



UNIVERSIDADE ESTADUAL PAULISTA

Pós-Graduação em Engenharia Elétrica

**“Implementação de uma Rede Neural de Base
Radial Utilizando Tecnologia BiCMOS”**

JEFERSON PONTIES DE OLIVEIRA

Ilha Solteira - SP

1210001300



1300

UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DISSERTAÇÃO DE MESTRADO

“Implementação de uma Rede Neural de Base
Radial Utilizando Tecnologia BiCMOS”

Proc. 071/2002 - NPD 061/02

UNIVERSIDADE ESTADUAL PAULISTA - CAMPUS DE ILHA SOLTEIRA	
SERVIÇO TÉCNICO DE BIBLIOTECA E DOCUMENTAÇÃO	
DATA DE CHEGADA	DATA DE TOMBO
19.9.02	30.09.02
RECEBIDO POR	TOMBO
Ailza	Te. 1300
AQUISIÇÃO	CLASSIFICAÇÃO
ilustração anexo R\$ 10,00	048 i

JEFERSON PONTIES DE OLIVEIRA

Dissertação apresentada à Comissão de Pós-Graduação da Faculdade de Engenharia de Ilha Solteira, como requisito para a obtenção do título de Mestre em Engenharia Elétrica

sup - 183396
sup - 54653

ORIENTADOR: PROF. DR. NOBUO OKI

ILHA SOLTEIRA - SP

MAIO - 2002



30400007
ENGENHARIA ELÉTRICA

IMPLEMENTAÇÃO DE UMA REDE NEURAL DE BASE RADIAL UTILIZANDO TECNOLOGIA BiCMOS

Jeferson Ponties de Oliveira

**DISSERTAÇÃO SUBMETIDA À FACULDADE DE ENGENHARIA -
CAMPUS DE ILHA SOLTEIRA – UNESP – COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO TÍTULO DE
MESTRE EM ENGENHARIA ELÉTRICA (ME).**



Prof. Dr. Antonio Padilha Feltrin – Coordenador

COMISSÃO EXAMINADORA:



Prof. Dr. Nobuo Oki – orientador



Dr. Saulo Finco



Prof. Dr. Alexandre César Rodrigues da Silva

Ilha Solteira, maio de 2002

*Aos meus pais,
pelo apoio e
carinho.*

BCpIS - FEIS - UNESP



Agradecimentos

Declaro meus agradecimentos a todos aqueles que, durante a elaboração desta dissertação, contribuíram de alguma forma para sua realização.

Ao meu orientador, Prof. Dr. Nobuo Oki, pela oportunidade de aprimoramento dos meus conhecimentos e pela atenção especial recebida durante a realização do trabalho.

À Coordenadoria de Aperfeiçoamento de Pessoal de Ensino Superior (CAPES)

Aos Professores do curso de Pós-graduação em Engenharia Elétrica, pela contribuição em minha formação acadêmica.

Ao Centro de Pesquisa Renato Archer (CenPRA), em especial ao Dr. Saulo Finco pelo apoio técnico à realização desse projeto.

À UNICAMP, em especial ao Prof. Dr. Carlos Alberto dos Reis Filho, pelo apoio científico desprendido.

Ao BSTC (Brazil Semiconductor Technology Center – Motorola) pelo apoio técnico.

A aqueles companheiros que de uma maneira especial, contribuíram para essa realização: Roberto Ianini, Wellington Melo e Paula.

Aos amigos e colegas do departamento de engenharia elétrica.

A minha família pelo apoio e compreensão nas horas difíceis.

E acima de tudo, agradeço a DEUS.



Resumo

Esse trabalho descreve a implementação analógica de uma Rede Neurál de Base Radial (RNBR) em tecnologia BiCMOS. Um sintetizador analógico programável de função gaussiana é proposto para a implementação. Esse sintetizador pode gerar uma função gaussiana, obtida através da característica tangente hiperbólica do par diferencial bipolar. Os parâmetros da gaussiana (ganho, centro e largura) podem ser alterados com uma fonte de corrente programável. Resultados experimentais do protótipo fabricado usando processo AMS 0.8 μ m BiCMOS, mostram a viabilidade prática dessa proposta.



Abstract

This work describes an analog implementation of Radial Basis Neural Networks (RBNN) in BiCMOS technology. A programmable analog gaussian function synthesizer is proposed for the implementation. This synthesizer can generate a gaussian function, obtained through the hyperbolic tangent characteristic of the bipolar differential pair. The gaussian parameters (gain, center and width) can be changed with a programmable current source. Experimental results from the prototype fabricate using the AMS 0.8um BiCMOS process show the feasibility of this propose.



Sumário

1) INTRODUÇÃO	1
1.1) MOTIVAÇÃO	2
1.2) OBJETIVOS	3
2) APRESENTAÇÃO	4
2.1) A IDÉIA	4
2.2) A PROPOSTA	4
3) DEFINIÇÕES	8
3.1) TECNOLOGIA	8
3.2) FAIXA OPERACIONAL	9
3.3) METODOLOGIA	9
4) BLOCO GERADOR DE FUNÇÃO GAUSSIANA	11
4.1) CARACTERIZAÇÃO	11
4.2) CONCEPÇÃO	18
4.3) ANÁLISE E DIMENSIONAMENTO	20
5) BLOCO CONTROLADOR DE PARÂMETROS	21
5.1) CARACTERIZAÇÃO	21
5.2) CONCEPÇÃO	22
5.3) ANÁLISE E DIMENSIONAMENTO	25
6) SIMULAÇÃO	27
6.1) AMBIENTE DE SIMULAÇÃO	27
6.2) DISPOSIÇÃO DOS BLOCOS	27
6.3) ANÁLISE DOS RESULTADOS	29
7) IMPLEMENTAÇÃO	34
7.1) LEIAUTE	34
8) TESTES	37
8.1) AMBIENTE DE TESTE	37
8.2) RESULTADOS EXPERIMENTAIS	37
9) CONCLUSÕES:	42
10) BIBLIOGRAFIA	43
APÊNDICE 1	46
APÊNDICE 2	51



Índice de Tabelas

TABELA 1: PASSO MÍNIMO DE CORRENTE EM MICRO AMPERES	9
TABELA 2: VALORES DOS TRANSISTORES DOS OTA'S	20
TABELA 3: VALORES DOS TRANSISTORES DA FONTE DE CORRENTE.	26

Índice de Figuras

FIGURA 1: DIAGRAMA ESQUEMÁTICO DE UMA RNFBR	5
FIGURA 2: DIAGRAMA BÁSICO DA REDE NEURAL IDEALIZADA	6
FIGURA 3: PAR DIFERENCIAL BIPOLAR	11
FIGURA 4: CURVA CARACTERÍSTICA DO PAR DIFERENCIAL	12
FIGURA 5: CURVA DE TRANSCONDUTÂNCIA DO PAR DIFERENCIAL	13
FIGURA 6: IMPLEMENTAÇÃO COMPUTACIONAL DA FUNÇÃO GAUSSIANA.	14
FIGURA 7: IMPLEMENTAÇÃO COMPUTACIONAL DA FUNÇÃO SECANTE HIPERBÓLICA QUADRÁTICA	15
FIGURA 8: COMPARAÇÃO DAS FUNÇÕES IMPLEMENTADAS COMPUTACIONALMENTE.	15
FIGURA 9: SIMBOLOGIA E ESQUEMA SIMPLIFICADO DO OTA	16
FIGURA 10: ARQUITETURA PROPOSTA PARA O CIRCUITO GERADOR DA FUNÇÃO GAUSSIANA.	17
FIGURA 11: CIRCUITO DO OTA COM OS DIODOS DE LINEARIZAÇÃO.	18
FIGURA 12: CIRCUITO COMPLETO DO GERADOR DE GAUSSIANA	19
FIGURA 13: ARQUITETURA PROPOSTA PARA O CIRCUITO CONTROLADOR DE PARÂMETROS.	21
FIGURA 14: CIRCUITO DETALHADO DO MULTIPLICADOR DE CORRENTE.	22
FIGURA 15: CONVERSOR CORRENTE-TENSÃO	24
FIGURA 16: CIRCUITO COMPLETO DAS QUATRO FONTES DE CONTROLE.	24
FIGURA 17: REGISTRADOR DE DESLOCAMENTO.	25
FIGURA 18: CIRCUITO DE TESTE DA REDE NEURAL PROPOSTA	28
FIGURA 19: CONFIGURAÇÃO DE TESTE DO CI.	29
FIGURA 20: GRÁFICO DA ENTRADA COMPARADA A SAÍDA	30
FIGURA 21: GRÁFICO DA VARIAÇÃO DO CENTRO DA GAUSSIANA	31
FIGURA 22: GRÁFICO DA VARIAÇÃO DA AMPLITUDE DA GAUSSIANA	32
FIGURA 23: GRÁFICO DA VARIAÇÃO DA LARGURA DA GAUSSIANA	33
FIGURA 24: LEIAUTE DA REDE COMPLETA	35
FIGURA 25: LEIAUTE DO CIRCUITO INTEGRADO ENVIADO PARA FABRICAÇÃO	36
FIGURA 26: COMPARAÇÃO DAS CURVAS DE ENTRADA E SAÍDA DO PROTÓTIPO.	38
FIGURA 27: CURVA DE VARIAÇÃO DO CENTRO DA GAUSSIANA	38
FIGURA 28: CURVA DE VARIAÇÃO DA AMPLITUDE DA GAUSSIANA	39
FIGURA 29: CURVA DA VARIAÇÃO DA LARGURA DA GAUSSIANA	40
FIGURA 30: CHIP COM DOIS SINTETIZADORES COMPLETOS IMPLEMENTADOS	51
FIGURA 31: CIRCUITO COMPLETO DO SINTETIZADOR	51
FIGURA 32: BLOCO GERADOR DA FUNÇÃO GAUSSIANA	52
FIGURA 33: AMPLIFICADOR OPERACIONAL DE TRANSCONDUTÂNCIA OTA	52
FIGURA 34: BLOCO CONTROLADOR DE PARÂMETROS	52
FIGURA 35: FONTE DE CORRENTE DE CONTROLE	52



FIGURA 36: REGISTRADOR DE DESLOCAMENTO.....53

FIGURA 37: IDENTIFICAÇÃO DOS REALIZADORES.....53



1) INTRODUÇÃO

A implementação analógica de redes neurais usando técnicas de fabricação VLSI (*Very Large Scale Integration*) é um importante campo de interesse, pois permite uma realização eficiente de funções aritméticas que são pré-requisitos na implementação de redes neurais. Funções como a adição e a multiplicação podem ser implementadas em pequenos circuitos analógicos permitindo assim a construção de grandes séries paralelas desses circuitos.

Por outro lado, os circuitos analógicos apresentam limitações conhecidas, como por exemplo, ser suscetível à variação de processo e ruído. Para a minimização desses problemas, um procedimento de aprendizado adaptativo pode ser adotado, como uma forma de eliminar imperfeições inerentes ao processo VLSI analógico, durante o aprendizado do chip. Para isso, torna-se necessário que a rede neural possa ser treinada através de programação durante a aprendizagem do *chip*.

Outro problema quando se propõe a implementação analógica de uma rede neural, é a definição da configuração a ser implementada, que normalmente inclui uma rede neural supervisionada e o algoritmo de *back-propagation* para o treinamento da camada oculta do *chip*. Porém, esta escolha traz dificuldades quando a implementação é realizada, pois o algoritmo de *back-propagation* não permite acessar a camada oculta durante o aprendizado, impedindo assim a minimização dos problemas causados durante a concepção dos circuitos em VLSI.

Uma possibilidade é a escolha de uma rede neural de base radial que apresenta três camadas e apenas a segunda (camada oculta) necessita ser treinada, permitindo então que o treinamento dessa camada seja realizado durante o aprendizado do *chip*, que nesse caso, não depende do algoritmo de *back-propagation*.

O algoritmo da rede neural de base radial pode processar medidas não lineares por meio de funções gaussianas.

As funções gaussianas programáveis são utilizadas freqüentemente em processadores de sinais complexos, como processadores de imagens, sistemas de compressão de dados e análise de sinais biomédicos, para transformação de pequenas variações na representação de sinais em freqüência no tempo. Essa transformação utiliza a



função gaussiana para escalonar o espaço variável, usando uma família de funções com janelas de formatos idênticos^[11].

As funções gaussianas também são utilizadas com frequência em algoritmos adaptativos para solucionar problemas de classificação de padrões^[12]. Se as funções gaussianas forem utilizadas nas unidades ocultas da rede neural com *back-propagation*, o tempo de treinamento também pode ser reduzido, em comparação com redes tradicionais implementadas com funções sigmoidais.

Recentemente, muitas propostas de circuitos têm sido apresentadas^[1,14], mas para obter maiores vantagens das propriedades computacionais dessas transformações, recomenda-se a utilização de um gerador de função gaussiana adaptativo.

Nesse trabalho, serão apresentadas as principais etapas de implementação de um sintetizador analógico programável, capaz de gerar uma função gaussiana, baseando-se em uma rede neural de base radial adaptativa e utilizando técnicas de circuitos integrados analógicos.

1.1) Motivação

Quando se trata de redes neurais artificiais, o maior objetivo é a construção física de um neurocomputador que se equipare a um cérebro humano. Mas, a implementação física traz grandes dificuldades em função de limitações e imperfeições inerentes aos processos de implementação, transformando esse objetivo em um grande desafio aos projetistas de circuitos integrados.

Tendo em vista que, a superação de parte dessas dificuldades, poderiam ser eliminadas através da própria programabilidade da rede, o desafio de se projetar um circuito que garantisse de forma satisfatória o controle dessa programação, também se apresentou como uma ótima oportunidade de desenvolvimento de mais um aprendizado.

Paralelamente a essas desafiadoras propostas de superar tais dificuldades, observou-se a grande oportunidade da realização do objetivo maior da engenharia, que se resume em transformar uma idéia em realidade (teoria em prática), buscando sempre prover benefícios à humanidade.



1.2) Objetivos

Baseando-se nas oportunidades observadas, definiu-se então os objetivos apresentados a seguir.

- Teórico:

Desenvolver um conjunto de blocos funcionais, que dessem condições plenas de comprovar a eficiência da rede neural idealizada para o sintetizador analógico programável.

- Prático:

Realizar a implementação física desses blocos, através da fabricação de protótipos em tecnologia BiCMOS de 0.8μ , buscando superar os desafios práticos que envolvem esse tipo de atividade, de forma que isso confirme os objetivos teóricos.



2) APRESENTAÇÃO

Neste capítulo apresenta-se uma proposta de implementação da rede, idealizada em função das necessidades conhecidas.

2.1) A Idéia

A rede neural de base radial, escolhida em função da necessidade de superar os problemas decorrentes do processo de implementação, tem como função de base, a função gaussiana, que é obtida através da característica tangente hiperbólica do par diferencial bipolar e que permite sua caracterização através do controle dos seus parâmetros de forma que são o centro, a largura e a amplitude. A seleção desses parâmetros, através de blocos adjacentes baseados em fontes de correntes programáveis, será controlada computacionalmente e tornará possível o aprendizado adaptativo da rede neural idealizada.

