Proposta de Implementação em Hardware dedicado de Redes Neurais Competitivas com Técnicas de Circuitos Integrados Analógicos

Autores

Prof. Dr. Paulo M. Engel – Universidade Federal do Rio Grande do Sul UFRGS/Brasil
e-mail: engel@inf.ufrgs.br

Prof. Ms. Rolf Fredi Molz – Universidade de Santa Cruz do Sul UNISC/Brasil
e-mail: rolf@dinf.unisc.br

Resumo

Neste trabalho apresenta-se uma proposta de uma técnica para implementação em *hardware*, das estruturas básicas de uma Rede Neural Competitiva, baseada em técnicas analógicas.

Através desta proposta, será abordada uma das classes mais interessantes de Redes Neurais Artificiais (RNA) que são as Redes Neurais Competitivas (RNC), que possuem forte inspiração biológica. As equações fundamentais que descrevem o comportamento da RNC foram derivadas de estudos interdisciplinares, a maioria envolvendo observações neurofisiológicas. O estudo do neurônio biológico, por exemplo, nos leva à clássica equação da membrana.

A técnica mostrada para a implementação das Redes Neurais Competitivas se baseia no uso das técnicas analógicas. Estas conduzem a um projeto mais compacto além de permitirem um processamento em tempo real, visto que o circuito computacional analógico altera simultaneamente e continuamente todos os estados dos neurônios que se encontram interligados em paralelo.

Para esta proposta de implementação, é mostrado que as equações fundamentais que governam as Redes Neurais Competitivas possuem uma relação com componentes eletrônicos básicos, podendo então, serem implementados através destes simples componentes com os quais as equações fundamentais se relacionam.

Para tanto, é mostrado por meio de simulações em software, o comportamento das equações fundamentais deste tipo de Redes Neurais, e então, é comparado este comportamento, com os obtidos através de simulações elétricas dos circuitos equivalentes oriundos destas equações fundamentais. Mostra-se também, em ambas as simulações, uma das características mais importantes existentes nos modelos de RNC, conhecida como Memória de Tempo Curto (STM).

Por fim, é apresentada uma aplicação típica na área de clusterização de padrões utilizando pesos sinápticos, a fim de, demonstrar a implementação utilizando as técnicas descritas durante o trabalho. Esta aplicação é demonstrada através de uma simulação elétrica, sendo esta realizada por meio do simulador HSPICE. Tal aplicação demonstra o correto desempenho da proposta deste trabalho.

Palavras-Chaves: Redes neurais artificiais, Redes neurais competitivas, Implementação em *hardware*, Técnicas analógicas.

“Proposta de Implementação em Hardware dedicado de Redes Neurais Competitivas com Técnicas de Circuitos Integrados Analógicos”

1. Introdução

Com o desenvolvimento da área de redes neurais artificiais, acentuou-se a necessidade da realização de circuitos integrados neurais, e não apenas a simulação dessas redes por meio de um programa. As principais razões para tal necessidade são: o aumento da capacidade de processamento; o desenvolvimento de arquiteturas adequadas para o processamento neural; e uma maior facilidade para o desenvolvimento de equipamentos comerciais com a redução de suas dimensões, de seus custos e de um aumento de confiabilidade dos mesmos.

A implementação de redes neurais artificiais em VLSI tira vantagens do inerente paralelismo para se obter soluções rápidas. Nas implementações em VLSI de redes neurais, técnicas analógicas são preferíveis, por essas conduzirem a um projeto mais compacto e permitirem o processamento em tempo real [1]. Esta velocidade, necessária para processamento em tempo real de informações, pode ser fornecida por um circuito computacional analógico porque todos os neurônios alteram simultaneamente e continuamente seus estados analógicos em paralelo. Quando comparado a modernos computadores digitais de propósito geral, construídos com circuitos convencionais, os circuitos neurais computacionais possuem diferenças marcantes quanto as características e à organização [1]. Cada porta lógica, tipicamente, obterá entradas de dois ou três outros neurônios, e um grande número de decisões binárias independentes são feitas no curso de uma computação digital. Em contrapartida, cada processador (neurônio) neural não linear em uma rede computacional analógica adquire entradas de dez ou centenas de outros neurônios e uma solução coletiva é computada na base de interações simultâneas de centenas de dispositivos.

