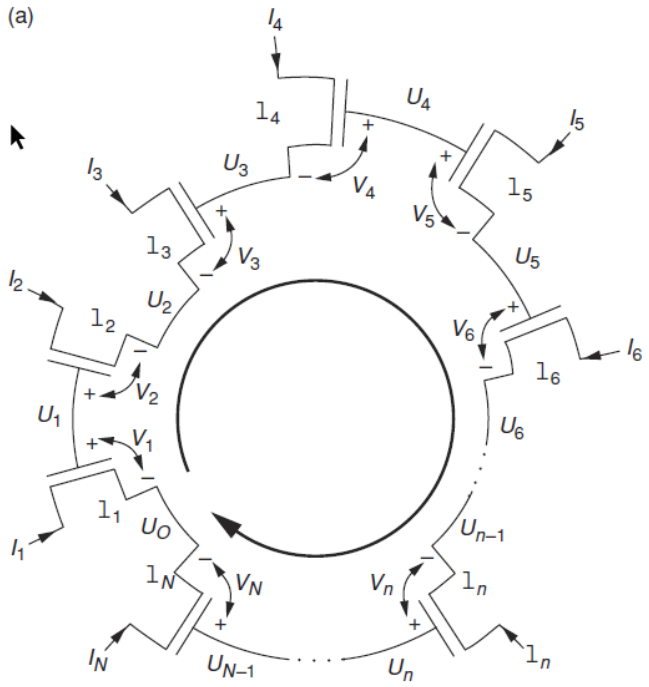
can be noticed that when the number of bump cells in such a cascaded implementation is increased in order to accomodate high dimensional data, the current scaling caused by Ibias is not entirely linear. This loss of linearity can be attributed to small inaccuracies of analog circuits which may be negligible for low dimensional inputs but as more bumps are connected in series they are accumulated and affect the output current considerably. In the SVM case in particular, the bias current of each cascaded bump circuit is the parameter that gets updated during the learning procedure so linear scaling of the RBF's output's current is of paramount importance.



Σχήμα 3.61: N-dimensional cascaded bump circuit

In order to achieve accurate linear scaling, the output current of each cascaded bump circuit is connected to an analog multiplier circuit, depicted in Fig.3.63. The multiplier consists of 5 transistors, $Mn5 - Mn9$ and is a translinear circuit. Translinear circuits operate based on the translinear principle [59]. In order to define the translinear principle, the concept of a translinear loop should be introduced. A conceptual translinear loop is depicted in Fig.3.62. It is made up of BJTs or mosfets transistors operating in subthreshold that all have their sources connected to the gates of the next transistor, forming a loop. The arrow in Fig.3.62 indicates the clockwise direction around the loop. Clockwise translinear elements are the transistors whose gate-to-source voltage is a voltage decrease in the clockwise direction of the loop while counterclockwise translinear elements are the transistors whose gate-to-source voltage is a voltage increase in the clockwise direction of the loop.

The translinear principle dictates that the product of the currents of the clockwise translinear elements in a translinear loop is equal to the product of the currents of the counterclockwise translinear elements of this loop. In essence, the translinear principle in subthreshold MOS transforms the sum of gate-to-source voltages across a translinear loop into product of currents.



Σχήμα 3.62: Conceptual translinear loop of N subthreshold MOS transistors

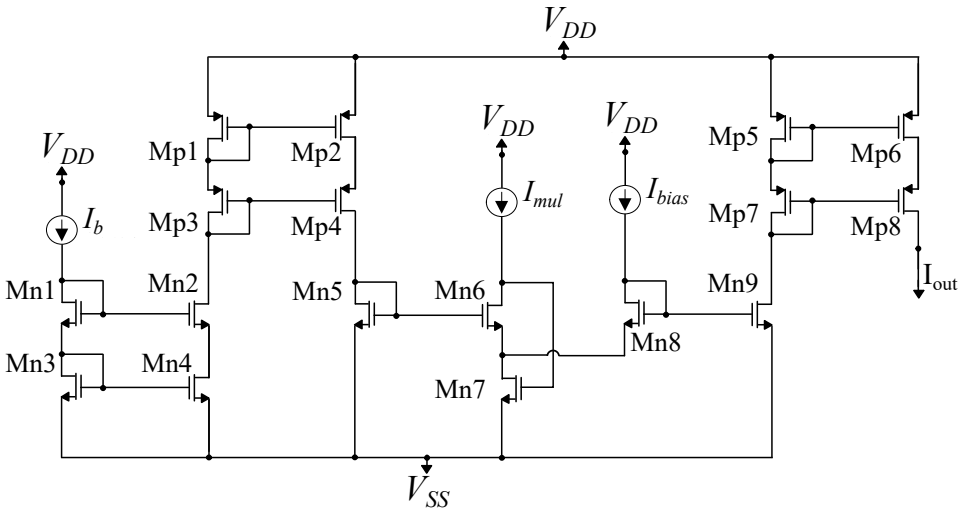
The sum of gate-to-source voltages across the loop is a result of Kirchhoff's voltage law applied around the loop and its translation to a product of currents is possible due to the exponential characteristics of the subthreshold mosfet's current with respect to its gate-to-source voltage.

In the proposed translinear multiplier circuit transistors $Mn5$, $Mn6$, $Mn8$ and $Mn9$ form a translinear loop with a so-called alternating loop topology which produces an output current independent of the subthreshold slope factor k. Supposing that all four transistors operate in subthreshold saturation region and based on the translinear principle the multiplier's output current is the following:

$$I_{out} = \frac{I_b I_{bias}}{I_{mul}} \tag{3.78}$$

where Ib is the cascaded bump circuit's output current, Ibias the multiplying term and Imul a normalizing constant current. Transistor $Mn7$ is used for proper biasing of the translinear loop. Furthermore, cascode NMOS and PMOS current mirrors have been used to achieve more precise current

mirroring.The multiplier circuit's transistor dimensions are summarized in Table 7.



Σχήμα 3.63: Analog Multiplier circuit.

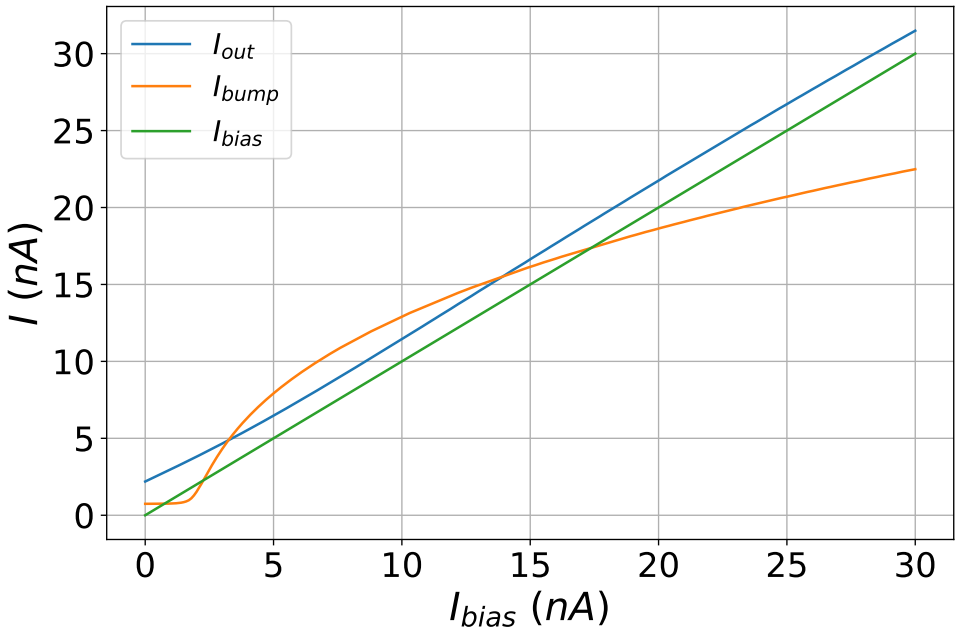
Πίνακας 3.7: Multiplier's MOS Transistors Dimensions.