2.2) A Proposta

De forma geral, uma função qualquer $y(x)$ pode ser representada pela soma ponderada de um número finito de funções de base $f(\cdot)$, conforme apresentado na equação 1.

$$y(x) \equiv \sum_{i=1}^M w_i f(x, p_i) + w_0 B \quad (1)$$

Sendo:

M = número de funções de base

w_i = peso referente à i -ésima função de base

w_0 = peso referente ao *bias*



$B = \text{bias}$ (em geral, $B=1$)

p_i = vetor de parâmetros da i -ésima função de base que inclui fatores de forma e de deslocamento.

Os parâmetros M , w_i , w_0 , e p_i , podem ser ajustados durante o processo de aprendizado da rede que utilizará essa função de base.

A equação 1 pode aproximar qualquer função arbitrária com um grau desejável de precisão, desde que se disponha de um número suficientemente grande de unidades intermediárias (camada oculta), e que a função de base $f(x, p_i)$ seja absolutamente integrável ou uma função monotônica limitada^[6].

Na figura 1, apresenta-se um esquemático de uma Rede Neural com Função de Base Radial (RNFBR), baseada na equação 1.

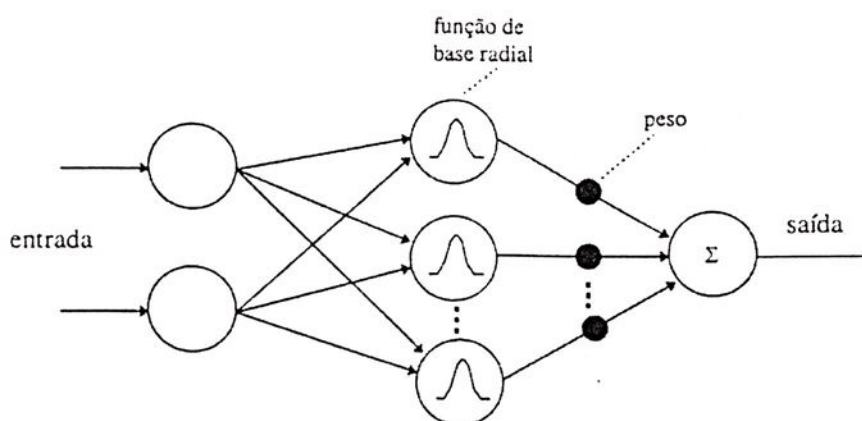


Figura 1: Diagrama esquemático de uma RNFBR

As funções radiais formam uma classe especial de funções, que se enquadram nos requisitos citados e têm como característica básica, a apresentação de uma resposta monotonicamente crescente ou decrescente, a partir de um centro^[6].

A função gaussiana faz parte dessa classe e pode ser submetida ao processo de aprendizagem que é composto por quatro etapas:

- 1) Determinação do numero de unidades da camada intermediária;
- 2) Localização dos centros;
- 3) Determinação da largura das funções;
- 4) Ajuste dos pesos de ligação entre as camadas intermediárias e de saídas.

Essas etapas de aprendizado podem ser realizadas através de várias metodologias, e esse processo é destacado pelo fato de ser um processo de grande facilidade e rapidez. Mas, essas características podem ser influenciadas por vários fatores relacionados a seguir:

- Função de base radial utilizada;
- Dimensão do vetor de entrada;
- Grau de liberdade permitido no ajuste dos parâmetros da rede (centro, largura e amplitude).

Desses fatores, os dois últimos são realmente importantes e indicam a viabilidade ou não, de utilizar uma rede neural de base radial na solução de um determinado problema.

Na figura 2 apresenta-se um diagrama básico da RNFBR idealizada.

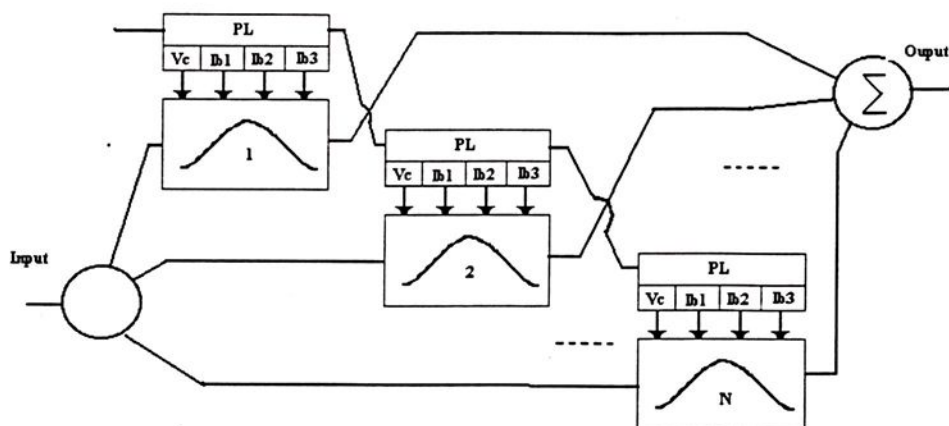


Figura 2: Diagrama básico da rede neural idealizada.

Muitas propostas de implementação da rede idealizada surgiram^[1-4], mas com as necessidades apresentadas, um diagrama básico da rede neural de base radial com 3 camadas de processamento foi adotado.

- A primeira camada (*input*) é linear e apenas distribui o sinal de entrada.
- A segunda camada ($1, 2, \dots, n$) é não linear e realiza a função de base radial.
- A terceira camada (*output*) realiza a combinação linear das saídas gaussianas.

Como se pode observar, a segunda camada pode conter tantos blocos quanto forem necessários, para que o número de amostras satisfaça a necessidade das aplicações. Cada bloco inserido na segunda camada é constituído por dois grandes sub-blocos.

O sub-bloco principal realiza a função gaussiana e o outro sub-bloco realiza o controle dos parâmetros dessa função. Esses sub-blocos serão analisados separadamente como dois blocos independentes, considerando a relação de interdependência entre os mesmos. Essa análise será feita detalhadamente, conforme as definições apresentadas no próximo capítulo.



3) DEFINIÇÕES

Nesse capítulo serão apresentadas definições como a tecnologia escolhida para atender as necessidades do sistema, a faixa de trabalho e a metodologia de análise adotada para cada bloco.

3.1) Tecnologia

Buscando-se atingir o objetivo prático apresentado, que por consequência leva a atingir o objetivo teórico, procurou-se adequar os circuitos a uma tecnologia atual, confiável, que suprisse as necessidades de implementação da rede e permitisse a fabricação de protótipos para testes reais. Decidiu-se então pela tecnologia BiCMOS 0.8mm da AMS (*Áustria Mikro Systeme*) que se enquadrava nos requisitos citados acima.

A definição pela utilização da tecnologia BiCMOS na concepção dos circuitos, ocorreu em função da necessidade da utilização da característica exponencial do transistor bipolar na implementação do par diferencial do amplificador operacional de transcondutância (*Operational Transconductance Amplifier - OTA*), que é responsável pela geração da gaussiana.

A tecnologia BiCMOS combina transistores Bipolares e CMOS em um único circuito integrado. Por possuir os benefícios das tecnologias Bipolar e CMOS, a tecnologia BiCMOS é capaz de prover circuitos em VLSI com desempenhos de velocidade, de potência e de densidade jamais alcançados anteriormente com qualquer uma das tecnologias individualmente. As principais vantagens da tecnologia BiCMOS são listadas a seguir:

- Maior Velocidade que a CMOS;
- Menor dissipação de potência que a Bipolar (simplificando o empacotamento e requisitos da placa);
- I/Os flexíveis (TTL, CMOS ou ECL);
- Alto desempenho analógico;



- Imunidade a *latchup*.

Adquiriu-se então as bibliotecas dos modelos fornecidos pela AMS e partiu-se para o dimensionamento dos circuitos, definindo sua faixa de trabalho, sua alimentação e os valores de seus componentes de maneira a satisfazer as exigências do processo tecnológico escolhido e as necessidades da rede.

3.2) Faixa Operacional

Baseando-se no fato de que a grande maioria das aplicações relativas a funcionalidade da rede proposta ocorre na faixa de micro ampères e de que a tensão de alimentação exigida pelo processo de fabricação deve estar compreendida em uma faixa de no máximo 5 Volts entre Vdd e Vss, procurou-se definir essa faixa como limite na busca de valores associados à faixa de trabalho do circuito.

Como o circuito trabalha em função de correntes controladas por lógica binária, definiu-se então o passo mínimo dessa corrente ($2,5\mu\text{A}$), de forma que qualquer valor definido pela palavra de 8 bits, que controlará o fornecimento de corrente, não saísse da faixa de micro amperes conforme mostrado na tabela 1.

Bit	0	1	2	3	4	5	6	7
Corrente (μA)	2,5	5	10	20	40	80	160	320

Tabela 1: Passo mínimo de corrente em micro ampères

Assim, garantiu-se que mesmo quando a palavra habilitasse todas as chaves, a soma máxima em corrente será de $637,5\mu\text{A}$.

3.3) Metodologia



Cada bloco projetado passou por seis etapas distintas de projeto, apresentadas a seguir:

- Caracterização: Especificação da função a ser realizada pelo bloco e a forma que o bloco deverá ser configurado para a realização dessa função.
- Concepção: Definição dos circuitos que irão compor cada bloco para realizar as funções definidas.
- Dimensionamento: Definição de valores para os componentes que compõe cada circuito.
- Simulação: Simulação computacional do bloco para comprovação da funcionalidade.
- Implementação: Confecção do leiaute para fabricação.
- Teste: Testes e medições nos protótipos para obtenção de resultados reais.



4) BLOCO GERADOR DE FUNÇÃO GAUSSIANA

Neste capítulo são apresentados as três primeiras etapas de projeto do principal bloco do sistema, que é responsável pela geração da função de base da rede neural.

4.1) Caracterização

O circuito proposto baseia-se na curva característica de transcondutância de um par diferencial bipolar. O par diferencial apresentado na figura 3, é um dos circuitos mais utilizados na implementação de redes neurais e gera como saída, a função diferença entre as duas tensões de entrada conforme descrito na figura 3.

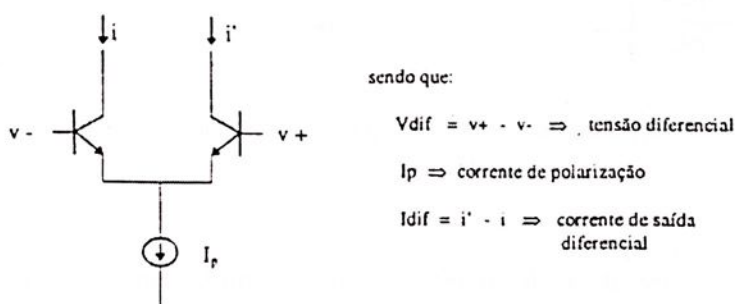


Figura 3: Par diferencial bipolar.

A equação que rege o par diferencial é dada por:

$$I_{dif} = \alpha_F \cdot I_p \cdot \tanh\left(\frac{V_{dif}}{2V_t}\right) \quad (2)$$

Sendo:

α_F = relação entre as correntes de coletor e emissor do transistor bipolar;

V_t = equivalente térmico de tensão.

A curva característica de saída do par diferencial é mostrada na figura 4.

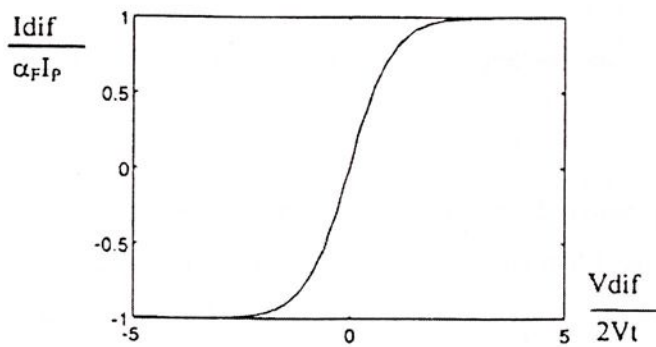


Figura 4: Curva característica do par diferencial.

A equação de transcondutância do par diferencial é dada por:

$$g = \frac{\partial I_{dif}}{\partial V_{dif}} = \frac{\alpha_F \cdot I_p}{2V_t} \cdot \text{sech}^2\left(\frac{V_{dif}}{2V_t}\right) \quad (3)$$

Sendo:

g = transcondutância do par diferencial

A curva característica de transcondutância do par diferencial é mostrada na figura 5.

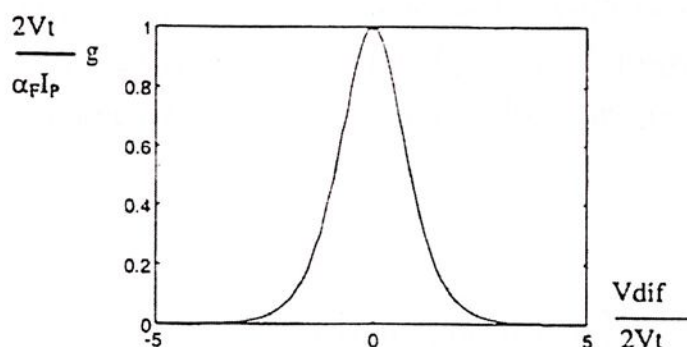


Figura 5: Curva de transcondutância do par diferencial.

Essa curva é uma secante hiperbólica quadrática derivada da função tangente hiperbólica relativa a curva característica do par diferencial. Essa derivada pode ser aproximada por uma diferença conforme descrito a seguir.

A fórmula da derivada de uma determinada função $f(x)$ é dada por:

$$f'(x) = \lim_{\Delta \rightarrow 0} \frac{f(x + \Delta) - f(x)}{\Delta} \quad (4)$$

Daí, depreende-se que para Δ suficientemente pequeno, pode-se conseguir uma boa aproximação da derivada. Assim, com dois pares diferenciais e uma pequena tensão de *offset* aplicada na entrada de um deles, garante-se uma implementação eletrônica da função derivada bem próxima da desejada conforme mostrado na equação 5.

$$f'(x) \cong \frac{f(x + \Delta) - f(x)}{\Delta} \quad (5)$$

Então, conforme descrito, para gerar a função gaussiana, o bloco utiliza a característica de um par diferencial bipolar cuja resposta é uma tangente hiperbólica que quando diferenciada, gera uma secante hiperbólica quadrática que se aproxima muito da função gaussiana apresentada na equação 6.

$$f(x, w, b, c) = w \cdot \exp[-b(x - c)^2] \quad (6)$$

Sendo:

c = centro

b = largura

w = amplitude

Pode-se comparar essa aproximação através de uma simulação computacional da equação (6).

O resultado apresentado nas figuras 6, 7 e 8 foi obtido através da utilização do programa de simulação Matlab.