Uma das classes mais interessantes de Redes Neurais Artificiais (RNA) são as Redes Neurais Competitivas (RNC), que possuem forte inspiração biológica. A rede pode ser composta por uma arquitetura multicamada, ou a competição entre os neurônios pode ser implementada em somente uma camada competitiva. Na camada competitiva cada unidade recebe um sinal de realimentação positivo, sinal excitatório, de si mesma, e envia um sinal inibitório para todas as outras unidades pertencentes a camada competitiva. Se algum padrão é apresentado a camada competitiva somente um neurônio terá a atividade mais intensa. A atividade deste vencedor inibirá a ativação dos outros neurônios por meio das conexões inibitórias. Mesmo se o padrão for removido da entrada, a atividade na saída da camada competitiva permanece. Esta propriedade existente nas RNC é conhecida como *Short Term Memory* (STM).

Pode-se selecionar os padrões a serem reconhecidos na camada competitiva através de pesos sinápticos adaptativos existentes nas entradas dos neurônios pertencentes a camada competitiva. Estes pesos sinápticos implementam o que é conhecido como *Long Term Memory* (LTM).

As equações fundamentais que descrevem o comportamento das RNC foram derivados de estudos interdisciplinares, muito envolvendo observações neurofisiológicas [2]. O estudo do neurônio biológico, por exemplo, conduz a clássica equação da membrana [3].

O objetivo deste artigo é propor uma nova técnica para a implementação em *hardware*, das estruturas básicas de uma rede neural artificial, mais específicamente uma rede neural do tipo competitiva, baseada em técnicas de circuitos integrados analógicos. Para tanto, serão utilizados circuitos básicos para a construção de redes neurais artificiais. Será mostrado que é possível desenvolver um modelo em hardware partindo dos modelos teóricos das diferentes estruturas de uma RNC.

2. Fundamentação Teórica

Nesta seção serão revisados os conceitos mais importantes envolvendo as redes neurais competitivas. As expressões que descrevem as funções das diferentes estruturas das redes neurais competitivas são apresentadas nas próximas seções e estas serão utilizadas como o modelo teórico desejado para a implementação em *hardware*.

2.1. Neurônio Artificial

O componente fundamental de uma Rede Neural Artificial (RNA) é o neurônio ou elemento de processamento (PE). Para simulações de RNA pode-se utilizar simples neurônios discretos, como o modelo de McCulloch-Pitts [4]. A Rede Neural Competitiva (RNC) é um sistema intrinsicamente dinâmico não linear contínuo. O modelo de McCulloch-Pitts não representa propriamente as características desejadas de um neurônio contínuo não linear nestes sistemas. A função de um neurônio para redes contínuas não lineares são modelados diretamente da equação da membrana dos neurônios biológicos. A equação da membrana que descreve a voltagem *V(t)* de um neurônio é dada pela lei:

$$C\frac{\partial V}{\partial t} = (V^{+} - V)g^{+} + (V^{-} - V)g^{-} + (V^{p} - V)g^{p} \qquad (1)$$

onde C é a capacitância; V^{+}, V^{-} e V^{p} são constantes excitatórias, inibitórias e pontos de saturação passivos; e g^{+}, g^{-} e g^{p} são respectivamente condutâncias excitatórias, inibitórias e passivas. Dependendo da função de um neurônio em uma camada particular, este pode ser modelado por uma expressão derivada da equação (1). Portanto, será analisado separadamente, a modelagem de um neurônio em cada camada da RNC.

