

Implementação em Hardware de Redes Neurais Parcialmente Conectadas para Reconstrução de Imagens Tomográficas

Luciano Frontino de Medeiros, Hamilton Pereira da Silva, *UNINTER-PR* e

Eduardo Parente Ribeiro, *UFPR*

Abstract—This article introduces a hardware architecture for the tomographic reconstruction based on partially connected neural network. A programmable logic circuit is designed for the address control unit of EEPROM containing the problem geometry weights and RAM image pixels, coupled to a synchronous arithmetic accumulator for calculating the projections values. This implementation furnishes a connexionist basis for building fast and portable CT equipments.

Index Terms—Neural networks, tomography, reconstruction, hardware, programmable logic.

I. INTRODUÇÃO

A reconstrução de imagens tomográficas por meio de redes neurais já foi abordada anteriormente, considerando a utilização de perceptron multicamada com algoritmo de retropropagação [1][2][3]. Em trabalhos prévios, a utilização de redes neurais parcialmente conectadas foi desenvolvida pelos autores [4][5][6][7]. A abordagem focada nestes trabalhos foi considerar para a construção do algoritmo as informações referentes à geometria do problema de reconstrução de imagens de CT (*Computerized Tomography*) e propor uma arquitetura de rede sem necessidade de treinamento posterior. Um software para reconstrução de imagens foi também elaborado para aplicar o algoritmo desenvolvido.

Implementações em hardware de redes neurais têm sido bastante estudadas, estando uma boa referência para vários trabalhos em [8]. Alguns trabalhos sobre as primeiras aplicações partem de [9][10][11], representação e precisão numérica para implementações de redes neurais são encontrados em [12][13], hardware na forma de coprocessadores para aceleração de aprendizado [14], chips com redes neurais paralelizadas para controle em tempo real [15] e síntese de hardware a partir de descrição em VHDL [16]. Também podem ser encontradas implementações sem multiplicação do algoritmo de retropropagação [17], e classificadores baseados em FPGA [18]. Implementações em VLSI podem ser também encontradas [19][20][13]. A área de visão computacional também possui várias referências de implementações em hardware [21][22][23].

A proposta do presente trabalho é apresentar um modelo de implementação em hardware baseado no algoritmo desenvolvido pelos autores. Esta implementação pode facilitar a construção de circuitos dedicados para a reconstrução de ima-

gens e permitir o uso em equipamentos de tomografia portáteis. Em virtude de utilizar imagens para a entrada e saída do processo, uma grande quantidade de memória é necessária para a obtenção da imagem final. A arquitetura aplicada aqui irá produzir imagens de tamanho menor, porém servirá para expor o potencial de utilização e tal idéia pode ser extrapolada para modelos de maior escala. Por agora, a implementação irá se basear na reconstrução de imagens sem interpolação para fins de explanação simplificada da arquitetura. Para este caso, a aritmética envolvida se reduz apenas à acumulação de valores das projeções. No caso de se considerar o algoritmo com interpolação, a arquitetura da rede irá modificar de acordo com os aspectos já estudados em trabalhos prévios, necessitando de aritmética adicional. Uma expansão para uso de arquitetura paralela do algoritmo de reconstrução também é explorada aqui.

Os autores consideraram o uso de lógica programável para a implementação dos circuitos de controle e cálculo, utilizando memórias RAM e EEPROM para armazenamento de tabelas e valores de pixels de imagens. Ainda que sejam necessários outros estágios para alimentação das projeções tomográficas ao circuito e gravação em dispositivo externo da imagem reconstruída, estas fases não serão abordadas aqui, ficando a totalidade da descrição a seguir apenas para o estágio de reconstrução ou retroprojeção propriamente dito.

II. REDES PARCIALMENTE CONECTADAS

Uma projeção tomográfica de um objeto é obtida quando se atravessa este mesmo objeto por um feixe de radiação e este feixe projete uma "sombra" em um anteparo devidamente colocado (figura 1). As propriedades deste objeto tais como massa ou densidade específica irão apresentar resistência exponencial à penetração da radiação[24][25],

$$I = I_0 \exp \left\{ \int_L f(x, y) du \right\} \quad (1)$$

O sinal observado consistirá na relação logarítmica,

$$p = \ln \left(\frac{I_0}{I} \right) \quad (2)$$

O par (s, θ) representa as coordenadas do raio relativas ao objeto, s representando o deslocamento da projeção e θ o ângulo ao qual a projeção está sendo obtida. O valor $p(s, \theta)$ representa o valor da projeção nestas coordenadas. As projeções são obtidas assim para vários ângulos, e o problema da reconstrução irá residir no mapeamento de $p(s, \theta)$ para o espaço da imagem de maneira a obter os pontos $f(x, y)$ referentes à seção projetada

Luciano Frontino de Medeiros atualmente é professor e coordenador de curso no grupo UNINTER-PR, e-mail: luciano@facinter.br

Hamilton Pereira da Silva atualmente é professor e diretor de tecnologia no grupo UNINTER-PR, e-mail: hsilva@facinter.br

Eduardo P. Ribeiro atualmente é professor do Departamento de Engenharia Elétrica da UFPR