FUNÇÃO GAUSSIANA

```
c=0;
b=1;
w=1;
x=-5:.1:5
y=w.*(exp(-b.*(x-c).^2))
```

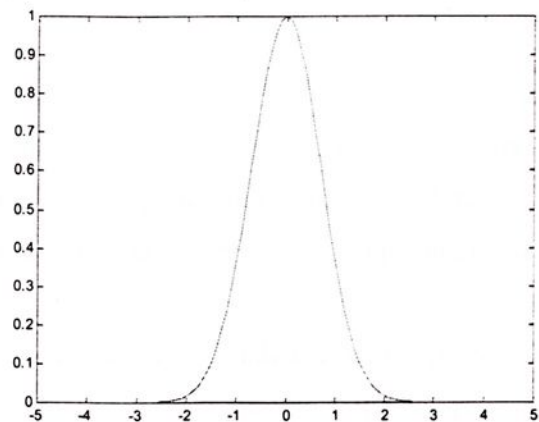


Figura 6: Implementação computacional da função gaussiana.

FUNÇÃO SECANTE
HIPERBÓLICA QUADRÁTICA

```

c=0;
b=1;
w=1;
s=-5:1:5
h=w.*(sech(b.*(s-c))).^2

```

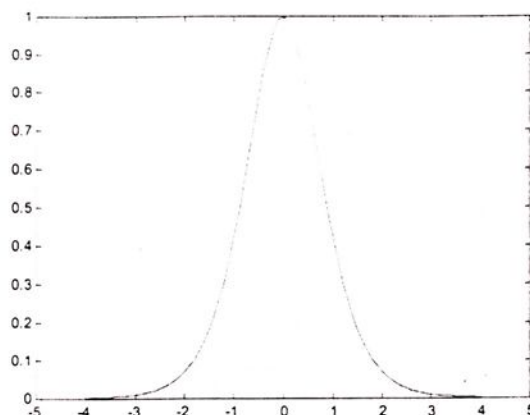


Figura 7: Implementação computacional da função secante hiperbólica quadrática.

COMPARAÇÃO

```

c=0;
b=1;
w=1;
x=-5:1:5 s=-5:1:5
y=w.*(exp(-b.*(x-c).^2))
h=w.*(sech(b.*(s-c))).^2

```

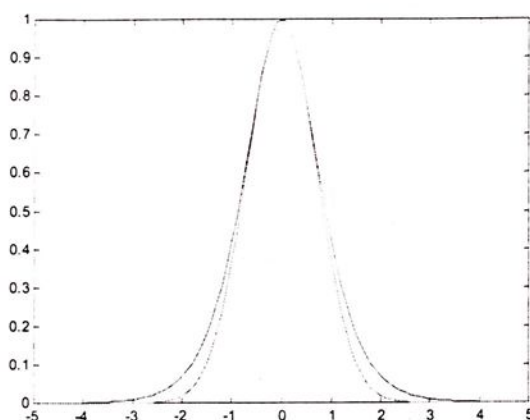


Figura 8: Comparação das funções implementadas computacionalmente.

No entanto, para a implementação prática do circuito, decidiu-se utilizar Amplificadores Operacionais de Transcondutância (OTA), para uma maior facilidade de implementação e casamento entre componentes, sem perder a característica fundamental do par diferencial.

O OTA consiste basicamente em um par diferencial na entrada e alguns espelhos de corrente, que realizam a subtração das correntes de coletor dos transistores que compõem o par diferencial, conforme mostrado na figura 9.

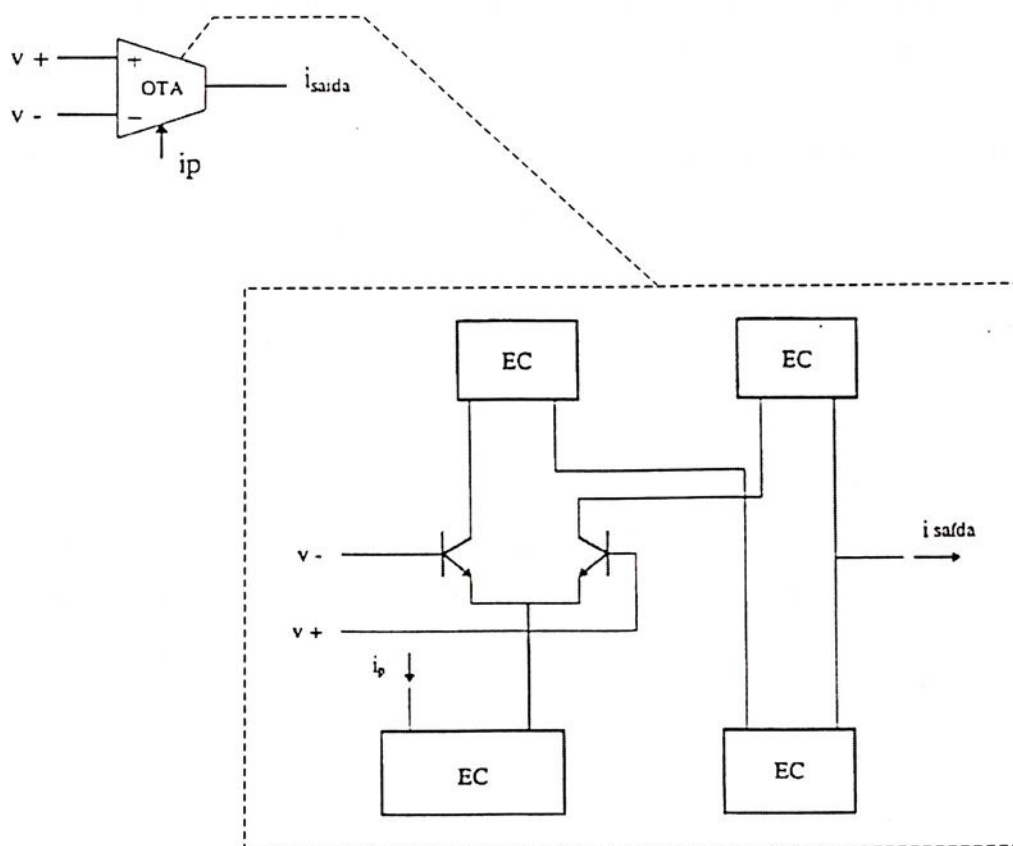


Figura 9: Simbologia e esquema simplificado do OTA

O comportamento do OTA é descrito pela equação 7.

$$I_{saida} = g \cdot I_p \cdot V_{ent} \quad (7)$$

Sendo:

g = transcondutância do OTA;

I_p = corrente de polarização;

V_{ent} = tensão diferencial na entrada.

O OTA efetua uma multiplicação entre a corrente de polarização e a tensão diferencial na entrada e apresenta como resposta característica uma curva praticamente idêntica a do par diferencial, e essa mesma similaridade pode ser considerada para a curva de transcondutância.

A programabilidade^[7] dos parâmetros dessa aproximação de função, é obtida através da utilização desses OTA's na implementação desse bloco.

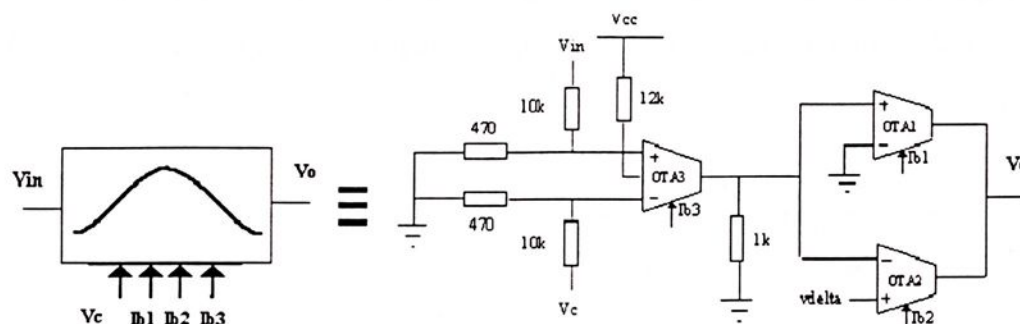


Figura 10: Arquitetura proposta para o circuito gerador da função gaussiana.

O OTA, além de preservar a característica tangente hiperbólica da resposta do amplificador diferencial bipolar, permite o controle do ganho de transcondutância linearizado^[6] através de corrente induzida.

O circuito proposto é mostrado na figura 10, onde o primeiro amplificador de transcondutância (OTA₃) possui uma rede de linearização^[5] na entrada e executa a função de multiplicador permitindo que se varie o centro e a largura da função através das entradas de tensão Vc e de corrente Ib₃ respectivamente.

Essa rede de linearização é normalmente utilizada nos OTA's para adequá-los a aplicações em circuitos lineares normalmente controlados por corrente e que exigem uma faixa linear mais abrangente na entrada.

Os outros dois amplificadores, OTA₁ e OTA₂, realizam uma função secante hiperbólica quadrática fixa em relação ao centro e a largura e possibilitam o controle da amplitude dessa função através de Ib₁ e Ib₂. Essa função é a derivada da tangente

hiperbólica realizada no OTA₃ obtida através da aplicação de uma pequena tensão fixa de *offset* em V_{delta} .

4.2) Concepção

Conforme a arquitetura proposta, o circuito gerador da função gaussiana é formado por um conjunto de três OTA's sendo um deles linearizado através de uma rede externa. São apresentadas a seguir as configurações do circuito OTA3 utilizado nesse bloco.

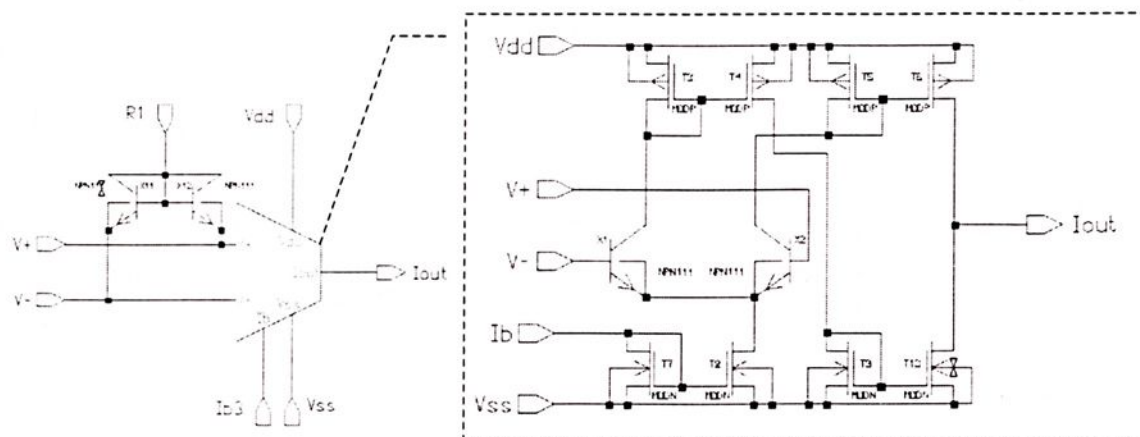


Figura 11: Circuito do OTA com os diodos de linearização.

Na figura 11, os transistores bipolares X_1 e X_2 operam como par diferencial, os transistores MOS T_3 - T_{10} formam espelhos de corrente e os transistores bipolares X_{11} e X_{12} são diodos conectados para a característica de linearização do OTA₃.

A saída linearizada ^[5] pode ser representada conforme a equação 8.

$$I_{out} = g_m \cdot (V_{(+)} - V_{(-)}) \quad (8)$$

$$g_m \cong \frac{I_{b3}}{12.2V} \quad (9)$$

Sendo:

12,2V = Fator de escala para operações na faixa de linearidade com erro insignificante onde: $\{-10V < (V_{(+)} - V_{(-)}) < 10V\}$.

Com a definição do circuito OTA₃, tem-se então o esquemático completo do gerador de gaussianas mostrado na figura 12.

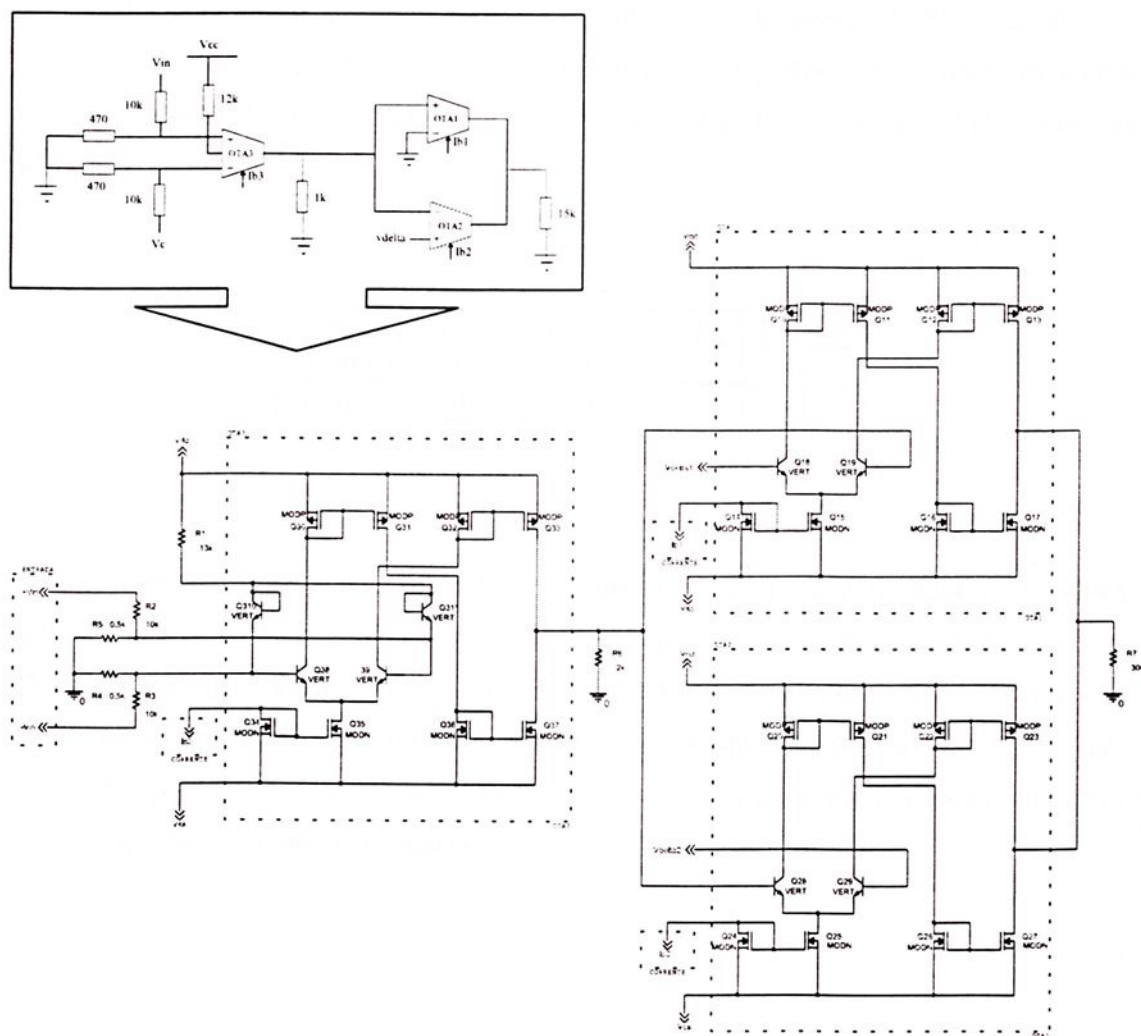


Figura 12: Circuito completo do gerador de gaussiana

Esse circuito portanto, possui as quatro entradas necessárias ao controle da função gerada, sendo uma delas controlada por tensão e as outras por corrente, conforme discriminado a seguir.