Current Mirrors	**W/L** (μ**m**)	**Translinear Loop**	**W/L** (μ**m**)
M_{n1},M_{n2},M_{n3},M_{n4}	0.4/1.6	M_{n5},M_{n9}	0.4/1.6
M_{p1},M_{p2},M_{p3},M_{p4}	0.4/1.6	M_{n6}	3.6/1.6
M_{p5},M_{p6},M_{p7},M_{p8}	0.4/1.6	M_{n8}	4/1.6
M_{n7}	1.2/0.8		

The contribution of the multiplier circuit in achieving linear scaling of the RBF cell's output current is evident in Fig.3.64.

In this figure the peak of a $16-D$ RBF cell's output current is depicted, for all input and parameter voltages of all the bump circuits that comprise the cell being equal to 0.

Ibump is the output current of the $16-D$ cascaded bump circuit when its peak is scaled by the bias current of the first bump circuit of the cell. *Iout* is the peak of the output current if a multiplier is used. The desirable linearity is achieved, with the output current having only a small and constant dc offset compared to *Ibias* which is the multiplier's input current that sets the



Σχήμα 3.64: Effect of multiplier on output current.

height.

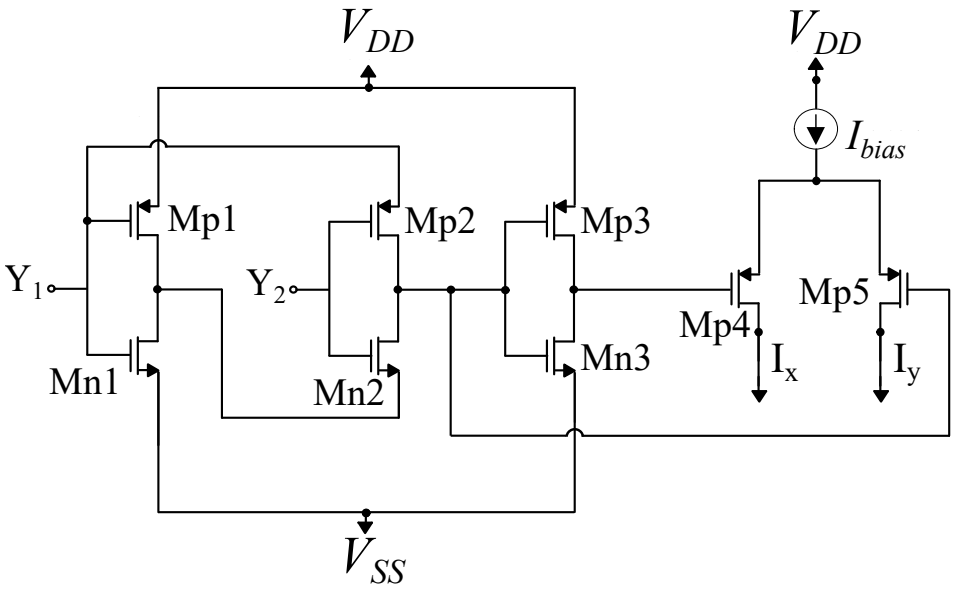
Instead of controlling the peak of the RBF output current through the bias current of the cascaded bump architecture, the cascaded bump circuit is biased with a constant bias current of $16nA$. The output current of the cascaded bump is inserted as Ib to the multiplier circuit of Fig.3.63. which is also biased with constant bias current $Imul = 16nA$. Thus,The heigt of the RBF cell's output current is determined by the multiplier's input current $Ibias$. This is the current that corresponds to the Lagrange Multipliers and realizes SVM's update rule.

3.11 Switch Cell

In the learning block, in order to satisfy the hardware-friendly SVM update rule, the product of the two learning samples'labels has to be multiplied with each Kernel. As the labels are either 1 or -1, the result of this product is either the positive or the negative Kernel's value for the specific learning samples. Thus, the output current of each RBF cell which represents the Kernel's value should be driven as a positive value I_y or as a negative value I_x.The positive value I_y corresponds to $Y1 = Y2$ while the negative value I_x

corresponds to $Y1 = -Y2$. The labels are represented with voltages, with a positive label corresponding to the positive power supply voltage and a negative label corresponding to the negative power supply voltage.

The selective driving of the RBF cell's current through either I_y or I_x is achieved via a switch circuit. The switch circuit is depicted in Fig.3.65 and essentialy implements an XOR gate. For inputs $Y1 = Y2$ RBF's current I_{bias} flows through $Mp5$ as I_y while for inputs $Y1 = -Y2$ RBF's current I_{bias} flows through $Mp4$ as I_x. This XOR gate implementation is compact area efficient as it consists of only 6 transistors, unlike the XOR gate of CMOS static logic which consists of 8 transistors. The switch cell's transistor dimensions are summarized in Table 8.



Σχήμα 3.65: Switch circuit.

Πίνακας 3.8: Switch's MOS Transistors Dimensions.

W/L (μ**m**)			
M_{n1},M_{n2},M_{n3}	0.8/0.2		
M_{p1},M_{p2},M_{p3}	0.8/0.2	M_{p4},M_{p5}	0.4/1.6

3.12 Adjuster Circuit

The hardware friendly SVM update rule of Eq. 2.10 can be transformed in the following current equation:

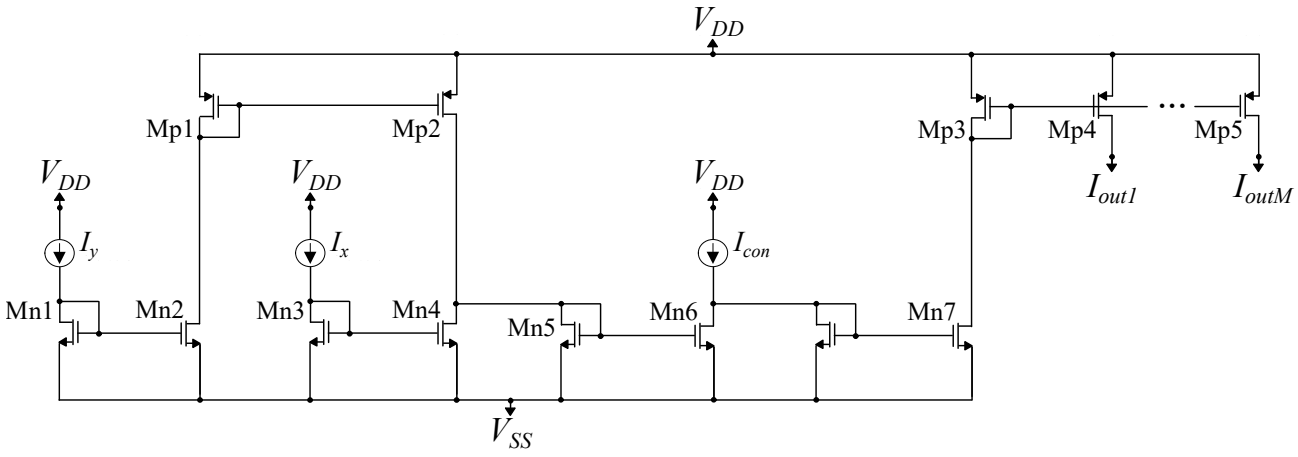
$$I_{new_i} = min(Icon, max(0, Icon - y_i \sum_{i \neq m} y_m I_m)) \tag{3.79}$$

where I_{new_i} is the updated value of the bias current of the i_{th} RBF cell and Icon is a parameter current corresponding to regularization parameter C of SVM. The adjuster is the circuit that performs these non-linear minimum and maximum operations and also performs iterations on the above mentioned equation forming a feedback loop to update the current values.