2.2. Redes Shunting Feedforward

A camada de entrada de uma RNC forma uma rede competitiva shunting feedforward (SFFN). Estas redes possuem propriedades de controle de ganho automático, capazes de solucionar o dilema ruído-saturação dos padrões de entrada [2]. Através de conexões centrais, excitatórias, e laterais, inibitórias, os sinais de saída da rede representam uma normalização dos sinais de entrada. Estas propriedades surgem das interconexões *on-center off-surround* dos padrões que são aplicados à entrada da SFFN. A Figura (1a) ilustra a arquitetura de uma SFFN com 2 entradas (I_i). A conexão entre um componente da entrada e o respectivo neurônio é excitatória (+). As conexões aos outros neurônios são inibitórias (-).

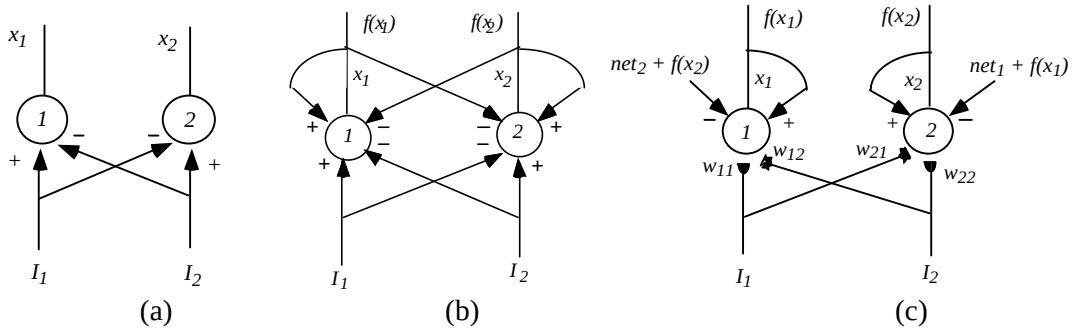


Figura 1: (a) Arquitetura de uma SFFN, (b) de uma SFBN e (c) de uma SFBN com pesos sinápticos.

O comportamento de uma SFFN pode ser descrito por uma equação STM derivada da equação da membrana:

$$\frac{d}{dt} x_i = - Ax_i + (B - x_i) I_i - x_i \sum_{k \neq i} I_k \quad (2)$$

Na equação (2), *A* e *B* são parâmetros que, respectivamente, governam o ganho geral e a taxa de decaimento do sistema. x_i representa a ativação do neurônio, antes de se aplicar a função de ativação.

2.3. Redes Shunting Feedback

A camada competitiva da RNC forma uma rede shunting feedback (SFBN). A SFBN possui importantes propriedades de memória associativa. Estas propriedades surgem das conexões *on-center off-surround* que se realimentam. A Figura (1b) representa esquematicamente a arquitetura básica de uma SFBN com dois neurônios. As conexões que se realimentam são responsáveis pelo comportamento dinâmico da SFBN. Uma vez a rede estabilizada em um estado estável, o sinal aplicado à entrada pode ser removido, e o padrão de ativação na saída da rede permanece ativo. Esta propriedade é muito desejável, e é relacionada aos traços STM destas redes. A equação fundamental desta rede é:

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)[f(x_i) + I_i] - x_i[\Sigma_{k \neq i} f(x_k) + J_i] \quad (3)$$

Em uma RNC completa, as conexões de entrada em um neurônio na camada competitiva passam através de pesos sinápticos, como mostrado na Figura (1c). Nesta rede o valor *net* na entrada dos neurônios são calculados de acordo com a usual soma de produtos. Os valores de saída dos neurônios são calculados pela aplicação da função de ativação aos valores de x_i. O valor de saída de um neurônio é realimentado a si próprio, por meio das conexões excitatórias e através de conexões inibitórias dos demais neurônios. Todas estas conexões realimentadas transformam os padrões de entrada e armazenam os padrões transformados em STM, como ativações dos neurônios. A equação fundamental desta rede é:

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)[f(x_i) + net_i] - x_i[\Sigma_{k \neq i} f(x_k) + \Sigma_{k \neq i} net_k] \quad (4)$$

3. Implementação da RNC por meio de um hardware analógico

Como comentado na introdução deste artigo, reconhecemos que as equações teóricas que descrevem o comportamento das diferentes camadas de uma RNC podem ser implementadas utilizando-se componentes eletrônicos, compondo um circuito integrado analógico. Os componentes do hardware e estruturas da RNC foram projetados em um modo que, suas equações de corrente ou voltagem são equivalentes as expressões que descrevem as equações teóricas abordadas anteriormente. Apresenta-se abaixo, os componentes e a estrutura envolvida nas RNC.