V_c = Tensão induzida para o controle do centro da gaussiana.

I_{b3} = Corrente induzida para o controle da largura da gaussiana.

I_{b1} e I_{b2} = Correntes induzidas para o controle da amplitude da gaussiana.

4.3) Análise e Dimensionamento

Com a definição da corrente máxima a ser induzida para o controle da transcondutância do OTA, passou-se a dimensionar os componentes do bloco gerador de gaussiana conforme a tabela 2, buscando ajustar os espelhos de corrente para que desempenhassem de maneira eficiente suas funções, permitindo assim o funcionamento adequado dos OTA's.

Transistor	T_3-T_6	T_7-T_{10}	X_1-X_2	$X_{11}-X_{12}$
W (μm)	300	120	2	2
L (μm)	10	10	1	1

Tabela 2: Valores dos transistores dos OTA's.

Observa-se que a relação W/L dos transistores T_3-T_6 que formam os espelhos PMOS é 2,5 vezes maior que a relação dos transistores T_7-T_{10} que formam os espelhos NMOS, para que dessa forma sejam eliminados os possíveis *offsets* sistemáticos normalmente apresentados pelos OTA's. Os altos valores apresentados nessa relação foram necessários para que fossem evitados problemas de modulação de canal, diminuindo assim as diferenças nas correntes espelhadas.



5) BLOCO CONTROLADOR DE PARÂMETROS

Neste capítulo são apresentados as três primeiras etapas de projeto do bloco do sistema que é responsável pelo controle dos parâmetros da função implementada.

5.1) Caracterização

Para a realização do controle das quatro entradas analisadas anteriormente, torna-se necessária a implementação de um bloco que forneça de maneira controlada e independente, os sinais induzidos que alimentarão essas entradas. Para isso, o bloco será composto por três fontes de corrente, uma fonte de tensão e uma lógica programável que controlará essas fontes. A arquitetura proposta é mostrada na figura 13.

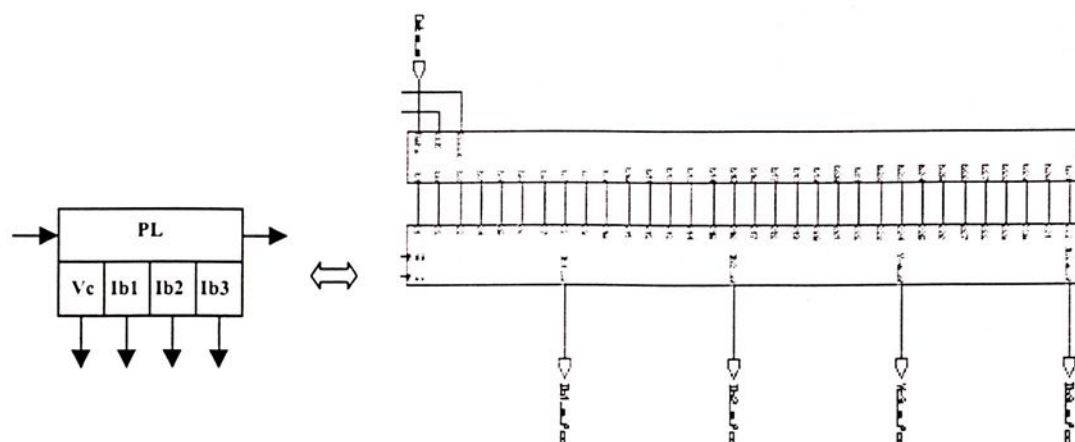


Figura 13: Arquitetura proposta para o circuito controlador de parâmetros.

Como se pode observar, a configuração possui uma entrada serial de dados de controle, que são fornecidos por um programa computacional e através de um registrador de deslocamento, distribuídos nos respectivos circuitos, que serão chaveados conforme as informações obtidas desses dados de controle. Cada fonte recebe um conjunto de 8 bits de controle, totalizando 32 bits para o circuito completo. Em cada fonte, um multiplicador de corrente recebe esses 8 bits e o sinal de saída é o resultado da multiplicação de uma

corrente de referência por uma constante D , onde D é uma palavra digital de 8 bits^[8], que determina o chaveamento adequado do circuito.

5.2) Concepção

Conforme a arquitetura proposta, o circuito é composto por três fontes de corrente, uma fonte de tensão e um registrador de deslocamento. Para que a lógica programável seja equivalente para as quatro fontes, facilitando assim a programabilidade das mesmas, foram utilizados circuitos multiplicadores de corrente, baseados em espelhos de corrente, para fornecer os sinais de controle ao gerador. A figura 14 apresenta o circuito detalhado.

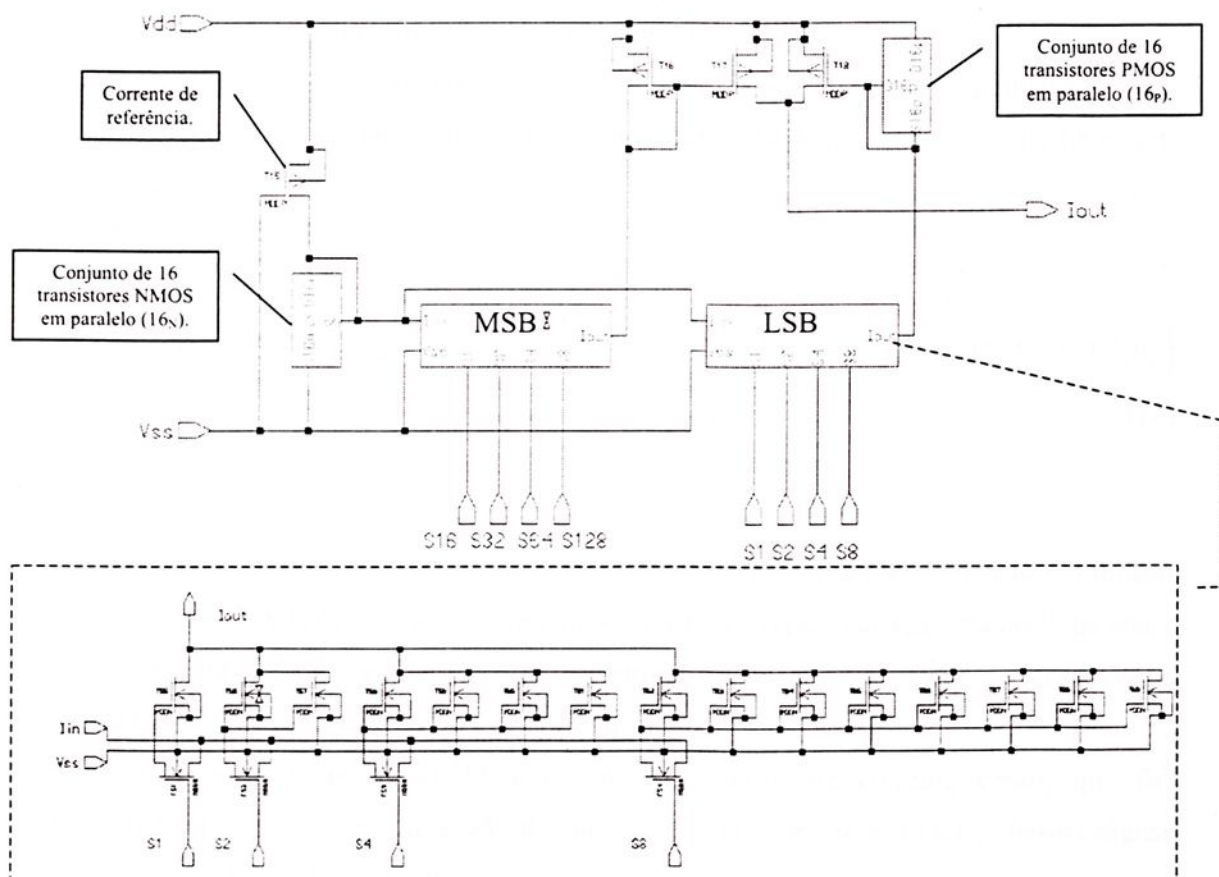


Figura 14: Circuito detalhado do multiplicador de corrente.

Analisando funcionalmente o circuito apresentado na figura 14, pode-se observar que o transistor T_{15} gera uma corrente de referência, que é dividida por 16 através do conjunto 16_N e disponibilizada em um espelho de corrente no qual, dois conjuntos de 1, 2, 4 e 8 transistores em paralelo podem ser habilitados através de chaves (S_1-S_{128}), que permitem ou não a passagem de corrente conforme a palavra digital inserida pelo registrador de deslocamento.

Observa-se que as correntes geradas pelo chaveamento dos transistores, são lançadas separadamente em outros dois espelhos de corrente, sendo que em um deles, a corrente é novamente dividida por 16 através do outro conjunto 16_P para que os 4 bits de controle desse conjunto sejam caracterizados como os menos significativos (*Last Significant Bit - LSB*) e no outro espelho a corrente não é dividida para caracterizá-los como os bits mais significativos (*Most Significant Bit - MSB*).

Em seguida, as correntes dos dois espelhos, que é gerada pelos conjuntos de bits mais significativos e menos significativos, são somadas para fornecer a corrente final dada pela equação 10.

$$I_{OUT} = I_{IN} \times [2^{-1}.b_0 + 2^{-2}.b_1 + 2^{-3}.b_2 + 2^{-4}.b_3 + 2^{-5}.b_4 + 2^{-6}.b_5 + 2^{-7}.b_6 + 2^{-8}.b_7] \quad (10)$$

Com a corrente de saída já definida, e sendo um dos controles realizado em função de uma tensão, torna-se necessária a inclusão de um conversor corrente-tensão^[9], na saída de um dos multiplicadores de corrente, permitindo assim que essa fonte de tensão varie com a mesma lógica das três fontes de corrente.

Apresenta-se na figura 15 um conversor básico de corrente-tensão, que foi implementado para fornecer a tensão de controle V_c , de maneira a seguir a mesma lógica programável das outras fontes, já que esse conversor tem a característica de variar linearmente a tensão de saída em relação à corrente de entrada.

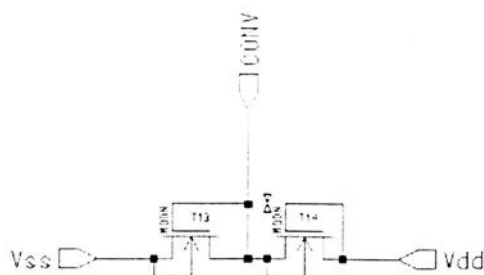


Figura 15: Conversor Corrente-Tensão

O circuito é composto por dois transistores NMOS que operam como um resistor, permitindo que a corrente induzida (I_{in}) no terminal CONV produza uma tensão proporcional (V_{out}) nesse mesmo terminal. Esse comportamento resistivo linear pode ser equacionado pela equação 11.

$$\frac{V_{out}}{I_{in}} = \frac{1}{2K(V_2 - 2V_t)} \quad (11)$$

Dessa forma, concebeu-se então as quatro fontes necessárias ao circuito de controle do gerador de gaussianas. A figura 16 apresenta o circuito completo.

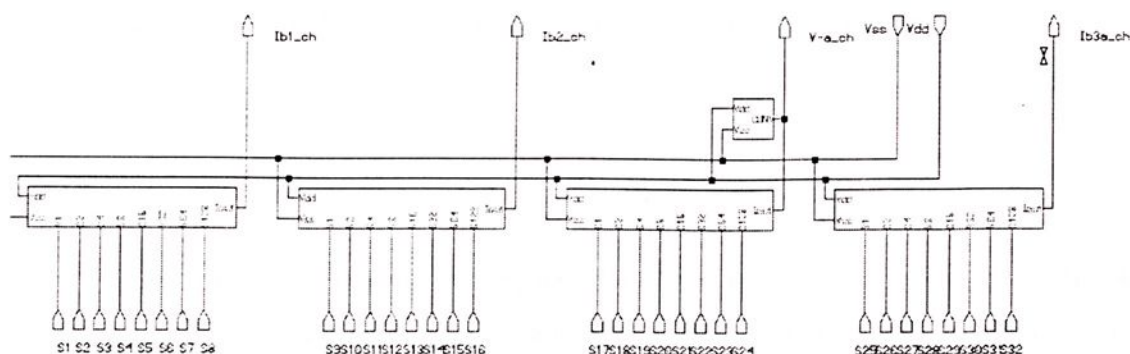


Figura 16: Circuito completo das quatro fontes de controle.

Mas, conforme observação feita anteriormente, cada uma dessas fontes são controladas por uma palavra binária de 8 bits em paralelo, totalizando 32 bits para um bloco completo.

Assim, buscando uma maior facilidade de programação e uma diminuição do número de entradas do circuito, optou-se pela implementação de um registrador de deslocamento mostrado na figura 17, permitindo assim que as quatro palavras binárias de 8 bits fossem programadas de forma serial transformando-se em uma palavra de 32 bits, o que exige uma única entrada e posteriormente é distribuída entre as 32 entradas das fontes de controle.

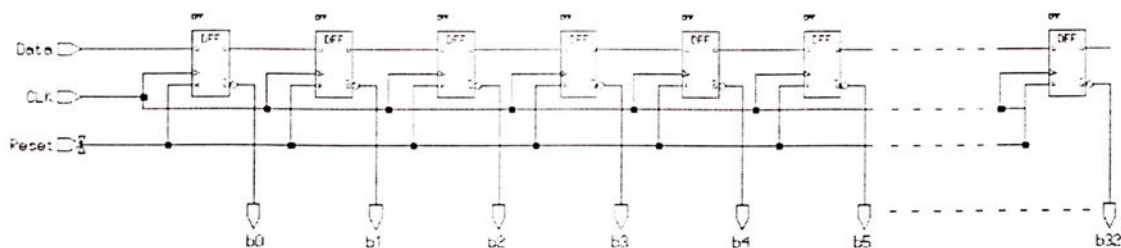


Figura 17: Registrador de deslocamento.

Essa é portanto, a única parte digital do circuito, que foi incluída para facilitar a programabilidade do gerador. Finalizando então a concepção desses blocos, que compõem o sistema de controle, partiu-se assim para dimensionamento dos componentes.

5.3) Análise e Dimensionamento

Em função dos valores de correntes definidos na tabela 1, dimensionou-se os valores dos transistores MOS do multiplicador de corrente com os valores apresentados na tabela 3.

Transistor	T ₁₅	16 _N	MSB	LSB	S ₁ -S ₁₂₈	T ₁₆	T ₁₇	T ₁₈	16 _P
W (μm)	46	10	10	10	2	300	300	20	20
L (μm)	14	10	10	10	2	4	4	4	4

Tabela 3: Valores dos transistores da fonte de corrente.