The adjuster circuit is depicted in Fig.3.66. It is a current mirror based circuit with parameter bias current $Icon = 40nA$ and the following input currents:

$I_y = \sum_{y_i = y_m} I_m$, $I_x = \sum_{y_i \neq y_k} I_k$ for the i_{th} adjuster circuit. The min and max operations are realized by the unilateral current flow in NMOS transistors $Mn6$ whose current can not be lower than zero and $Mn7$ whose current may not exceed the value of I_{con}. Fig.3.67 demonstrates the proper operation of the adjuster circuit for input currents I_y , different values of I_x and $I_{con} = 30nA$. The adjuster circuit exhibits the desirable behaviour based on the following expression:

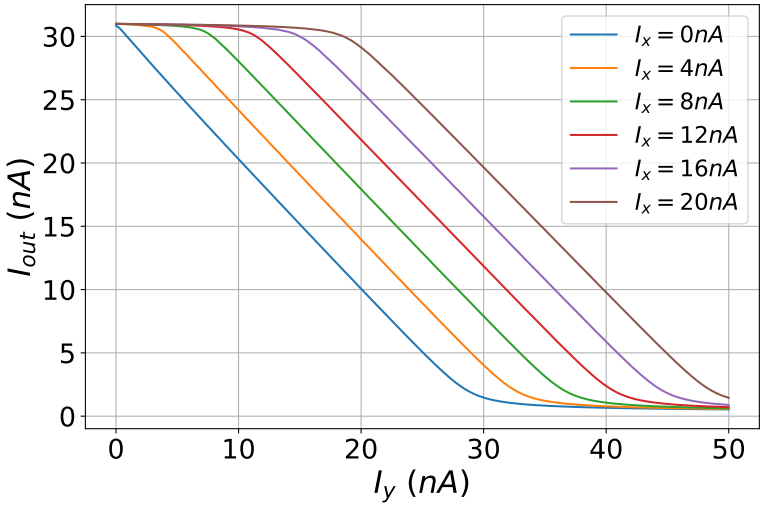
$$I_{out} = min(Icon, max(0, Icon - I_y + I_x) \tag{3.80}$$



Σχήμα 3.66: Adjuster circuit.

Πίνακας 3.9: Adjuster's MOS Transistors Dimensions.

W/L (μ**m**)	
M_{n1},M_{n2}	0.4/6.4
M_{n3},M_{n4}	0.4/6.4
M_{n5},M_{n6},M_{n7}	0.4/6.4
M_{p1},M_{p2}	0.4/6.4
M_{p3},M_{p4}	0.4/6.4
M_{p5},M_{p6},M_{p7}	0.4/6.4

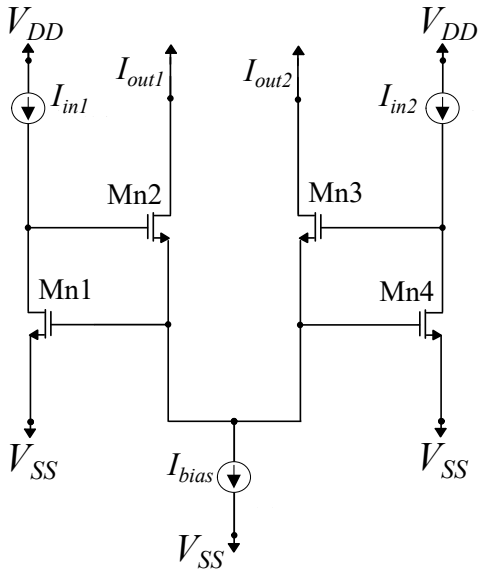


Σχήμα 3.67: Adjuster output current.

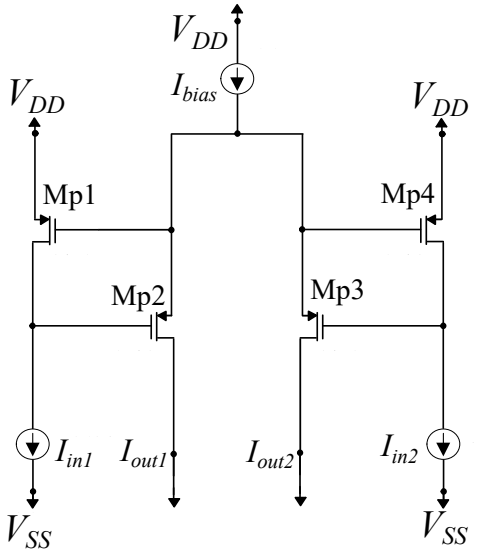
3.13 Proposed Winner-Take-All Circuit

The Winner-Take-All circuit is a circuit that receives N input signals and presents in the output the response of only the largest input signal while supressing the responses of the other $N-1$ inputs. In essence, the WTA circuit implements a max() function. There have been several voltage mode WTA circuit implementations[93, 94, 95] as well as analog current mode WTA circuits[96, 97, 98, 99] . All such current mode WTA circuit architectures are modifications of the original WTA circuit presented by Lazzaro(1989).

The circuit architecture of the simple WTA circuit for two inputs and a NMOS and a PMOS implementation are presented in Figs.3.68 and 3.69 respectively.For the NMOS case, the simple WTA circuit is composed of



Σχήμα 3.68: Simple NMOS Winner-Take-All Circuit.



Σχήμα 3.69: Simple PMOS Winner-Take-All Circuit.

4 NMOS transistors of the same W and L operating in subthreshold and a constant current source Ibias.The transistors' dimensions are summarized in

Table 3.10. For input currents $I_{in1} = I_{in2}$ then $I_{out1} = I_{out2} = 0.5I_{bias}$. Due to the fact that Mn1 and Mn4 have the same V_{GS} voltage, for input currents $I_{in1} > I_{in2}$ it follows that $V_{D_{Mn1}} = V_{G_{Mn2}} > V_{G_{Mn3}} = V_{D_{Mn4}}$. Supposing that both output transistors Mn2 and Mn3 operate in saturation and due to the fact that they both have the same source voltage a small difference in their gate voltages results in an exponentially larger difference in the output currents. In this case, $I_{out1} = I_{bias}$ and $I_{out2} = 0$. Thus, for input currents differing by a sufficient amount, only the output current corresponding to the largest input current will be nonzero.