3.1. Implementação da rede Shunting Feedforward

O componente fundamental para todas as redes contínuas não lineares é um neurônio analógico. A exata expressão que representa a função de um neurônio depende especificamente da arquitetura da rede. Nas redes Shunting Feedforward, os neurônios realizam a soma de um sinal de entrada excitatório e de vários sinais de entrada inibitórios. A equação (2) expressa esta operação. Uma equação equivalente pode ser obtida com um circuito RC utilizando transistores MOS operando na região linear. A Figura 2 representa um diagrama de um circuito que implementa o neurônio *1* e as interconexões de uma SFFN com duas entradas e dois neurônios, como representado na Figura 1a.

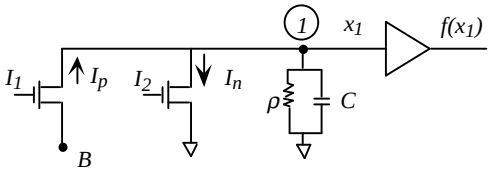


Figura 2: Implementação em hardware de um simples neurônio em uma SFFN.

Nesta figura, Figura 2, I_p é uma corrente fluindo pelo transistor MOS cujo dreno está conectado a uma fonte de voltagem *B*. O transistor está operando na região linear e o sinal de entrada excitatório, I_1 , é aplicado ao terminal do gatilho. O sinal de entrada inibitório, I_2, é aplicado ao transistor MOS que está conectado entre o ponto *1* e o terra comum. Este transistor também opera na região linear. A Figura 2 representa o caso com somente uma conexão inibitória. Os componentes ρ e *C* determinam o parâmetro de atraso para a evolução temporal da voltagem do ponto *1*, x_1. Esta voltagem é o sinal de entrada de um amplificador operacional. A função de transferência do amplificador modela uma função sigmoidal de alto ganho. A equação da corrente para o ponto *1* é:

$$i_C = I_p - I_n - i_r \tag{5}$$

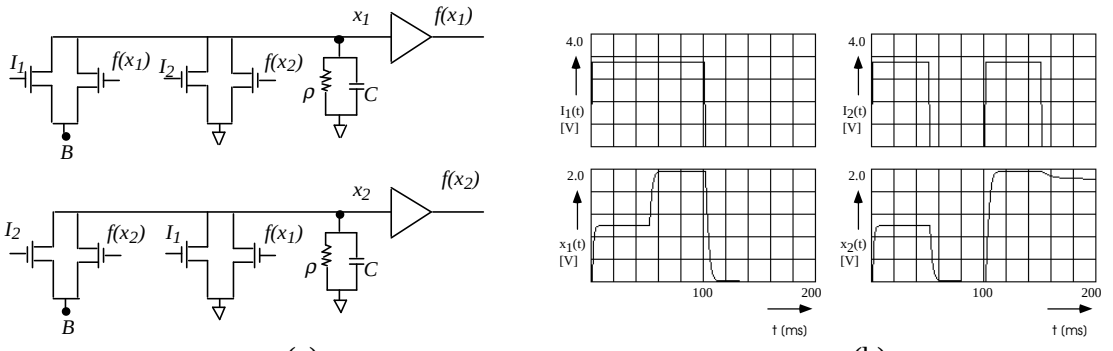
Escrevendo novamente a equação (5) em função da voltagem no ponto x_1 , obtêm-se:

$$\frac{d}{dt} x_1 = - Ax_1 + (B - x_1) G_p - x_1 G_n \tag{6}$$

Nesta equação, G_p e G_n representam os canais de condutâncias na região linear dos transistores que estão realizando as ligações excitatórias e inibitórias, divididas pela capacitância *C*. *A* é o parâmetro de atraso *RC*. Uma SFFN com *n* entradas necessita de *n-1* transistores inibitórios em paralelo. Cada transistor MOS operando na região linear possui uma canal de condutância cujo valor depende linearmente do sinal de entrada (I_i) aplicado ao terminal de gatilho.