Vale lembrar que, esse dimensionamento levou em consideração a tensão de alimentação exigida pelo processo de fabricação, que deve estar compreendida em uma faixa de no máximo 5 Volts entre V_{dd} e V_{ss} . Essa consideração é essencial para o dimensionamento dos transistores do conversor corrente-tensão, já que a fonte de tensão controla o centro da gaussiana, que deve ser mantido em 0 Volts (valor central considerando a alimentação de $-2,5$ em V_{ss} e $+2,5$ em V_{dd}), permitindo que sejam anulados prováveis deslocamentos em função de *offset* e/ou casamento de componentes.

Para isso, uma variação de $\pm 1V$ em torno do centro já satisfaz a condição necessária e permite o dimensionamento dos transistores MOS na tabela 4 para que o conversor, através de uma variação de corrente entre 2,5mA e 637,5mA, produza uma variação linear de tensão entre -1V e 1V.

Transistor	T ₁₃	T ₁₄
W (μm)	25	2
L (μm)	10	10

Tabela 4: Valores dos transistores do conversor corrente-tensão.

Em relação ao circuito do registrador de deslocamento, optou-se pela utilização de *Flip-Flop's* tipo D, que apresentam a vantagem de serem sensíveis à borda de subida do relógio. A implementação foi realizada através da utilização de modelos fornecidos e dimensionados pela própria AMS. Após a realização do dimensionamento dos circuitos, partiu-se para a fase de simulações em busca dos resultados desejados.



6) SIMULAÇÃO

Nessa fase do projeto, a simulação de cada circuito dos conjuntos que formam os blocos foi realizada separadamente com testes exaustivos, para que fossem detectados problemas ocasionais de concepção, evitando assim o aparecimento de respostas indesejadas, que por consequência minimizaria as dificuldades a serem superadas após a união desses circuitos para a simulação da rede completa.

6.1) Ambiente de Simulação

As primeiras simulações foram realizadas no ambiente de simulação Spice e as respostas individuais de cada circuito corresponderam satisfatoriamente às expectativas teóricas de concepção de cada bloco, mas essas primeiras simulações serviram apenas de base para a etapa seguinte, que utilizaria uma ferramenta mais confiável e que já considerasse a biblioteca fornecida pelo fabricante (*Design Kit*).

Dessa forma, partiu-se então para uma simulação da rede completa, onde foi utilizada uma ferramenta de simulação da Mentor Graphics, já que a etapa seguinte, de confecção de leiaute para a implementação física, também seria suportada por outra ferramenta do mesmo fornecedor, facilitando assim, a submissão dos arquivos à AMS para a fabricação do circuito.

6.2) Disposição dos Blocos

O esquema mostrado na figura 18, apresenta dois conjuntos completos da rede neural proposta, sendo que um deles possui uma série de pontos internos abertos, visando os testes práticos após a fabricação do circuito integrado e o outro apresenta-se como um bloco fechado, contendo apenas as entradas e saídas necessárias para o seu funcionamento.



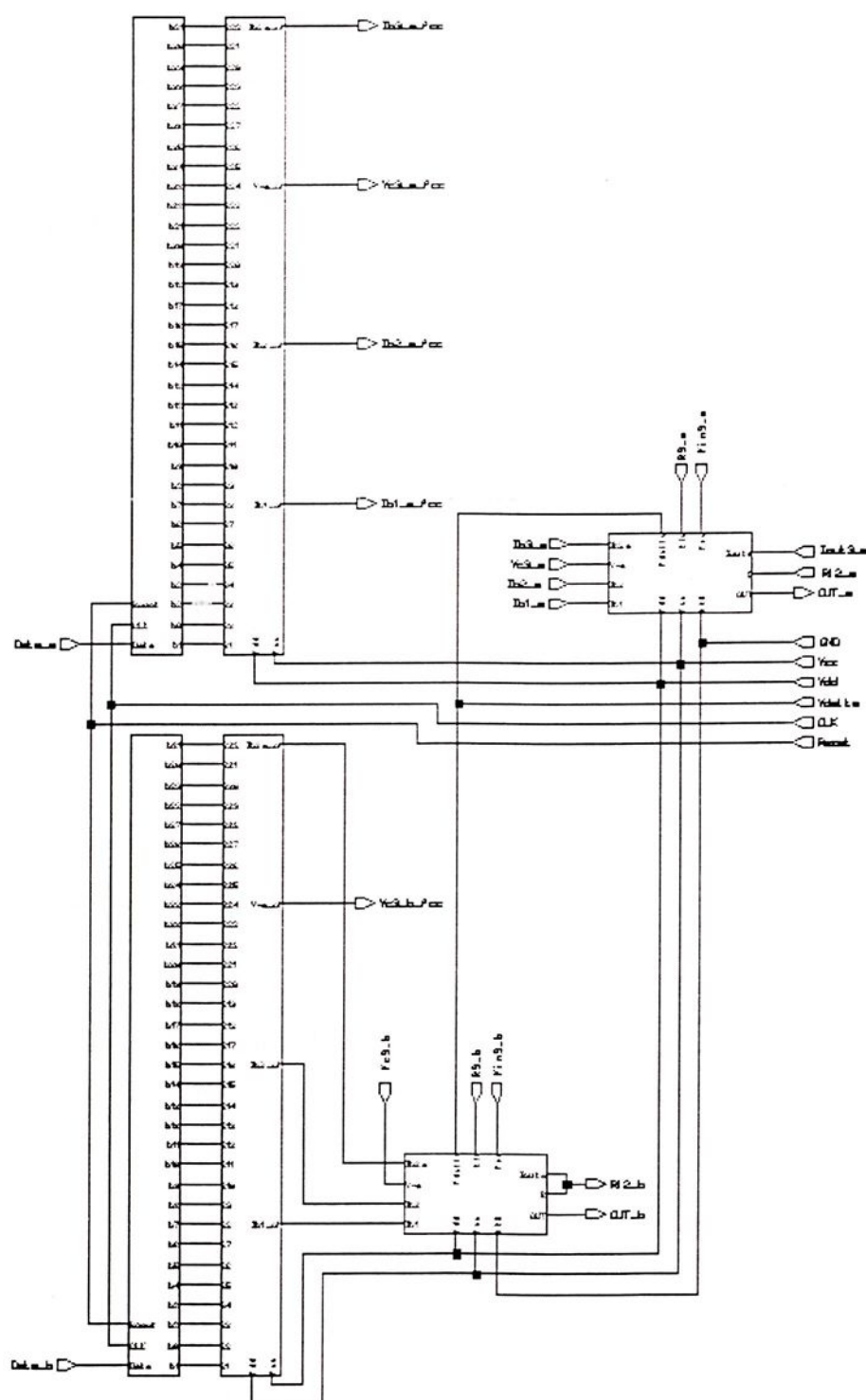


Figura 18: Circuito de teste da rede neural proposta

As simulações foram realizadas procurando obedecer da melhor maneira possível as condições de testes reais, que seriam realizados posteriormente com o circuito fabricado, e para isso o circuito mostrado na figura 18 foi disposto em um formato próximo ao da simulação do circuito real, com todos os resistores externos que complementam o sistema proposto conforme a figura 19.

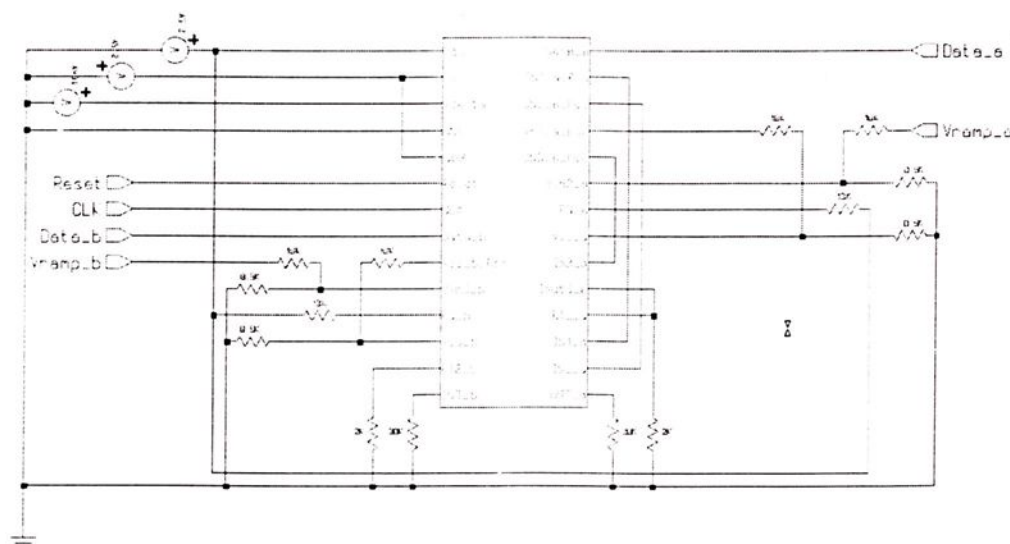


Figura 19: Configuração de teste do CI.

As simulações realizadas buscaram analisar a resposta do circuito em função da variação programada das fontes de corrente que controlam os parâmetros de centro, de largura e de amplitude das gaussianas geradas.

6.3) Análise dos Resultados

Os resultados obtidos durante as simulações, certificaram o comportamento funcional esperado para o circuito, de acordo com a teoria apresentada. A representação gráfica desses resultados é mostrada a seguir.

O primeiro gráfico apresentado na figura 20, mostra em escala a relação da entrada em forma de rampa com a saída caracterizada em forma de gaussiana.

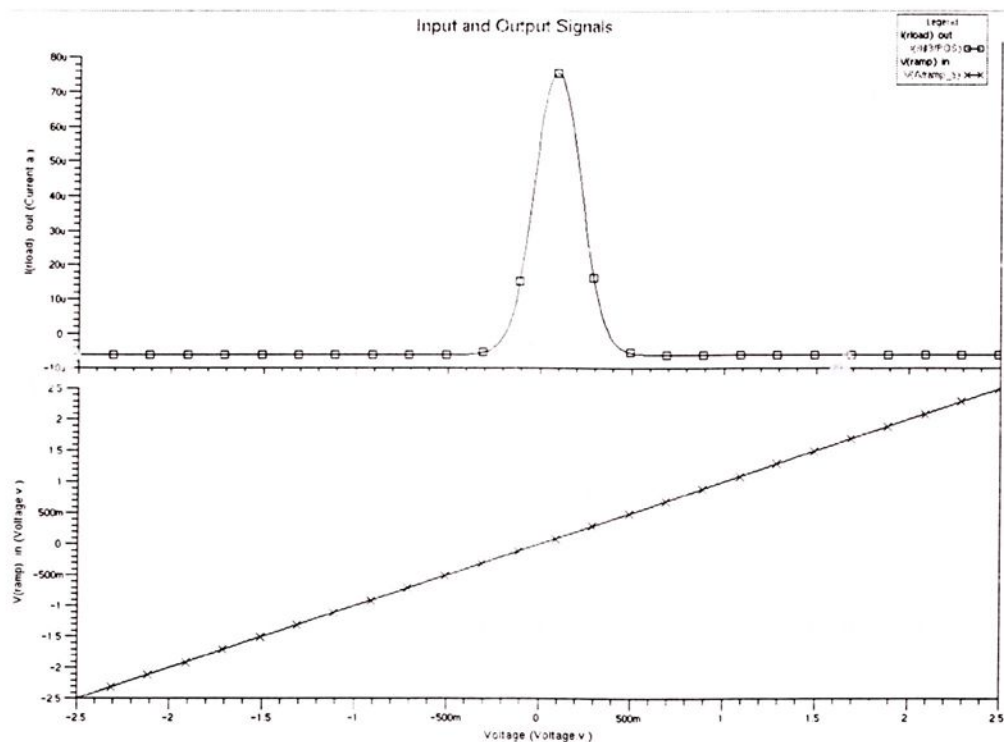


Figura 20: Gráfico da entrada comparada a saída.

O segundo gráfico apresentado na figura 21, foi gerado variando a fonte de tensão entre $-1V$ e $1V$ para analisar os extremos obtidos na variação do centro da gaussiana.

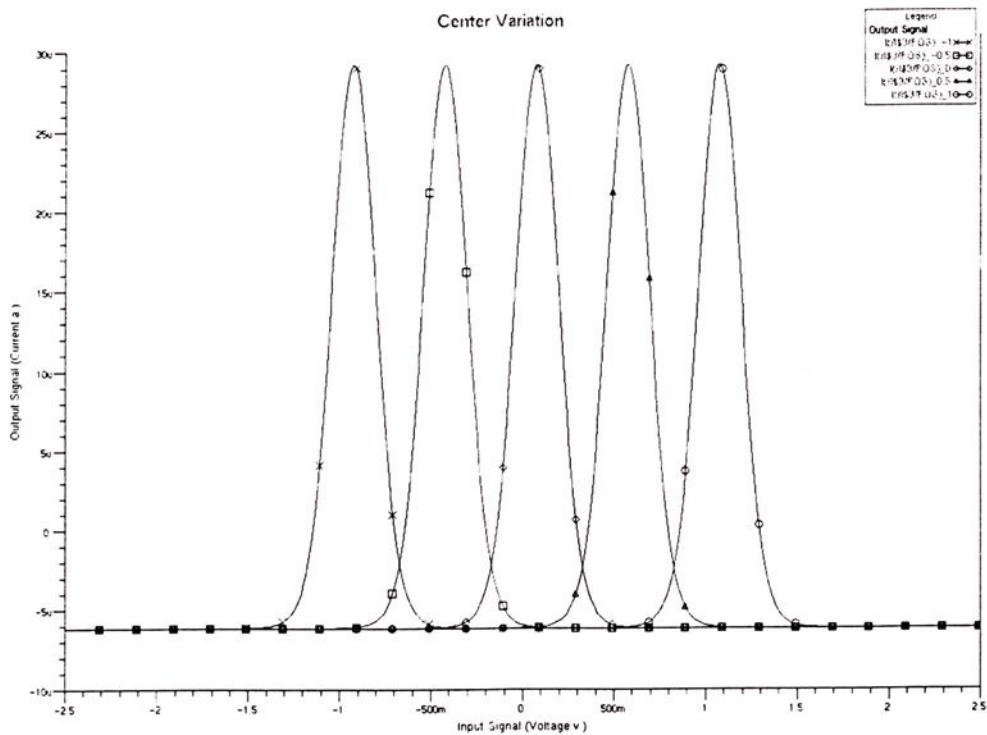


Figura 21: Gráfico da variação do centro da gaussiana.

O terceiro gráfico apresentado na figura 22, foi gerado variando uma fonte de corrente entre $2,5\mu\text{A}$ e $637,5\mu\text{A}$ para analisar os extremos obtidos na variação da amplitude da gaussiana.