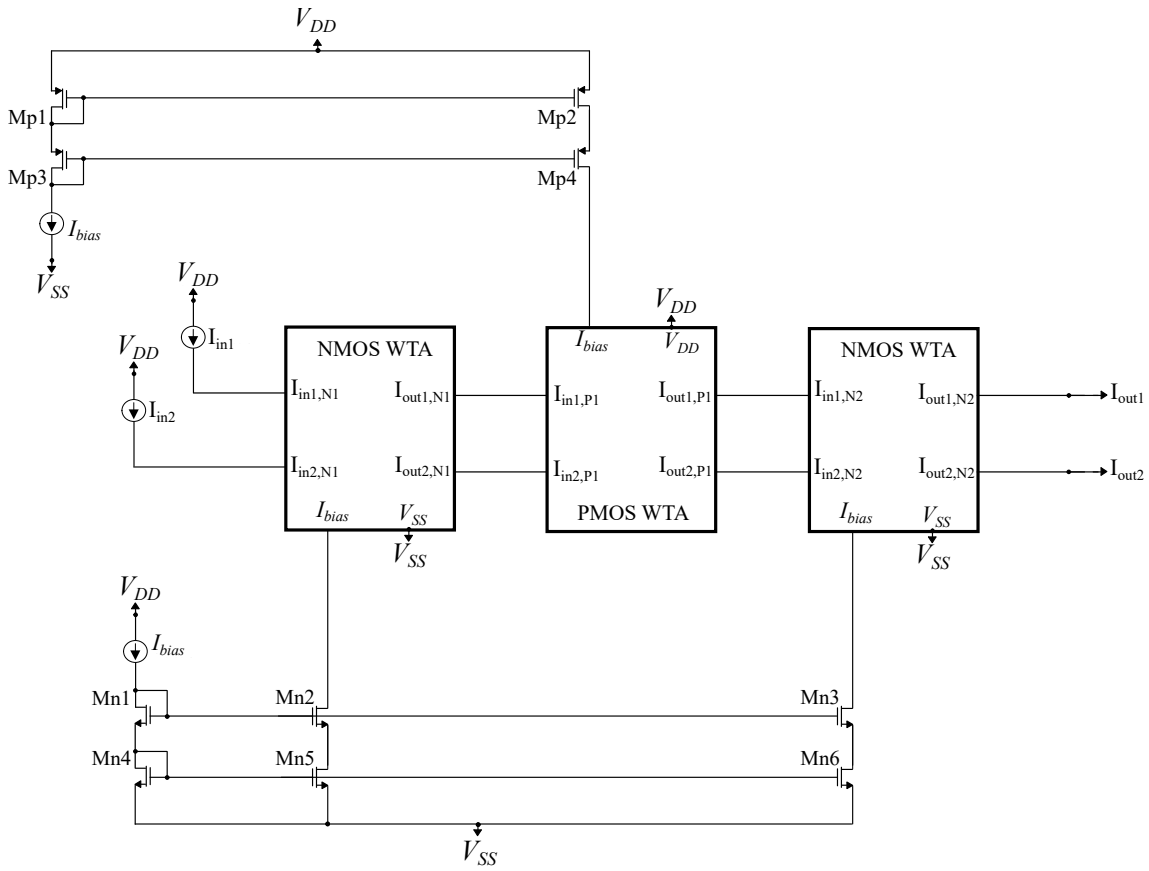
The WTA circuit can be extended to accomodate multiple inputs. In our case however two inputs are required in order for the circuit to compare the positive and the negative Kernel values and perform classification based on SVM decision rule.

In the proposed circuit architecture, instead of using a simple NMOS or PMOS WTA circuit, a triple WTA circuit is used, depicted in Fig.3.70. It consists of a NMOS,a PMOS and another NMOS WTA circuits connected in series, with the output currents of the one WTA block being the input currents to the next one. All 3 WTA blocks are biased with the same constant $Ibias = 40A$ and essentially perform the max() function 3 consecutive times.

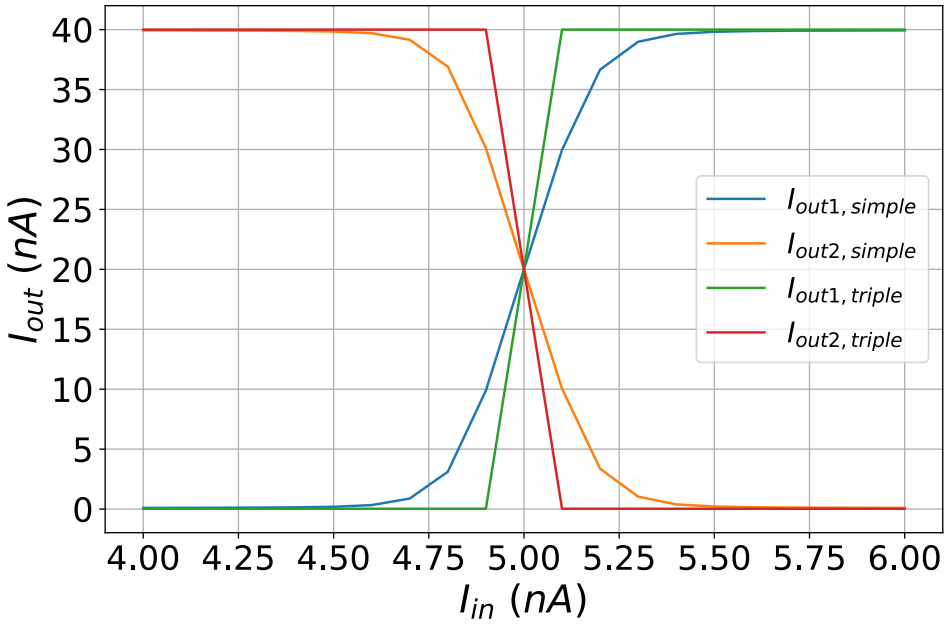
Πίνακας 3.10: WTA's MOS Transistors Dimensions.

W/L (μ**m**)	
M_{n1},M_{n2}	0.4/1.6
M_{n3},M_{n4}	0.4/1.6
M_{p1},M_{p2}	0.4/1.6
M_{p3},M_{p4}	0.4/1.6

In Fig.3.71 it can be observed that by using the triple WTA circuit as opposed to the simple architecture, the minimum current difference required by the WTA system to differentiate its inputs is cut down in half. As a result, the accuracy of the classification procedure is increased.



Σχήμα 3.70: Proposed Triple Winner-Take-All Circuit.



Σχήμα 3.71: Comparison between Simple and Proposed Triple WTA circuits.

Κεφάλαιο 4

Proof of Concept Classifier

4.1 Application Specific Classifier Architecture

The successful operation and efficiency of the proposed system architecture for SVM learning and classification is tested and verified using a real dataset[100]. The dataset is taken from Mendeley Data and is called:"Bearing Vibration Data under Time-varying Rotational Speed Conditions". The data contain vibration signals collected from bearings of different health conditions under time-varying rotational speed conditions. In order to use this data for SVM learning and classification, 2 of the total classes of the dataset are chosen for each test. After python processing of the data, 13 features are extracted.

These features are then converted to analog voltages between -300mV and 300mV which are the negative and positive supply rails of the hardware architecture respectively. Thus, the inputs of the system are analog voltage vectors of N=13 dimensions. Due to the fact that the size of the learning block's matrix of RBF cells scales proportionally to the square of learning samples, we choose to use 8 learning samples(8 vectors of 13dimensions) for the learning block. This results in 64 RBF cells, each of which consists of a cascaded 13-dimensional bump circuit and a multiplier. Also, 56 switch cells are used in the learning block as well as 8 adjuster circuits with each one producing 9 copies of its output current.