3.2. Implementação da rede Shunting Feedback

Para a rede Shunting Feedback deve-se implementar as conexões que se realimentam. Isto é feito essencialmente do mesmo modo que para a SFFN. Para se estudar as características da SFBN implementada por meio de circuitos analógicos, simulou-se uma rede com dois neurônios. A Figura 3 representa uma SFBN com duas entradas. O circuito possui dois transistores excitatórios, um para cada condutância, correspondendo a I_i e $f(x_i)$. Para uma rede de dois neurônios e duas entradas, o circuito possui dois transistores inibitórios, um para cada condutância, correspondendo a I_j e $f(x_j)$. A Figura 3a mostra este circuito.



(a) (b)

Figura 3: Diagrama de um circuito elétrico para uma SFBN com 2 entradas e 2 neurônios (a) e resultados da simulação (b) .

A Figura 3b apresenta os resultados obtidos das simulações deste circuito, com $\rho C = 1ms$, e $B = +2V$. Nesta simulação um sinal de $+3V$ foi aplicado em ambas as entradas da rede durante os primeiros $50\ ms$. Nos próximos $50ms$ foi aplicado um sinal de 0V na segunda entrada. Entre 1$00ms$ e $150ms$ os sinais de entrada foram trocados: a entrada do primeiro neurônio foi colocada em 0V e a entrada do segundo neurônio foi fixado em $+3V$. Após $150ms$ ambos os sinais de entrada foram colocados em 0V. Desta figura pode-se observar que a rede SFBN mantém os traços STM mesmo quando os sinais de entrada são colocados em 0V.

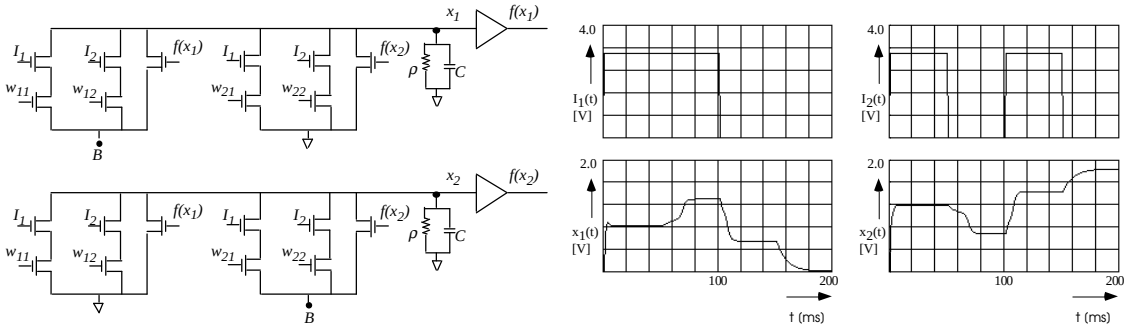
3.3. Implementação da rede Shunting Feedback com pesos sinápticos

Para aplicações práticas da rede Shunting Feedback é necessário fornecer pesos às conexões entre o sinal de entrada e os neurônios. Os pesos sinápticos podem ser implementados como transistores MOS na região linear ligados em série com as condutâncias dos transistores de entrada. Similarmente como G_p e G_n, estes pesos podem ser excitatórios ou inibitórios, dependendo se os transistores estão em série a G_p ou G_n. A Figura 4a representa a implementação em hardware da SFBN com pesos sinápticos cuja arquitetura é mostrada na figura 1c.