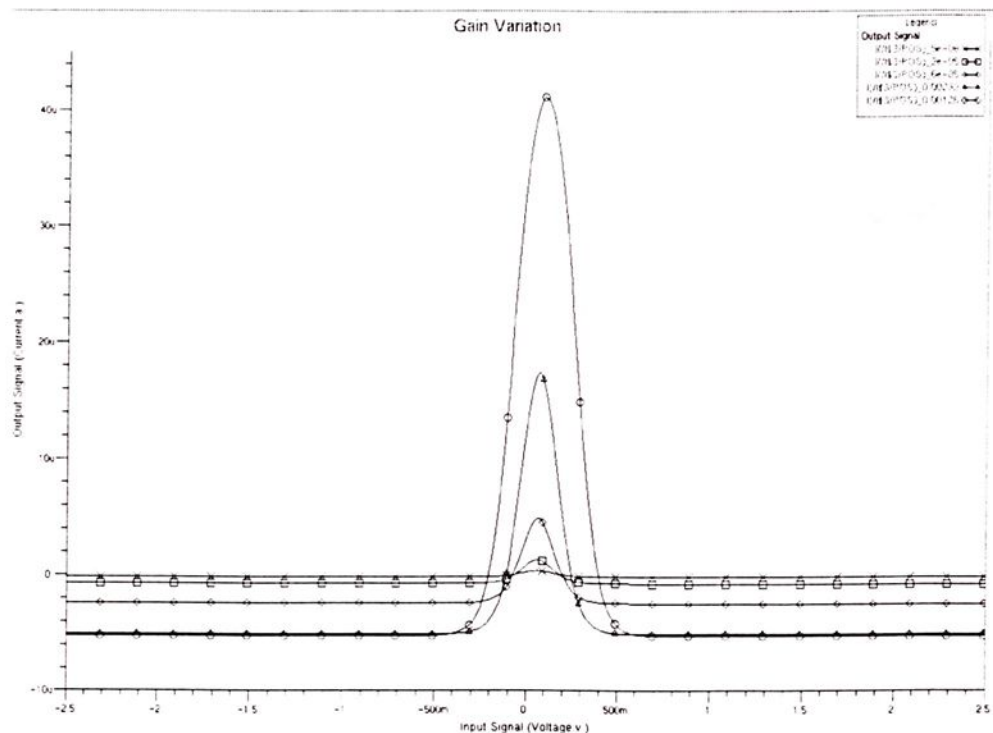


Figura 22: Gráfico da variação da amplitude da gaussiana

O quarto e último gráfico apresentado na figura 23, foi gerado variando outra fonte de corrente entre $2.5\mu\text{A}$ e $637.5\mu\text{A}$ para analisar os extremos obtidos na variação da largura da gaussiana.

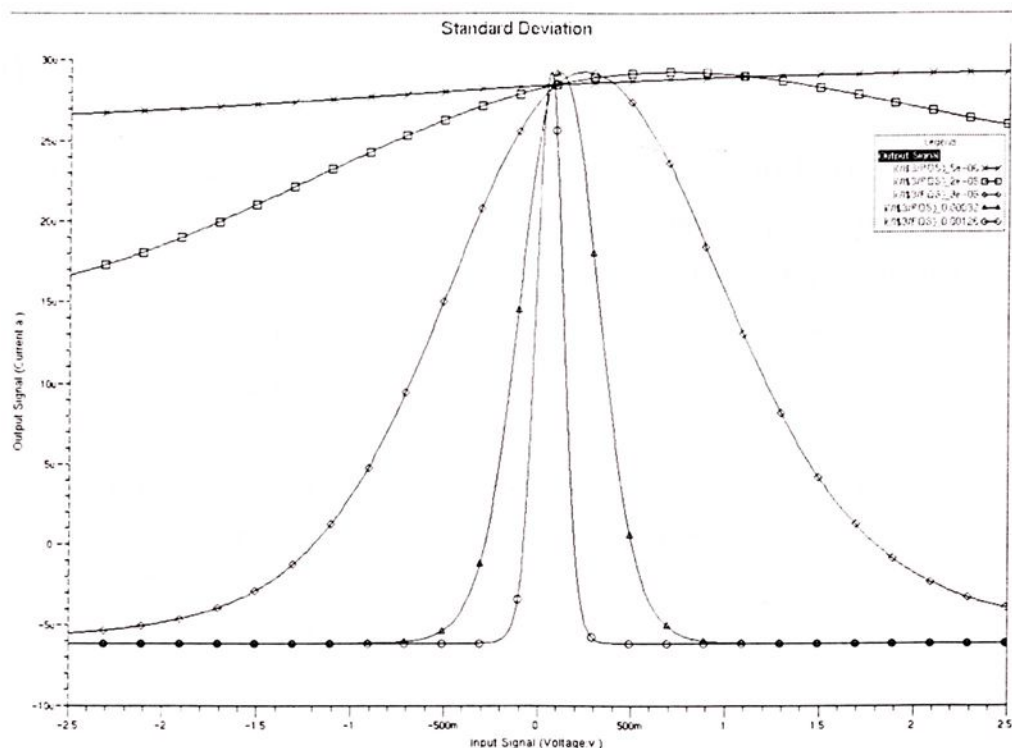


Figura 23: Gráfico da variação da largura da gaussiana

Com os gráficos obtidos, foram comprovadas a funcionalidade da rede neural proposta e a possibilidade de se controlar computacionalmente a resposta dessa camada adaptativa geradora da função gaussiana. Partiu-se então para a etapa de implementação que será descrita a seguir.

7) IMPLEMENTAÇÃO

Com as simulações tendo alcançado resultados satisfatórios, partiu-se para a última etapa antes da fabricação, que é a confecção do leiaute, utilizando outra ferramenta da Mentor Graphics, que permite realizar os testes necessários e gerar os arquivos a serem enviados para a fábrica.

7.1) Leiaute

Da mesma forma que foram realizadas as simulações, os leiautes também foram criados separadamente para cada um dos circuitos, a fim de que os problemas fossem solucionados de maneira mais eficiente e os erros na etapa final fossem minimizados.

Após a finalização do leiaute de cada bloco, foram realizados dois testes de checagem descritos a seguir.

- DRC (*Design Rules Check*): Teste realizado para checar se as regras de projeto estabelecidas para a tecnologia utilizada haviam sido obedecidas.
- LVS (*Layout Versus Schematic*): Teste de comparação realizado para checar a equivalência lógica entre leiaute e esquemático.

Somente após a aprovação nesses testes é que os circuitos foram unidos em um só bloco, formando o chip completo, que também foi submetido a uma checagem final após a inserção das portas (*pads*).

A decisão de não incluir no leiaute os resistores de linearização, foi tomada em busca da diminuição do tamanho efetivo do circuito integrado e o número de *pads* foi limitado em 28, para que os testes reais fossem facilitados em função da utilização da aparelhagem de medição disponibilizada em laboratório.

A seguir apresenta-se o leiaute da rede neural proposta.



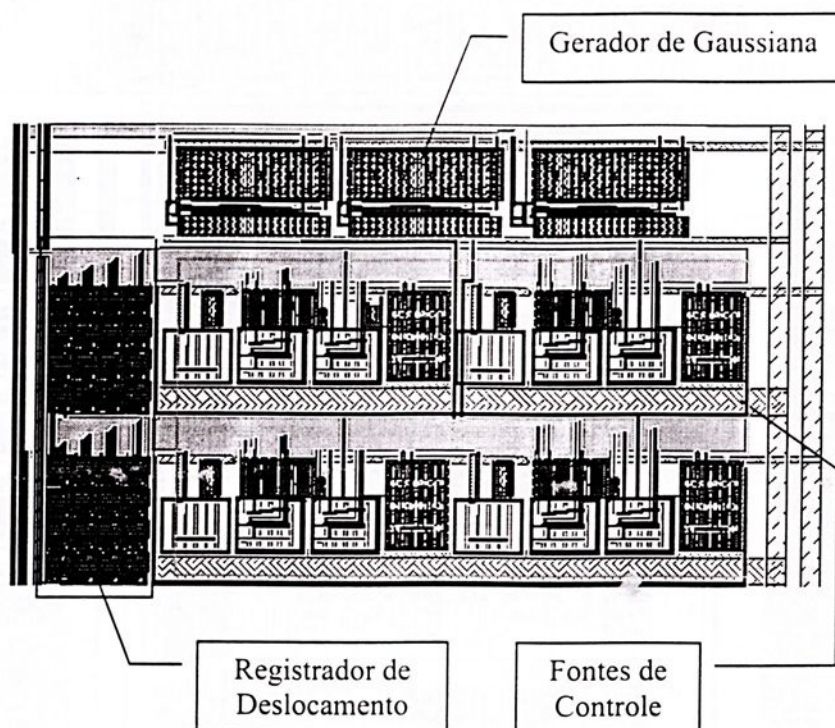


Figura 24: Leiaute da rede completa

Na figura 24, pode-se observar na parte superior os três OTA's que compõem o gerador de gaussianas. Na parte central observa-se as quatro fontes de corrente, onde uma delas possui um conversor corrente-tensão em sua saída, para o controle dos parâmetros da gaussiana, e a direita dessas fontes encontra-se o registrador de deslocamento que permite o controle computacional das fontes de controle.

Para a definição final do leiaute que foi enviado para fabricação, foram inclusos no circuito integrado duas redes completas com suas respectivas entradas e saídas, e ainda algumas aberturas para testes em uma delas. Todas essas interfaces foram distribuídas em 28 portas correspondentes, conforme mostra a figura 25.

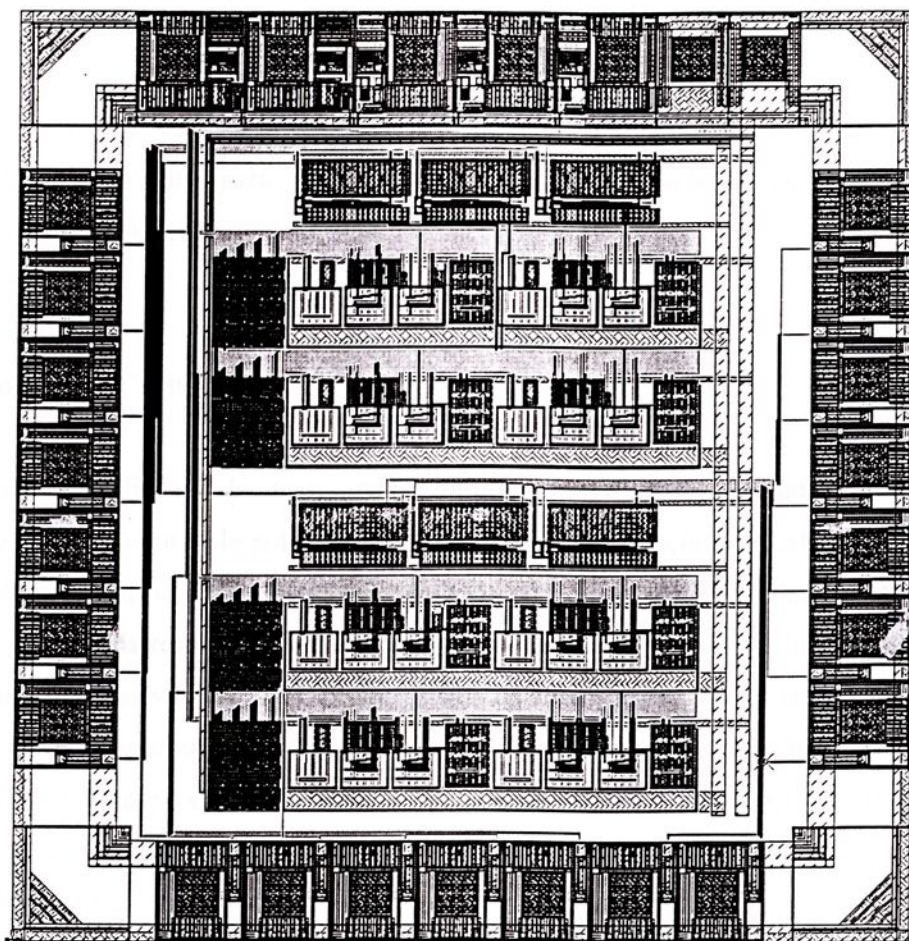


Figura 25: Leiaute do circuito integrado enviado para fabricação

Com a fase de implementação encerrada, aguardou-se o retorno dos protótipos para que se desse início aos testes laboratoriais em busca dos resultados práticos, mostrados no próximo capítulo.

8) TESTES

Nessa etapa serão apresentados os resultados obtidos durante os testes laboratoriais realizados com o chip, para sua caracterização e identificação de possíveis defeitos de projeto ou de fabricação.

8.1) Ambiente de Teste

Para a realização dos testes, os protótipos foram dispostos em um *protoboard*, com os respectivos resistores de polarização, conforme a configuração proposta na figura 19.

Como o princípio de funcionamento do circuito estava relacionado a um controle computacional dos resultados, e esse controle deveria ser realizado de forma dinâmica, um programa foi desenvolvido para enviar os bits de forma serial para a entrada de dados do circuito. Esse programa foi desenvolvido em linguagem Basic (Apêndice1), e utiliza uma porta de saída paralela de um computador como interface para transmitir não só os dados de controle, mas também os sinais de relógio e de *reset*, necessários ao funcionamento da parte digital do circuito.

8.2) Resultados Experimentais

Com todos esses cuidados tomados na preparação do ambiente de teste, e seguindo a mesma metodologia utilizada na fase de simulação, foram obtidas através da utilização de um osciloscópio digital Tektronix TD210, as curvas de resposta do circuito com tempo de aquisição de 10 μ s e 2500 pontos por medida.

As variações observadas em relação aos valores de tensão e corrente em que estão compreendidas as curvas de resposta, devem ser analisadas levando em consideração o fato de que a faixa de tensão de alimentação do circuito estava entre 0V e 5V, diferentemente da simulação onde foi utilizada uma faixa entre -2,5V e +2,5V.



O primeiro gráfico apresentado na figura 26, mostra em escala a relação da entrada em forma de rampa com a saída caracterizada em forma de gaussiana.

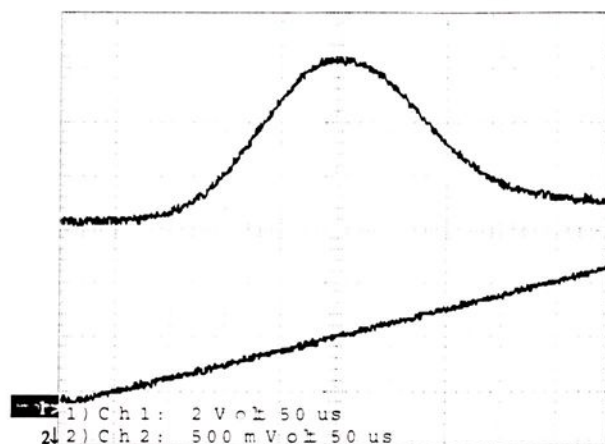


Figura 26: Relação das curvas de entrada e saída do protótipo.

O segundo gráfico apresentado na figura 27, foi gerado variando a fonte de tensão aproximadamente entre 2V e 4.7V para analisar a variação do centro da gaussiana.