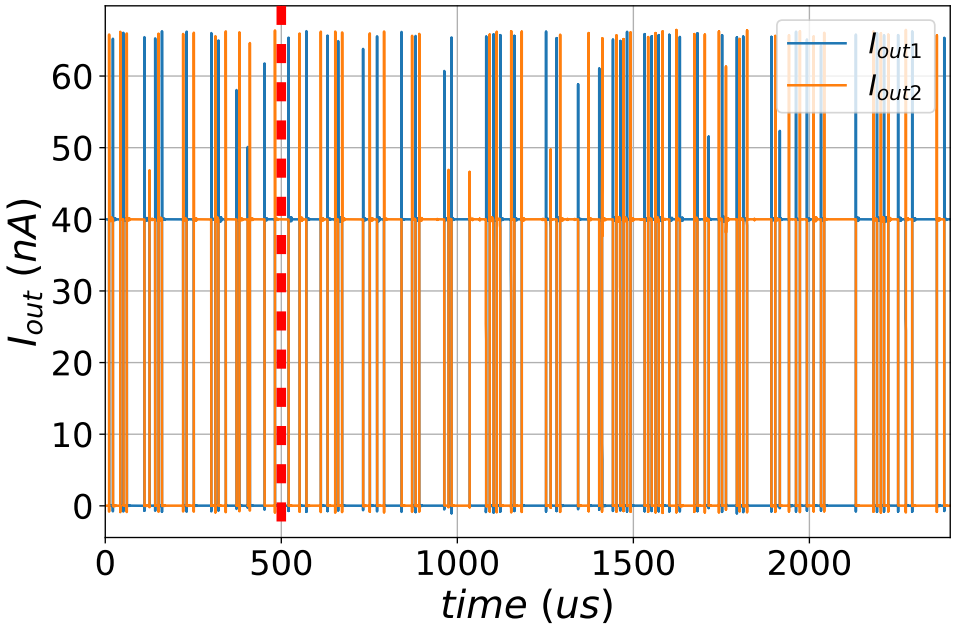
The classification block consists of 8 RBF cells, 8 switches and a two-input WTA circuit. Apart from the learning samples which are inserted to the system to commence the learning procedure, test samples(vectors of 13 dimensions) are sequentially inserted to the system in the form of analog voltages. The on-chip learning phase proceeds autonomously and the

values of the Lagrange multipliers are determined without any clock-based iterations. The classification phase is dictated by a clock with a period of 10us. During each cycle, a new testing sample is being inserted to the classification block and categorized.

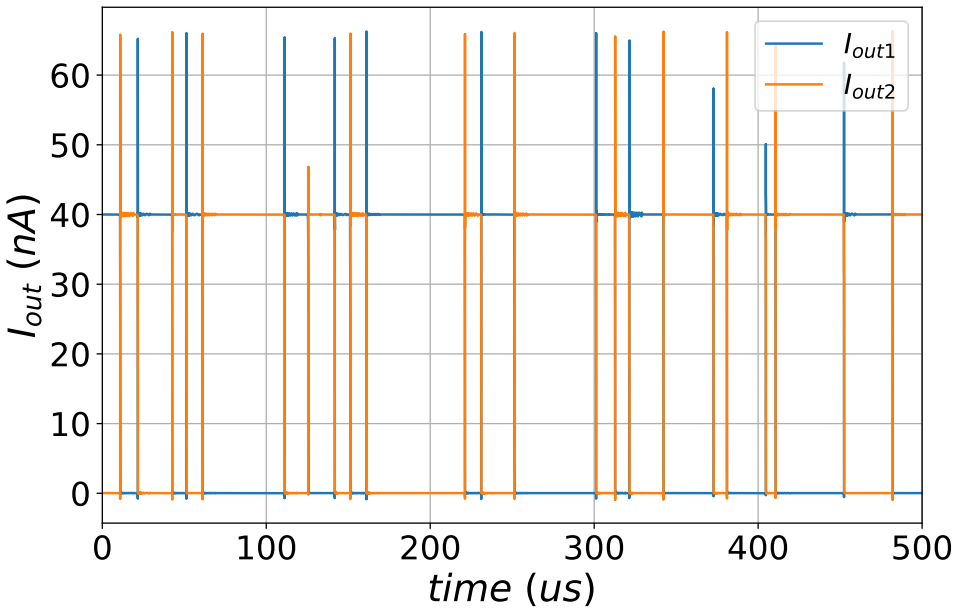
4.2 Classifier Simulation Results

In Figs.4.1-4. the output currents of the WTA circuit are depicted in the time domain. For the clock cycles that $I_{out1} = 40nA$ and $I_{out2} = 0$ then the test sample corresponding to the specific cycle is categorized in class 1. Similarly, for the clock cycles that $I_{out2} = 40nA$ and $I_{out1} = 0$ then the test sample corresponding to the specific cycle is categorized in class 2. The observed current values that exceed 40nA are transient spikes of very small duration compared to the clock cycle and do not affect the classification results. The classification results for categorization between the 1st and the 2nd class of the total dataset are depicted in Figs.4.1 and 4.2 while the classification results for categorization between the 1st and the 3rd class of the total dataset are depicted in Figs.4.3 and 4.4. In Figs. 4.1 and 4.3 the classification result for the whole dataset is presented, with 2.5ms total classification time. In these 2 Figures the vertical red dotted lines represent the end point of the time period of 500us which is magnified and depicted in Figs 4.2 and 4.4 in order to more clearly demonstrate the system's operation.

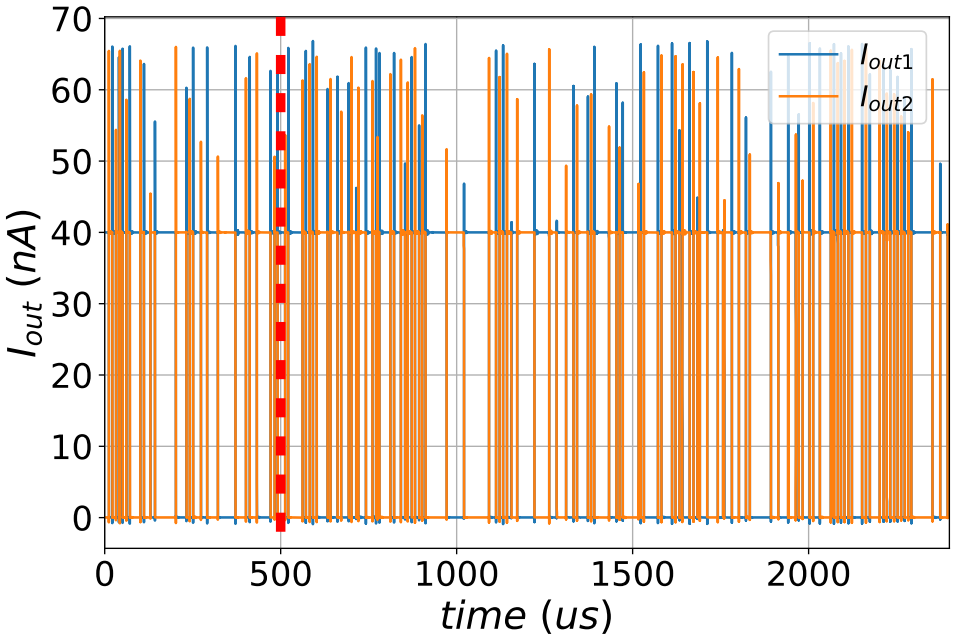
The performance of the system was tested and validated in comparison to a standard software SVM implementation with python for the same dataset and number of learning samples. For the case of classification between the first and the second class of the dataset, the simulated hardware architecture achieved a validation accuracy of 83% while Python validation accuracy was 84%. For the case of classification between the first and the third class of the dataset, the simulated hardware architecture achieved a validation accuracy of 94% while Python validation accuracy was 93%. The errors between software and the proposed hardware implementation are minimal, which proves the correct operation and the efficacy of the presented architecture.