Como um exemplo, a condutância equivalente destes circuitos em série da figura 4a, envolvendo os transistores de entrada de sinal I_1 e o correspondente transistor do peso sináptico w_{11} é calculado como:

$$Gp1 = \frac{GI1 \bullet Gw11}{GI1 + Gw11} \qquad (7)$$

onde $Gp1$ é a condutância equivalente da conexão em série da condutância G_{I1}, do transistor conectado a I_1, e da condutância G_{w11} do transistor conectado a w_{11}. A associação em série das condutâncias realizam uma soma ponderada modificada para o termo *net*. A grande vantagem desta técnica é a simplicidade de sua implementação em hardware. Com uma adequada relação entre valores de condutância, pode-se demonstrar que o comportamento geral da rede é similar ao obtido com um valor de *net* calculado com pura soma de produtos. Como uma ilustração, a Figura 4b mostra os resultados obtidos para uma simulação de uma SFBN com a associação em série de condutâncias.



(a) (b)

Figura 4: Diagrama de uma SFBN com pesos sinápticos (a) e resultados simulados (b).

4. Aplicação de uma RNC para clusterização de padrões

Nesta seção, demonstra-se uma aplicação de RNC no campo de clusterização de padrões utilizando-se uma rede SFBN com pesos sinápticos. A Rede Neural Competitiva utilizada para esta aplicação é composta de dezesseis sinais de entrada e dois neurônios na camada competitiva. Para os sinais de entrada, pode-se aplicar padrões em uma forma matricial de 4x4. Na Figura 5a, observa-se o modelo neural utilizado para esta aplicação. A Figura 5b mostra os

padrões de entrada e as correspondentes saídas desejadas. A Figura 5c mostra a circuito implementado para simulação.