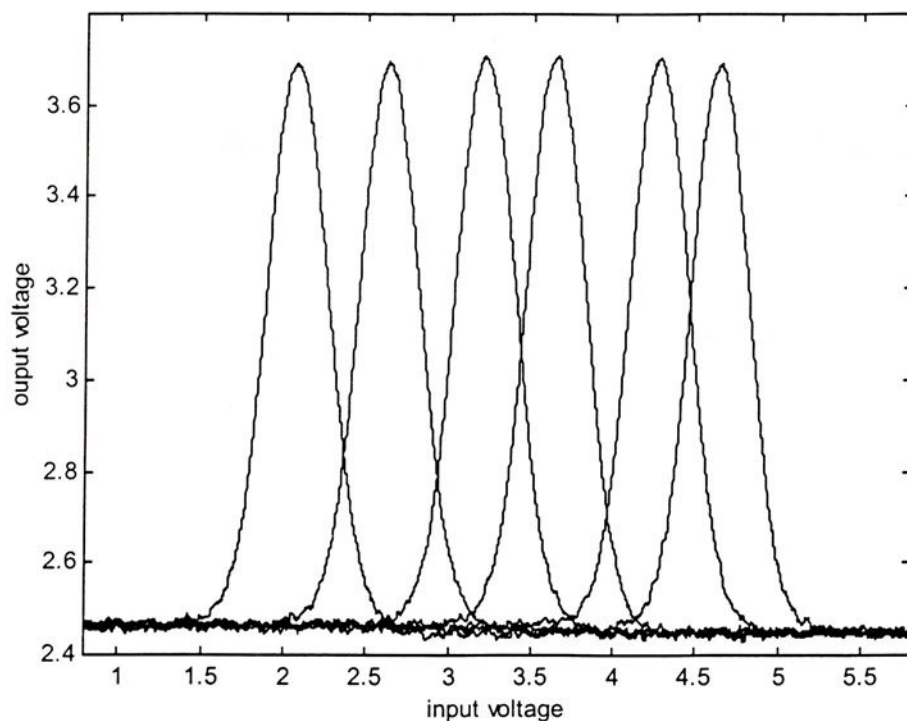


Figura 27: Curva de variação do centro da gaussiana.

O terceiro gráfico apresentado na figura 28, foi gerado variando uma fonte de corrente aproximadamente entre $4\mu\text{A}$ e 1.2mA para analisar os extremos obtidos na variação da amplitude da gaussiana.

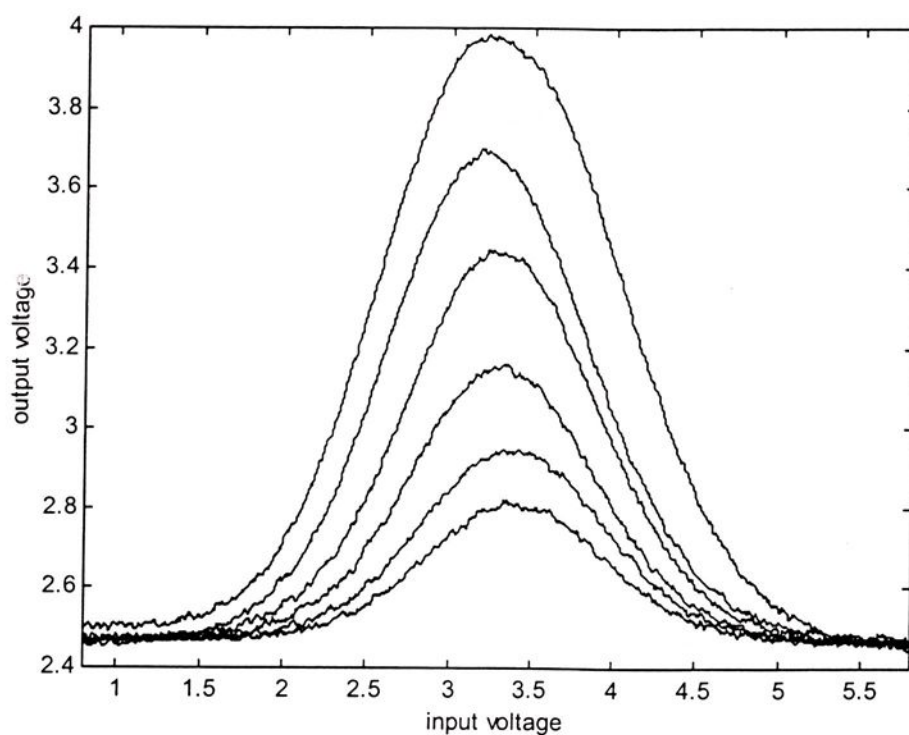


Figura 28: Curva de variação da amplitude da gaussiana

O quarto e último gráfico apresentado na figura 29, foi gerado variando outra fonte de corrente entre $4\mu\text{A}$ e $1,2\text{mA}$ para analisar os extremos obtidos na variação da largura da gaussiana.

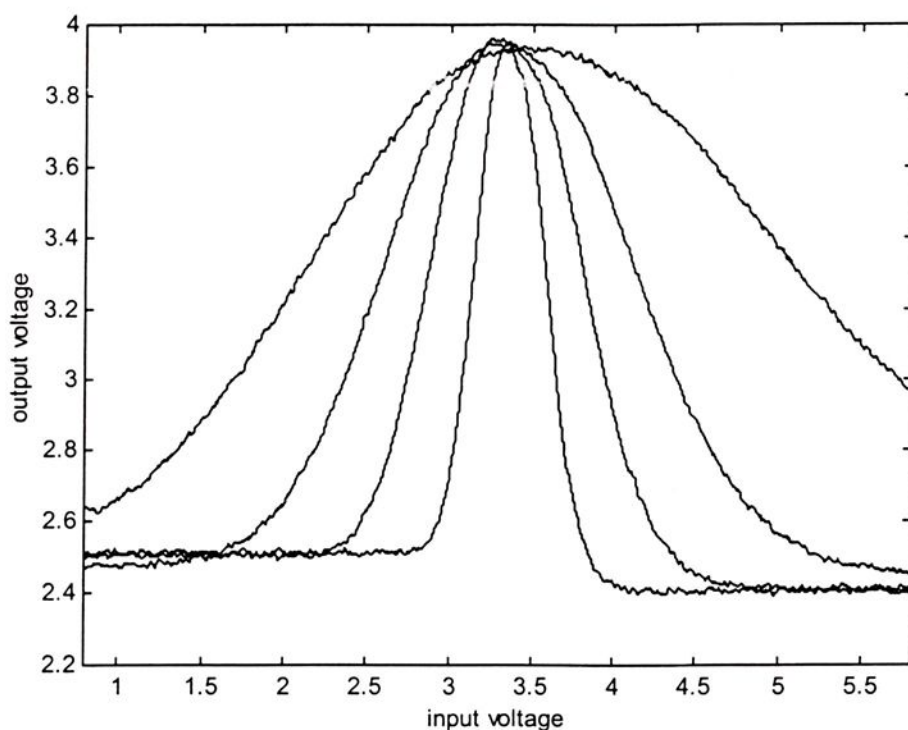


Figura 29: Curva da variação da largura da gaussiana

Os resultados coletados, foram transferidos para o programa Matlab e colocados na forma gráfica conveniente. É importante salientar, que devido a precisão de oito bits do osciloscópio, foi realizada uma média móvel de dez amostras de resultados.

Os resultados obtidos apresentaram variações em relação à faixa de trabalho esperada para o circuito fornecedor de corrente, muito provavelmente devido a idealidade

dos modelos utilizados na concepção e do casamento dos componentes dimensionados em função dessas simulações.



9) CONCLUSÕES:

Passando por todas as etapas descritas anteriormente, foi possível observar a grandiosidade que pode atingir um projeto comercial de um circuito integrado.

Os resultados experimentais comprovaram a viabilidade da proposta de implementação da rede neural de base radial em tecnologia BiCMOS.

Alguns problemas foram identificados durante os testes, como por exemplo um atraso no fechamento da chave de corrente do circuito de controle provavelmente causado por uma capacitância parasita. Essa capacitância pode ser eliminada com a utilização de um modelo de chave mais elaborado, que apresente caminhos para uma corrente de fuga, permitindo assim uma descarga imediata dessa capacitância e o conseqüente chaveamento do circuito.

Outro problema foi a pequena variação na forma de onda da gaussiana, durante a introdução dos bits pelo programa de controle. A simples inserção de latches nas saídas dos registradores eliminaria esse problema, fazendo com que os bits só fossem entregues após o ultimo flip-flop ter recebido o dado esperado.

Mas nenhum desses problemas impediu que os objetivos definidos fossem alcançados. O circuito teve dimensões finais de 4mm^2 e um total de 1056 transistores. Várias idéias novas surgiram durante as etapas realizadas, mostrando que algumas coisas poderiam ser melhoradas, mas em busca de vencermos a grande barreira de alcançarmos a fabricação, optamos por manter as idéias iniciais, que foram as mais simples encontradas para essa realização.

Uma das propostas é a inclusão de um módulo de memória para que a rede não perca as informações cada vez que for desligada. Outra proposta seria a verificação de aplicações desses resultados na solução de problemas reais, ou ainda, estudar a geração dessa gaussiana em um plano bidimensional através da multiplicação das mesmas.

O grande benefício foi a oportunidade de adquirir conhecimentos e vencer desafios, que fazem parte do dia-dia dos projetistas de circuito integrado e nos mostram que temos muito mais para aprender e devemos estar sempre abertos a novas descobertas.



10) BIBLIOGRAFIA

- [1] J. Choi, B. J. Sheu, and J. C-F. Chang, "A Gaussian Synapse Circuit for Analog VLSI Neural Networks", IEEE Transactions on VLSI Systems, vol.2, no.1, pp. 129-133, March 1994.
- [2] J. Madrenas, M. Verleysen, P. Thissen, and J. L. Voz, "A CMOS Analog Circuit for Gaussian Functions", IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, vol.43, no.1, pp. 70-74, January 1996.
- [3] 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, March 1988.
- [4] L. Theogarajan and L. A. Akers, "A Scalable Low Voltage Analog Gaussian Radial Basis Circuit", IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, vol.44, no.11, pp. 977-979, November 1997.
- [5] S. Franco. Design with Operational Amplifiers and Analog Integrated Circuits. 2nd Edition, Boston, McGraw-Hill, 1998.
- [6] M. B. Lucks and N. Oki, "An analog implementation of radial basis function network appropriate for transducer linearizer", 42nd Midwest Symposium on Circuits and Systems, Volume: 2 , 2000 , Page(s): 1109 -1112 vol. 2, Aug. 2000.
- [7] M. B. Lucks and N. Oki, "A radial basis function network (RBFN) for function approximation", 42nd Midwest Symposium on Circuits and Systems, Volume: 2 , 2000 , Page(s): 1099 -1101 vol. 2, Aug. 2000.
- [8] N. Paulino and J. E. Franca, "A CMOS digitally programmable current multiplier" ISCAS – International Symposium of Circuits and Systems, Vol. 1, pp. 254-257, 1996.



- [9] K. Bult and G. J. G. M. Geelen, "An inherently linear and compact MOST-only current division technique", IEEE JSSC, vol. 27, No. 12, pp. 1730-1735, Dec. 1992.
- [10] I. C. Cebikbas et al. "VLSI Implementation of GRBF (Gaussian Radial Basis Function) Networks", ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000 Geneva, Switzerland, pp.III-646-649.
- [11] Melendez-Rodriguez M. and Silva-Martinez J, "A fully-programmable temperature-compensated analogue circuit for gaussian functions", ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000 Geneva, Switzerland, pp. III-481-484.
- [12] L. Yingwei, N. Sundarajan and P. Saratchandran, IEEE Trans. On Neural Networks, 9, 308 (1998).
- [13] E. Gatt, J. Micallef and E. Chilton, in 8th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2001, p.627.
- [14] S. Churcher, A. F. Murray and H. M. Reekei, Electronics Letters, 29, 1603 (1993).
- [15] J. P. Oliveira and N. Oki, 44th Midwest Symposium on Circuits and Systems, p.247.



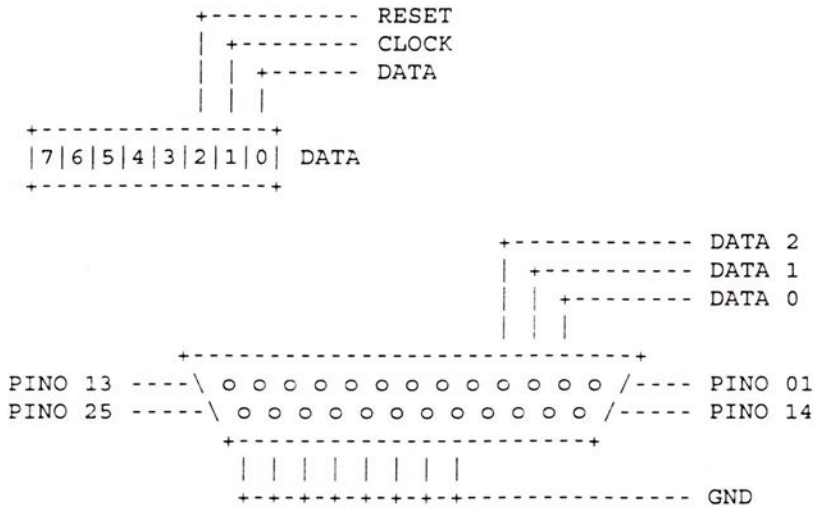
APÊNDICE



Apêndice I

Nesse apêndice, apresenta-se o programa de controle computacional da rede desenvolvido em linguagem Basic.

O byte de saída da porta paralela esta organizado da seguinte maneira:



'Rotina de insercao e envio de dados

INICIO:

CLS

PRINT "PROGRAMA DE CONTROLE DE PARAMETROS DA GAUSSIANA"

GOSUB atraso

GOSUB atraso

PRINT "Enviar dados do Bloco B!"