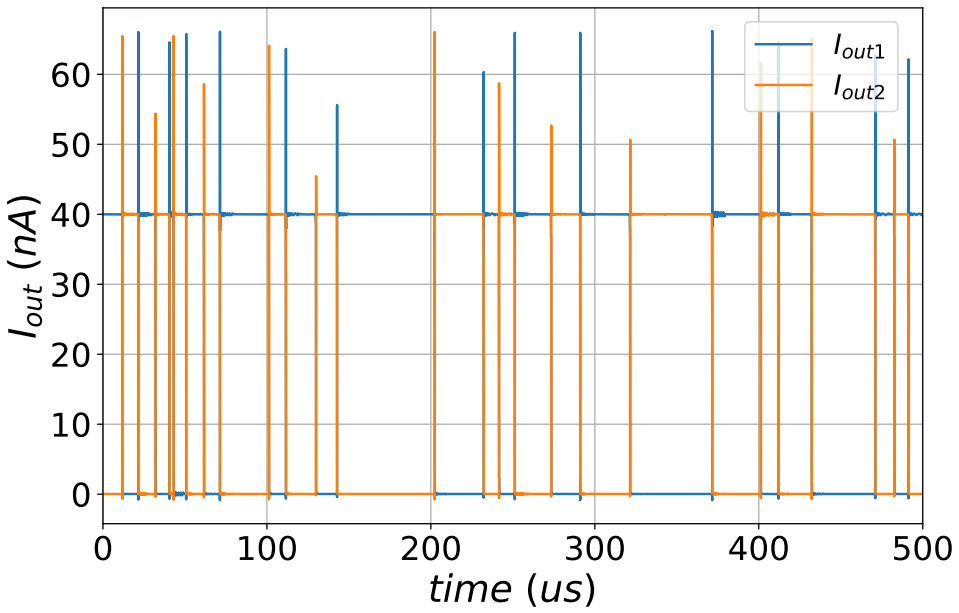
Σχήμα 4.1: SVM classification result between the 1st and the 3rd class of the dataset.



Σχήμα 4.2: SVM classification result between the 1st and the 3rd class of the dataset magnified for 500us.



Σχήμα 4.3: SVM classification result between the 1st and the 2nd class of the dataset.



Σχήμα 4.4: SVM classification result between the 1st and the 2nd class of the dataset magnified for 500us.

Κεφάλαιο 5

Conclusion and Future Work

In this work, a fully analog, ultra low power and massively parallel hardware architecture for the implementation of SVM with on-chip learning capablity was presented. The system level architecture was analysed and its basic building blocks were discussed in detail in transistor level. Four different novel circuit architectures were presented for the analog hardware realization of RBF Kernels,as well as multipliers, switches, adjusters and WTA circuits. The proposed circuit architecture performs both learning and classification in an entirely analog fashion and includes very low power circuit building blocks. It performs efficiently in binary classification problems, as it was tested with a real dataset and presented errors no more than 1% compared to a traditional software implementation.

Future work related to this hardware architecture could include layout and chip fabrication and measurements of the proposed circuitry. Furthermore, analog and low power circuits for memory storage interfacing with the proposed architecture could be designed, with the aim of storing parameter values. This architecture could also be scaled in order to accomodate input vectors of higher dimension than 13 and with more available learning samples than 8,as it was the case in this work. Apart from SVM implementation, the basic building blocks of this architecture could be further optimized and used in other machine learning hardware implementations, such as other Kernel Methods, Gaussian Mixure Models, K-Nearest Neighbors Algorithm and others.

Δημοσιεύσεις

1)An Ultra-Low Power, ± 0.3V Supply, Fully-Tunable Gaussian function Circuit architecture for Radial-Basis Functions analog hardware implementation / Marios Gourdouparis, Vassilis Alimisis, Christos Dimas, Paul P. Sotiriadis / Elsevier AEÜ - International Journal of Electronics and Communications,Volume 136, July 2021

2)Ultra-Low Power (4nW), 0.6V Fully-Tunable Bump Circuit operating in Sub-threshold regime / Marios Gourdouparis, Vassilis Alimisis, Christos Dimas, Paul P. Sotiriadis / 3rd IEEE 2021 International Conference on Design & Test of integrated micro & nano-Systems, 7-10 June 2021

3)A 0.6V , 3.3nW, Adjustable Gaussian Circuit for Tunable Kernel Functions / Vassilis Alimisis, Marios Gourdouparis, Christos Dimas, Paul P. Sotiriadis / 34rd IEEE Symposium on Integrated Circuits and Systems Design, 23-27 August 2021

4)Ultra-Low Power, Low-Voltage, Fully-Tunable, Bulk-Controlled Bump Circuit / Vassilis Alimisis, Marios Gourdouparis, Christos Dimas, Paul P. Sotiriadis / International Conference on Modern Circuits and Systems Technologies (MOCAST) on Electronics and Communications,5-7 July 2021

5)Implementation of Fractional-order Model of Nickel-Cadmium Cell using Current Feedback Operational Amplifiers / Vassilis Alimisis, Marios Gourdouparis, Christos Dimas, Paul P. Sotiriadis / 2020 IEEE European Conference on Circuit Theory and Design (ECCTD), September 7-10, 2020

[1] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[2] S. Haykin, *Neural networks and learning machines, 3/E*. Pearson Education India, 2010.

[3] M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, *et al.*, "Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences," *NPJ digital medicine*, vol. 2, no. 1, pp. 1–11, 2019.

[4] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE access*, vol. 7, pp. 19143–19165, 2019.

[5] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," *IATSS research*, vol. 43, no. 4, pp. 244–252, 2019.

[6] S. P. Chatzis, V. Siakoulis, A. Petropoulos, E. Stavroulakis, and N. Vlachogiannakis, "Forecasting stock market crisis events using deep and statistical machine learning techniques," *Expert systems with applications*, vol. 112, pp. 353–371, 2018.

[7] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," 2019.

[8] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 27–39, 2016.

[9] K. J. Lee, J. Lee, S. Choi, and H.-J. Yoo, "The development of silicon for ai: Different design approaches," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4719–4732, 2020.

[10] I. Akita, T. Okazawa, Y. Kurui, A. Fujimoto, and T. Asano, "A feed-forward noise reduction technique in capacitive mems accelerometer analog front-end for ultra-low-power iot applications," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 6, pp. 1599–1609, 2019.

[11] E. Farella, M. Rusci, B. Milosevic, and A. L. Murphy, "Technologies for a thing-centric internet of things," in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, vol. 55, pp. 77–84, IEEE, 2017.

[12] A. Shawahna, S. M. Sait, and A. El-Maleh, "Fpga-based accelerators of deep learning networks for learning and classification: A review," *IEEE Access*, vol. 7, pp. 7823–7859, 2018.

[13] W. Haensch, T. Gokmen, and R. Puri, "The next generation of deep learning hardware: Analog computing," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 108–122, 2018.

[14] N. Guo, Y. Huang, T. Mai, S. Patil, C. Cao, M. Seok, S. Sethumadhavan, and Y. Tsividis, "Energy-efficient hybrid analog/digital approximate computation in continuous time," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 7, pp. 1514–1524, 2016.

[15] M. Verhelst and A. Bahai, "Where analog meets digital: Analog? to? information conversion and beyond," *IEEE Solid-state circuits magazine*, vol. 7, no. 3, pp. 67–80, 2015.

[16] M. Jabri, R. J. Coggins, and B. G. Flower. Springer Science & Business Media.

[17] B. J. Sheu and J. Choi, *Neural information processing and VLSI*, vol. 304. Springer Science & Business Media, 2012.

[18] M. Valle, "Analog vlsi implementation of artificial neural networks with supervised on-chip learning," *Analog Integrated Circuits and Signal Processing*, vol. 33, no. 3, pp. 263–287, 2002.

[19] M. Valle. Springer Science & Business Media.

[20] G. M. Bo, D. D. Caviglia, H. Chible, and M. Valle, "1 4 analog vls i on-ch ip learning neural network with learning rate adaptation," *work*, vol. 3, p. 19, 1999.

[21] T. Lehmann. PhD thesis.

[22] C. Lu, B.-X. Shi, and L. Chen, "An on-chip bp learning neural network with ideal neuron characteristics and learning rate adaptation," *Analog Integrated Circuits and Signal Processing*, vol. 31, no. 1, pp. 55–62, 2002.