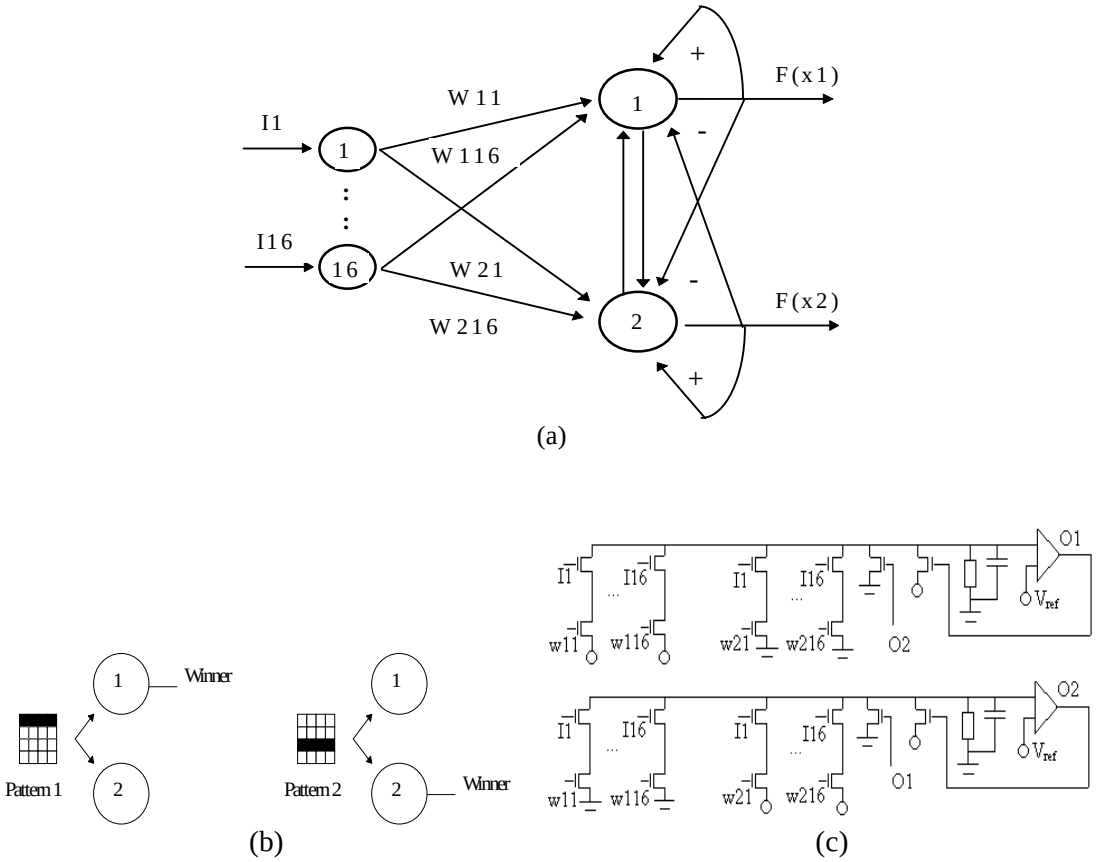


Figura 5: Modelo neural para a aplicação (a). Padrões de entrada e saída desejada para esta aplicação (b) e o diagrama do circuito para a aplicação (c).

Este circuito foi simulado por meio da ferramenta HSPICE. A Figura 6 mostra os resultados obtidos com esta simulação. As curvas 1 e 2 desta figura mostram os padrões apresentados na entrada e a curva 3 mostra os sinais de saída dos neurônios. Nesta figura pode-se observar que a rede SFBN mantém os traços STM mesmo quando os sinais de entrada são retirados.

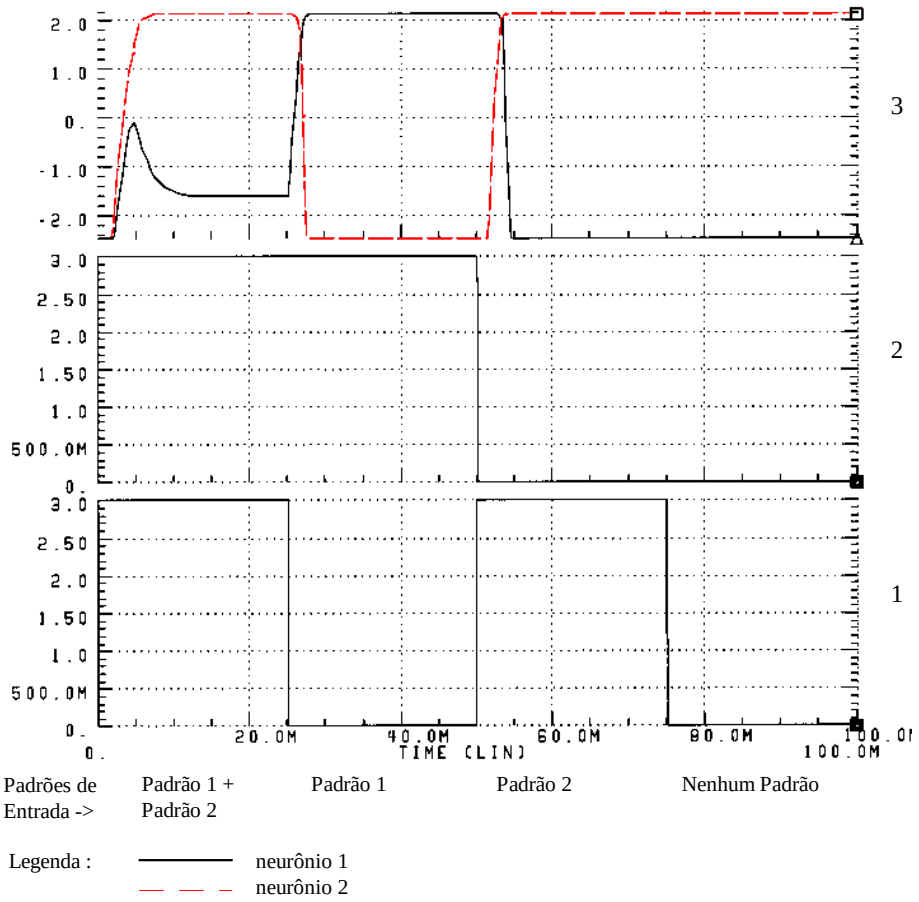


Figura 6: Resultado da simulação de uma SFBN para esta aplicação.

5. Conclusões

Neste trabalho apresentou-se uma nova proposta de implementação de redes neurais competitivas utilizando técnicas de circuitos integrados analógicos.