GOSUB atraso

FOR B = 0 TO 7

COLOR 3

INPUT "Insira os bits da primeira string"; K

V(B) = K

PRINT V(B)

PRINT B

IF K > 1 THEN

V(0) = 1

V(1) = 0

V(2) = 1

V(3) = 0

V(4) = 1

V(5) = 0

V(6) = 1

V(7) = 0

B = 8

END IF

NEXT B

FOR B = 0 TO 7

```
COLOR 13
INPUT "Insira os bits da segunda string"; K
X(B) = K
PRINT X(B)
PRINT B + 8
IF K > 1 THEN
    X(0) = 1
    X(1) = 0
    X(2) = 1
    X(3) = 0
    X(4) = 1
    X(5) = 0
    X(6) = 1
    X(7) = 0
    B = 8
END IF
NEXT B
FOR B = 0 TO 7
    COLOR 14
    INPUT "Insira os bits da terceira string"; K
    Y(B) = K
    PRINT Y(B)
    PRINT B + 16
    IF K > 1 THEN
        Y(0) = 1
        Y(1) = 0
        Y(2) = 1
        Y(3) = 0
        Y(4) = 1
        Y(5) = 0
        Y(6) = 1
        Y(7) = 0
        B = 8
    END IF
NEXT B
FOR B = 0 TO 7
    COLOR 9
    INPUT "Insira os bits da quarta string"; K
    Z(B) = K
    PRINT Z(B)
    PRINT B + 24
    IF K > 1 THEN
        Z(0) = 1
        Z(1) = 0
        Z(2) = 1
        Z(3) = 0
        Z(4) = 1
        Z(5) = 0
        Z(6) = 1
        Z(7) = 0
        B = 8
    END IF
NEXT B
'Estrategia de reset sincrono com duracao de 32 pulsos de clock
COLOR 11
PRINT "INICIANDO RESET"
RESET:
```



```

FOR I = 0 TO 7 'DURACAO DO PULSO DE RESET EM T_CLOCKS
' reset active high
OUT (888), (0 OR 2) '00000110 = 6 = CLK_HIGH / RST_HIGH
' reset active low
GOSUB atraso '1 SECOND T_LOW
' reset active high
OUT (888), (0 OR 0) '00000100 = 4 = CLK_LOW / RST_HIGH
' reset active low
GOSUB atraso 'T_HIGH
NEXT I
PRINT "RESET CONCLUIDO!"
GOSUB atraso
GOSUB atraso
COLOR 12
PRINT "ENVIANDO OS BITS!!!"
'estrutura de envio dos bits e geracao de clock
FOR A = 0 TO 7
PRINT "ENVIANDO BIT: "; A
OUT (888), (4 OR V(A) OR 0) 'dado entregue ANTES da borda positiva
OUT (888), (4 OR V(A) OR 2) 'DATA change @ clock positive edge
GOSUB atraso
OUT (888), (4 OR V(A) OR 0) 'T_LOW of clock cycle
GOSUB atraso
NEXT A
FOR A = 0 TO 7
PRINT "ENVIANDO BIT: "; A + 8
OUT (888), (4 OR V(A) OR 0) 'dado entregue ANTES da borda positiva
OUT (888), (4 OR X(A) OR 2) 'DATA change @ clock positive edge
GOSUB atraso
OUT (888), (4 OR X(A) OR 0) 'T_LOW of clock cycle
GOSUB atraso
NEXT A
FOR A = 0 TO 7
PRINT "ENVIANDO BIT: "; A + 16
OUT (888), (4 OR V(A) OR 0) 'dado entregue ANTES da borda positiva
OUT (888), (4 OR Y(A) OR 2) 'DATA change @ clock positive EDGE
GOSUB atraso
OUT (888), (4 OR Y(A) OR 0) 'T_LOW of clock cycle
GOSUB atraso
NEXT A
FOR A = 0 TO 7
PRINT "ENVIANDO BIT: "; A + 24
OUT (888), (4 OR V(A) OR 0) 'dado entregue ANTES da borda positiva
OUT (888), (4 OR Z(A) OR 2) 'DATA change @ clock positive edge
GOSUB atraso
OUT (888), (4 OR Z(A) OR 0) 'T_LOW of clock cycle
GOSUB atraso
NEXT A
PRINT "Dados do Bloco B enviados com sucesso!"
COLOR 11
PRINT "Enviar dados do bloco A!"
FOR B = 0 TO 7
COLOR 3
INPUT "Insira os bits da primeira string"; K
V(B) = K
PRINT V(B)
PRINT B

```



```
IF K > 1 THEN
    V(0) = 1
    V(1) = 0
    V(2) = 1
    V(3) = 0
    V(4) = 1
    V(5) = 0
    V(6) = 1
    V(7) = 0
    B = 8
END IF
NEXT B
FOR B = 0 TO 7
    COLOR 13
    INPUT "Insira os bits da segunda string"; K
    X(B) = K
    PRINT X(B)
    PRINT B + 8
    IF K > 1 THEN
        X(0) = 1
        X(1) = 0
        X(2) = 1
        X(3) = 0
        X(4) = 1
        X(5) = 0
        X(6) = 1
        X(7) = 0
        B = 8
    END IF
NEXT B
FOR B = 0 TO 7
    COLOR 14
    INPUT "Insira os bits da terceira string"; K
    Y(B) = K
    PRINT Y(B)
    PRINT B + 16
    IF K > 1 THEN
        Y(0) = 1
        Y(1) = 0
        Y(2) = 1
        Y(3) = 0
        Y(4) = 1
        Y(5) = 0
        Y(6) = 1
        Y(7) = 0
        B = 8
    END IF
NEXT B
FOR B = 0 TO 7
    COLOR 9
    INPUT "Insira os bits da quarta string"; K
    Z(B) = K
    PRINT Z(B)
    PRINT B + 24
    IF K > 1 THEN
        Z(0) = 1
        Z(1) = 0
```



```

        Z(2) = 1
        Z(3) = 0
        Z(4) = 1
        Z(5) = 0
        Z(6) = 1
        Z(7) = 0
        B = 8

    END IF
NEXT B
COLOR 12
PRINT "ENVIANDO OS BITS!!!"
'estrutura de envio dos bits e geracao de clock
FOR A = 0 TO 7
    PRINT "ENVIANDO BIT: "; A + 32
    OUT (888), (4 OR V(A) OR 0) 'dado entregue ANTES da borda positiva
    OUT (888), (4 OR V(A) OR 2) 'DATA change @ clock positive edge
    GOSUB atraso
    OUT (888), (4 OR V(A) OR 0) 'T_LOW of clock cycle
    GOSUB atraso
NEXT A
FOR A = 0 TO 7
    PRINT "ENVIANDO BIT: "; A + 40
    OUT (888), (4 OR V(A) OR 0) 'dado entregue ANTES da borda p sitiva
    OUT (888), (4 OR X(A) OR 2) 'DATA change @ clock positive edge
    GOSUB atraso
    OUT (888), (4 OR X(A) OR 0) 'T_LOW of clock cycle
    GOSUB atraso
NEXT A
FOR A = 0 TO 7
    PRINT "ENVIANDO BIT: "; A + 48
    OUT (888), (4 OR V(A) OR 0) 'dado entregue ANTES da borda positiva
    OUT (888), (4 OR Y(A) OR 2) 'DATA change @ clock positive EDGE
    GOSUB atraso
    OUT (888), (4 OR Y(A) OR 0) 'T_LOW of clock cycle
    GOSUB atraso
NEXT A
FOR A = 0 TO 7
    PRINT "ENVIANDO BIT: "; A + 56
    OUT (888), (4 OR V(A) OR 0) 'dado entregue ANTES da borda positiva
    OUT (888), (4 OR Z(A) OR 2) 'DATA change @ clock positive edge
    GOSUB atraso
    OUT (888), (4 OR Z(A) OR 0) 'T_LOW of clock cycle
    GOSUB atraso
NEXT A
PRINT "Dados do Bloco A enviados com sucesso!"
GOSUB atraso
COLOR 15
INPUT "Deseja resetar tudo e reenviar novos dados? (digite y para
continuar)"; W$
IF (W$ = "Y" OR W$ = "y") THEN GOTO INICIO
END
atraso:
FOR H = 0 TO 50000
NEXT H
RETURN

```



Apêndice 2

Nesse apêndice, apresenta-se as microfotografias do chip e seus respectivos blocos.

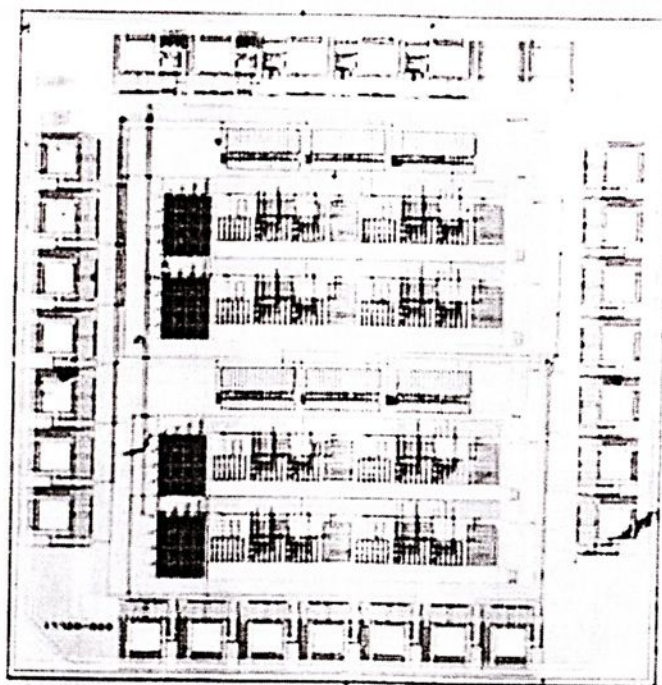


Figura 30: Chip com dois sintetizadores completos implementados

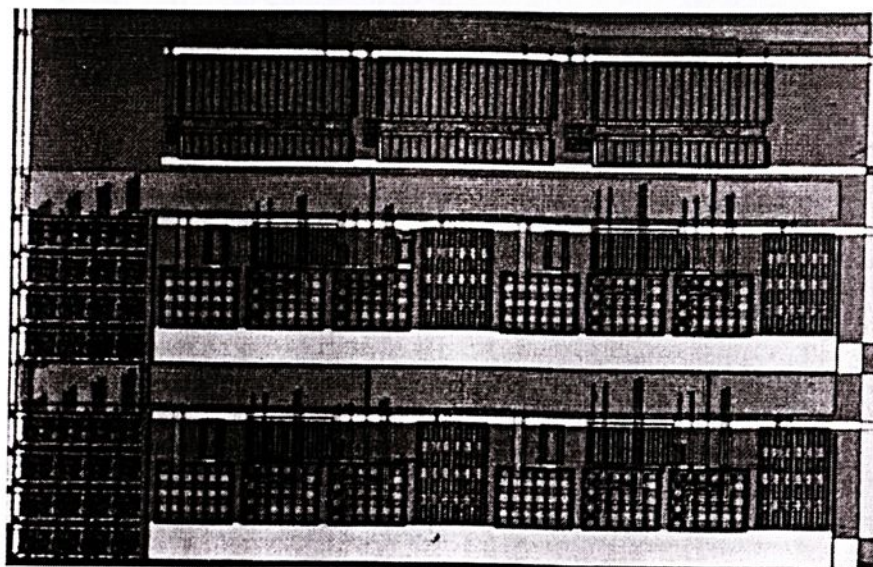


Figura 31: Circuito completo do sintetizador

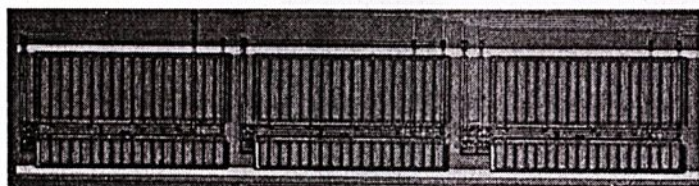


Figura 32: Bloco gerador da função gaussiana

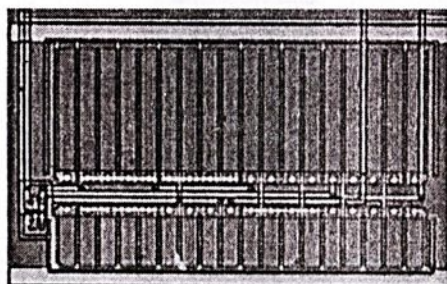


Figura 33: Amplificador Operacional de Transcondutância OTA

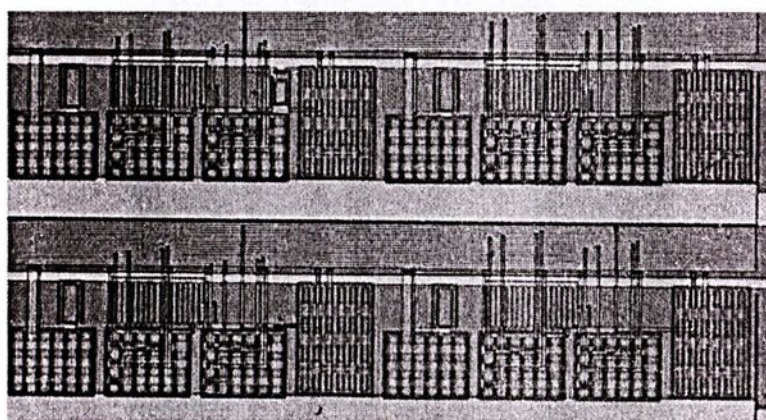


Figura 34: Bloco controlador de parâmetros

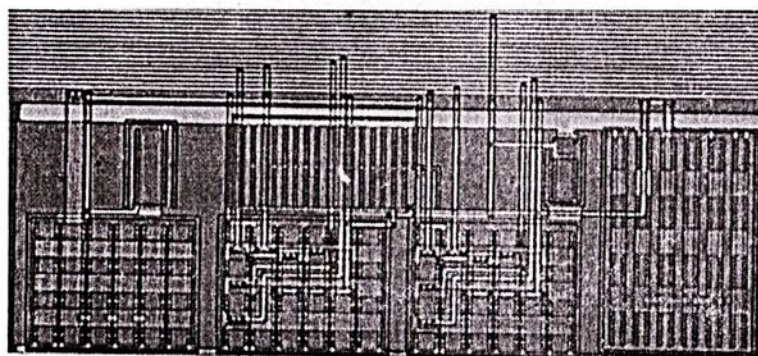


Figura 35: Fonte de corrente de controle

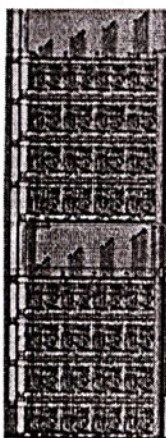


Figura 36: Registrador de deslocamento

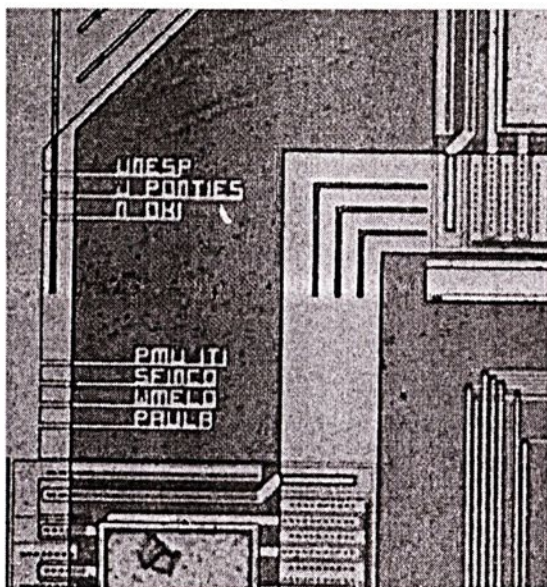


Figura 37: Identificação dos realizadores.



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Câmpus de Ilha Solteira
Programa de Pós-Graduação em Engenharia Elétrica
Av. Brasil Centro, 56
15385-000 Ilha Solteira - SP
www.dee.feis.unesp.br