[23] G. Cauwenberghs, "An analog vlsi recurrent neural network learning a continuous-time trajectory," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 346–361, 1996.

[24] M. Valle and F. Diotalevi, "A dedicated very low power analog vlsi architecture for smart adaptive systems," *Applied soft computing*, vol. 4, no. 3, pp. 206–226, 2004.

[25] G. Cauwenberghs, "Analog vlsi stochastic perturbative learning architectures," *Analog Integrated Circuits and Signal Processing*, vol. 13, no. 1, pp. 195–209, 1997.

[26] A. J. Montalvo, R. S. Gyurcsik, and J. J. Paulos, "An analog vlsi neural network with on-chip perturbation learning," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 4, pp. 535–543, 1997.

[27] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-based neuromorphic systems*. John Wiley & Sons, 2014.

[28] S. Soman, M. Suri, *et al.*, "Recent trends in neuromorphic engineering," *Big Data Analytics*, vol. 1, no. 1, pp. 1–19, 2016.

[29] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in neuroscience*, vol. 5, p. 73, 2011.

[30] S. A. Aamir, Y. Stradmann, P. Müller, C. Pehle, A. Hartel, A. Grübl, J. Schemmel, and K. Meier, "An accelerated lif neuronal network array for a large-scale mixed-signal neuromorphic architecture," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4299–4312, 2018.

[31] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, 2014.

[32] S. Mitra, S. Fusi, and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic vlsi," *IEEE transactions on biomedical circuits and systems*, vol. 3, no. 1, pp. 32–42, 2008.

[33] N. Zheng and P. Mazumder, "Hardware implementations of spiking neural networks," 2020.

[34] S. P. Adhikari, H. Kim, R. K. Budhathoki, C. Yang, and L. O. Chua, "A circuit-based learning architecture for multilayer neural networks with memristor bridge synapses," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 1, pp. 215–223, 2014.

[35] V. Milo, G. Malavena, C. Monzio Compagnoni, and D. Ielmini, "Memristive and cmos devices for neuromorphic computing," *Materials*, vol. 13, no. 1, p. 166, 2020.

[36] V. Saxena, X. Wu, and K. Zhu, "Energy-efficient cmos memristive synapses for mixed-signal neuromorphic system-on-a-chip," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2018.

[37] L. A. Camuñas-Mesa, B. Linares-Barranco, and T. Serrano-Gotarredona, "Neuromorphic spiking neural networks and their memristor-cmos hardware implementations," *Materials*, vol. 12, no. 17, p. 2745, 2019.

[38] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *Ieee Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[39] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. F. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing neuromorphic computing with loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[40] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in neuroscience*, vol. 9, p. 141, 2015.

[41] J.-Y. Kim, M. Kim, S. Lee, J. Oh, K. Kim, and H.-J. Yoo, "A 201.4 gops 496 mw real-time multi-object recognition processor with bio-inspired neural perception engine," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 32–45, 2009.

[42] J. Oh, G. Kim, B.-G. Nam, and H.-J. Yoo, "A 57 mw 12.5 μj/epoch embedded mixed-mode neuro-fuzzy processor for mobile real-time object recognition," *IEEE journal of solid-state circuits*, vol. 48, no. 11, pp. 2894–2907, 2013.

[43] J. Lu, "An analog vlsi deep machine learning implementation," 2014.

[44] R. Zhang and T. Shibata, "An analog on-line-learning k-means processor employing fully parallel self-converging circuitry," *Analog Integrated Circuits and Signal Processing*, vol. 75, no. 2, pp. 267–277, 2013.

[45] K. Lee, J. Park, and H.-J. Yoo, "A low-power, mixed-mode neural network classifier for robust scene classification," *JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE*, vol. 19, no. 1, pp. 129–136, 2019.

[46] S.-Y. Peng, P. E. Hasler, and D. V. Anderson, "An analog programmable multidimensional radial basis function based classifier," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 10, pp. 2148–2158, 2007.

[47] S.-Y. Peng, B. A. Minch, and P. Hasler, "Analog vlsi implementation of support vector machine learning and classification," in *2008 IEEE International Symposium on Circuits and Systems*, pp. 860–863, IEEE, 2008.

[48] K. Kang and T. Shibata, "An on-chip-trainable gaussian-kernel analog support vector machine," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1513–1524, 2009.

[49] R. Zhang and T. Shibata, "Fully parallel self-learning analog support vector machine employing compact gaussian generation circuits," *Japanese Journal of Applied Physics*, vol. 51, no. 4S, p. 04DE10, 2012.

[50] R. Genov and G. Cauwenberghs, "Kerneltron: support vector" machine" in silicon," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1426–1434, 2003.

[51] S. Chakrabartty and G. Cauwenberghs, "Sub-microwatt analog vlsi trainable pattern classifier," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 5, pp. 1169–1179, 2007.

[52] R. Zhang and T. Shibata, "A vlsi hardware implementation study of svdd algorithm using analog gaussian-cell array for on-chip learning," in *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1–6, IEEE, 2012.

[53] R. Zhang, N. Uetake, T. Nakada, and Y. Nakashima, "Design of programmable analog calculation unit by implementing support vector regression for approximate computing," *IEEE Micro*, vol. 38, no. 6, pp. 73–82, 2018.

[54] S. Afifi, H. GholamHosseini, and R. Sinha, "Fpga implementations of svm classifiers: A review," *SN Computer Science*, vol. 1, no. 3, pp. 1–17, 2020.

[55] R. Wiśniewski, G. Bazydło, and P. Szcześniak, "Svm algorithm oriented for implementation in a low-cost xilinx fpga," *Integration*, vol. 64, pp. 163–172, 2019.

[56] H. Wang, W. Shi, and C.-S. Choy, "Hardware design of real time epileptic seizure detection based on stft and svm," *IEEE Access*, vol. 6, pp. 67277–67290, 2018.

[57] A. Luo, F. An, X. Zhang, and H. J. Mattausch, "A hardware-efficient recognition accelerator using haar-like feature and svm classifier," *IEEE Access*, vol. 7, pp. 14472–14487, 2019.

[58] F. F. Lopes, J. C. Ferreira, and M. A. Fernandes, "Parallel implementation on fpga of support vector machines using stochastic gradient descent," *Electronics*, vol. 8, no. 6, p. 631, 2019.

[59] S.-C. Liu, T. Delbruck, J. Kramer, G. Indiveri, R. Douglas, and A. VLSI, "Circuits and principles," 2002.

[60] T. Delbrueck and C. Mead, "Bump circuits," in *Proceedings of International Joint Conference on Neural Networks*, vol. 1, pp. 475–479, 1993.

[61] J. Lu, T. Yang, M. Jahan, and J. Holleman, "Nano-power tunable bump circuit using wide-input-range pseudo-differential transconductor," *Electronics letters*, vol. 50, no. 13, pp. 921–923, 2014.

[62] A. R. Mohamed, L. Qi, Y. Li, and G. Wang, "A generic nano-watt power fully tunable 1-d gaussian kernel circuit for artificial neural network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 9, pp. 1529–1533, 2020.

[63] D. Vrtaric, V. Ceperic, and A. Baric, "Area-efficient differential gaussian circuit for dedicated hardware implementations of gaussian function based machine learning algorithms," *Neurocomputing*, vol. 118, pp. 329–333, 2013.

[64] B. A. Minch, "A simple variable-width cmos bump circuit," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1354–1357, IEEE, 2016.

[65] M. Payvand and G. Indiveri, "Spike-based plasticity circuits for always-on on-line learning in neuromorphic systems," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2019.