Realizaram-se diversas simulações elétricas para se comprovar esta proposta. Tais simulações foram realizadas através do simulador de circuitos elétricos HSPICE. Os modelos aqui mostrados, foram desenvolvidos utilizando-se componentes eletrônicos básicos, o que demonstra a simplicidade do modelo proposto. Juntamente com tais simulações, foram realizadas simulações a nível computacional, através do desenvolvimento de um aplicativo que solucionou as equações diferenciais. Tal aplicativo tem a sua importância na comparação dos resultados obtidos entre os modelos elétricos aqui propostos e as equações aqui mostradas.

A seguir apresentou-se um sistema prático utilizando as técnicas demonstradas nas seções anteriores. Tal sistema foi utilizado para clusterizar até dois padrões distintos. Foram realizadas simulações que comprovaram o funcionamento do sistema proposto através da correta clusterização dos padrões apresentados.

É importante ressaltar o fato de que durante as simulações elétricas houve o ajuste manual dos pesos. A característica de aprendizado destas redes neurais não foi investigado. Com isso dedicou-se um maior tempo para as diversas simulações realizadas. Esta característica de aprendizado deve num trabalho próximo, ser analisada e devidamente implementada. Apenas como um ponto de partida, pode-se consultar [5] e [6].

Como sequência deste trabalho, está sendo desenhado o *lay-out* de um *chip*, correspondente a aplicação mostrada, para ser realizado em uma *foundry*.

Salienta-se, também, que estas redes são de fácil desenho a nível de *lay-out* para futura implementação em *hardware* ,devido ao fato de se conseguir criar blocos que podem ser repetidos conforme o número de neurônios existentes na rede neural. Contudo, com o aumento do número de neurônios pertencentes a rede neural, observa-se um problema com relação a implementação dos pesos sinápticos, formada pelos transistores CMOS, e também do roteamento necessário para a completa interligação, tornando-se um fator limitante do tamanho da rede a ser implementada em *hardware*. Com isso, conclui-se que embora tenha se discutido sobre algumas das possíveis implementações de redes neurais artificiais em um circuito integrado, há um problema dentre outros, que é limitador para o tamanho das redes neurais, é ele: o número de conexões entre os neurônios .

Este problema é devido a grande quantidade de conexões entre os neurônios que dificultam o roteamento dos sinais entre os neurônios do circuito integrado, aumentando a sua área. O aumento de área do circuito tem reflexos nos custos e no desempenho do chip, com a diminuição da frequência de operação do mesmo. O aumento da área provoca além do aumento de custos, um aumento do consumo de potência do circuito integrado, trazendo problemas na dissipação de potência e aumento dos custos de encapsulamento do circuito. Além disso, um maior número de conexões entre os neurônios do chip, diminui a frequência de operação do circuito por aumentar as capacitâncias parasitas do circuito.

6. Referências

[1] C. Mead. "*Analog VLSI and Neural Systems"*. [S.l]: Addison Wesley, 1989.

[2] S. Grossberg, "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures". *Neural Networks*, Vol.1, pp 17-61, 1988.

[3] Hodgkin, A.L., and Huxley, A.F. "A quantitative description of membrane current and its application to conduction and excitation in nerve". *Journal of Physiology* , Vol 117, pp 500-544, 1952.

[4] McCulloch, W.S. and Pitts, W. "A logical calculus of the ideas immanent in nervous activity". *Bulletin of Mathematical Biophysics*, Vol. 5, pp 115-133.

[5] Montalvo, A J.; Gyurcsik, R. S.; Paulos, J. J. "Toward a General-Purpose Analog VLSI Neural Network with on-chip Learning". *IEEE Trans. On Neural Network*, Vol. 8, num 5, pp 413-423, Mar. 1997.

[6] Hollis, P W.; Paulos, J. J. "A Neural Network Learning Algorithm Tailored for VLSI Implementation". *IEEE Trans. On Neural Network*, Vol. 5, num 5, pp 784-791, Sept. 1994.