[66] S. Moshfe, A. Khoei, K. Hadidi, and B. Mashoufi, "A fully programmable nano-watt analogue cmos circuit for gaussian functions," in *2010 International conference on electronic devices, systems and applications*, pp. 82–87, IEEE, 2010.

[67] F. Li, C.-H. Chang, A. Basu, and L. Siek, "A 0.7 v low-power fully programmable gaussian function generator for brain-inspired gaussian correlation associative memory," *Neurocomputing*, vol. 138, pp. 69–77, 2014.

[68] D. S. Masmoudi, A. T. Dieng, and M. Masmoudi, "A subthreshold mode programmable implementation of the gaussian function for rbf neural networks applications," in *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, pp. 454–459, IEEE, 2002.

[69] M. Melendez-Rodriguez and J. Silva-Martínez, "A fully-programmable temperature-compensated analogue circuit for gaussian functions," in *Proceedings of the Third International Workshop on Design of Mixed-Mode Integrated Circuits and Applications (Cat. No. 99EX303)*, pp. 159–162, IEEE, 1999.

[70] J. A. Bragg, E. A. Brown, and S. P. DeWeerth, "A tunable voltage correlator," *Analog Integrated Circuits and Signal Processing*, vol. 39, no. 1, pp. 89–94, 2004.

[71] A. Shylendra, P. Shukla, S. Mukhopadhyay, S. Bhunia, and A. R. Trivedi, "Low power unsupervised anomaly detection by nonparametric modeling of sensor statistics," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020.

[72] R. Srivastava, M. Gupta, and U. Singh, "Fgmos transistor based low voltage and low power fully programmable gaussian function generator," *Analog Integrated Circuits and Signal Processing*, vol. 78, no. 1, pp. 245–252, 2014.

[73] I. Cevikhas, A. Ogrenci, G. Dundar, and S. Balkur, "Vlsi implementation of grbf (gaussian radial basis function) networks," in *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 3, pp. 646–649, IEEE, 2000.

[74] K. Lee, J. Park, G. Kim, I. Hong, and H.-J. Yoo, "A multi-modal and tunable radial-basis-funtion circuit with supply and temperature compensation," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1608–1611, IEEE, 2013.

[75] J. Choi, B. J. Sheu, and J.-F. Chang, "A gaussian synapse circuit for analog vlsi neural networks," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 2, no. 1, pp. 129–133, 1994.

[76] S.-Y. Lin, R.-J. Huang, and T.-D. Chiueh, "A tunable gaussian/square function computation circuit for analog neural networks," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 3, pp. 441–446, 1998.

[77] M. B. Lucks and N. Oki, "Radial basis function circuits using folded cascode differential pairs," in *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*, pp. 417–421, IEEE, 2010.

[78] M. Azadmehr, L. Marchetti, and Y. Berg, "A low power analog voltage similarity circuit," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, 2017.

[79] C. Popa, "Low-voltage improved accuracy gaussian function generator with fourth-order approximation," *Microelectronics Journal*, vol. 43, no. 8, pp. 515–520, 2012.

[80] A. N. Saatlo and S. Ozoguz, "On the realization of gaussian membership function circuit operating in saturation region," in *2015 38th*

international conference on telecommunications and signal processing (TSP), pp. 1–4, IEEE, 2015.

[81] M. E. Pour and B. Mashoufi, "A low power consumption and compact mixed-signal gaussian membership function circuit for neural/fuzzy hardware," in *2011 International conference on electronic devices, systems and applications (ICEDSA)*, 2011.

[82] S. Azimi and H. Miar-Naimi, "Designing programmable current-mode gaussian and bell-shaped membership function," *Analog Integrated Circuits and Signal Processing*, vol. 102, no. 2, pp. 323–330, 2020.

[83] M. T. Abuelma'Ati and A. Shwehneh, "A reconfigurable gaussian/triangular basis functions computation circuit," *Analog Integrated Circuits and Signal Processing*, vol. 47, no. 1, pp. 53–64, 2006.

[84] M. T. Abuelma'atti and S. R. Al-Abbas, "A new analog implementation for the gaussian function," in *2016 IEEE Industrial Electronics and Applications Conference (IEACon)*, pp. 219–224, IEEE, 2016.

[85] E. Donati, M. Payvand, N. Risi, R. Krause, and G. Indiveri, "Discrimination of emg signals using a neuromorphic implementation of a spiking neural network," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 5, pp. 795–803, 2019.

[86] M. Payvand, M. E. Fouda, F. Kurdahi, A. Eltawil, and E. O. Neftci, "Error-triggered three-factor learning dynamics for crossbar arrays," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 218–222, IEEE, 2020.

[87] A. Dorzhigulov and A. P. James, "Generalized bell-shaped membership function generation circuit for memristive neural networks," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2019.

[88] M. Nam and K. Cho, "Implementation of real-time image edge detector based on a bump circuit and active pixels in a cmos image sensor," *Integration*, vol. 60, pp. 56–62, 2018.

[89] M. Gourdouparis, V. Alimisis, C. Dimas, and P. P. Sotiriadis, "An ultra-low power,±0.3 v supply, fully-tunable gaussian function circuit architecture for radial-basis functions analog hardware implementation," *AEU-International Journal of Electronics and Communications*, vol. 136, p. 153755, 2021.

[90] V. Alimisis, C. Dimas, G. Pappas, and P. P. Sotiriadis, "Analog realization of fractional-order skin-electrode model for tetrapolar bioimpedance measurements," *Technologies*, vol. 8, no. 4, p. 61, 2020.

[91] I. Dimeas, I. Petras, and C. Psychalinos, "New analog implementation technique for fractional-order controller: A dc motor control," *AEU-International Journal of Electronics and Communications*, vol. 78, pp. 192–200, 2017.

[92] G. Tsirimokou, C. Psychalinos, and A. S. Elwakil, "Emulation of a constant phase element using operational transconductance amplifiers," *Analog Integrated Circuits and Signal Processing*, vol. 85, no. 3, pp. 413–423, 2015.

[93] P. Prommee and K. Chattrakun, "Cmos wta maximum and minimum circuits with their applications to analog switch and rectifiers," *Microelectronics journal*, vol. 42, no. 1, pp. 52–62, 2011.

[94] E. Rahiminejad, M. Saberi, R. Lotfi, M. Taherzadeh-Sani, and F. Nabki, "A low-voltage high-precision time-domain winner-take-all circuit," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 4–8, 2019.

[95] T. Kulej and F. Khateb, "Sub 0.5-v bulk-driven winner take all circuit based on a new voltage follower," *Analog Integrated Circuits and Signal Processing*, vol. 90, no. 3, pp. 687–691, 2017.

[96] A. Fish, V. Milrud, and O. Yadid-Pecht, "High-speed and high-precision current winner-take-all circuit," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 3, pp. 131–135, 2005.

[97] D. Moro-Frias, C. Ventura-Arizmendi, M.-T. Sanz-Pascual, and C.-A. de la Cruz-Blas, "Current-mode winner-take-all circuit with improved dynamic response," *Circuits, Systems, and Signal Processing*, vol. 34, no. 2, pp. 625–639, 2015.

[98] S. Ramakrishnan and J. Hasler, "A compact programmable analog classifier using a vmm+ wta network," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2538–2542, IEEE, 2013.

[99] G. Indiveri, "A current-mode hysteretic winner-take-all network, with excitatory and inhibitory coupling," *Analog Integrated Circuits and Signal Processing*, vol. 28, no. 3, pp. 279–291, 2001.

[100] H. Huang and N. Baddour, "Bearing vibration data under time-varying rotational speed conditions," *Mendeley Data*, 2018.