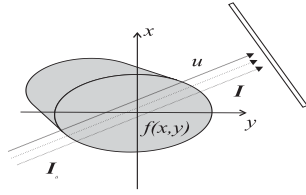


Fig. 1. Feixe de radiação atravessando um objeto.

do objeto. A transformada de Radon, definida como a integral de linha ao longo da linha inclinada a um certo ângulo θ do eixo y e à distância s da origem, descreve o processo de formação da projeção. A transformada inversa de Radon é o que se deseja calcular para obtenção da imagem tomográfica:

$$f(r, \phi) = \left(\frac{1}{2\pi^2} \right) \int_0^\pi \int_{-\infty}^{+\infty} \frac{\frac{\partial p}{\partial s}(s, \theta)}{r \cos(\theta - \phi) - s} ds d\theta \quad (3)$$

sendo (r, ϕ) as coordenadas polares da imagem a ser reconstruída. A quantidade $p(s, \theta)$ que é a projeção medida pelos detectores é chamada de *senograma* ou também de *raio-soma*, simbolizando o somatório dos pontos representados por $f(r, \phi)$ ao longo do raio a uma distância s e ângulo θ . Com este mapeamento, cada ponto em (s, θ) terá sua linha correspondente em (r, ϕ) . A maioria dos equipamentos, no entanto, utiliza como método para a reconstrução da imagem não a transformada inversa mas a retroprojeção filtrada, por uma maior facilidade numérica. Na figura 2 vemos o feixe incidindo sobre um

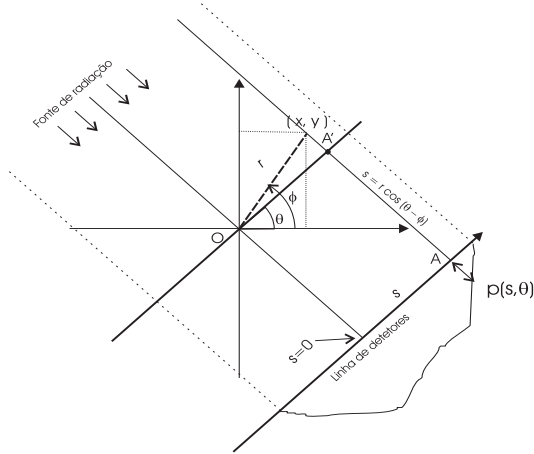


Fig. 2. Representação geométrica das projeções com raios paralelos.

objeto produzindo um perfil sobre uma linha de detectores. O ponto A se refere ao ponto sobre esta linha de detectores onde incide um raio particular s . Pode-se traçar uma reta paralela a esta linha cruzando a origem O . Assim a linha reta perpendicular que passa por A' refere-se ao raio-soma produzindo um valor de projeção $p(s, \theta)$. O ângulo θ refere-se então ao ângulo formado pelo segmento $\overline{OA'}$ com o eixo x . O deslocamento s , equivalente ao segmento $\overline{OA'}$, pode assim ser representado pela seguinte relação geométrica

$$s = r \cos(\theta - \phi) = x \cos \theta + y \sin \theta \quad (4)$$

Esta equação é fundamental para a compreensão do processo de reconstrução tomográfica, mostrando a relação puramente geométrica entre o domínio das projeções e o domínio espacial, não sendo relacionada ao conteúdo específico das projeções ou da imagem final.

A imagem tomográfica pode ser obtida com a reconstrução a partir das projeções, utilizando o algoritmo da *retroprojeção filtrada* [24][25], desenvolvido por Ramachandran e Lakshminarayanan. Em nossa abordagem, a retroprojeção filtrada é implementada por duas redes parcialmente conectadas para tirar proveito do paralelismo desta estrutura e fornecer maior rapidez no cálculo [4][7]: rede parcialmente conectada para filtragem e para retroprojeção. A arquitetura aqui proposta estará restrita à rede para retroprojeção.

O operador de retroprojeção é definido no intervalo $0 \leq \theta < \pi$. A quantidade $f(r, \phi)$ ou $f(x, y)$ é chamada de retroprojeção de $q(s, \theta)$, assumindo que são as projeções previamente filtradas. Este operador deve ser entendido como sendo uma acumulação dos raios-soma cruzando o ponto (x, y) . Portanto, é feita a integração dos raios-soma que recaem sobre a senóide dada pela equação 4. Na figura 3 vemos como exemplo a contribuição dos pontos sobre uma senóide para o ponto (x, y) na imagem final. Expressa em coordenadas polares, a equação na forma contínua é dada por

$$f(x, y) = f(r, \phi) = \int_0^\pi q(r \cos(\theta - \phi), \theta) d\theta \quad (5)$$

A forma discreta da equação anterior é

$$f(r, \phi) = \frac{1}{M} \sum_{i=1}^M q(r \cos(\theta_i - \phi), \theta_i) \quad (6)$$

onde os θ_i são os ângulos obtidos a intervalos fixos $\Delta\theta$. A

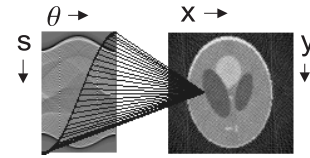


Fig. 3. Contribuição dos pontos em (s, θ) sob a senóide para um ponto (x, y) da imagem final.

expressão da rede neural parcialmente conectada pode ser interpretada como casos especiais de uma solução genérica, envolvendo a total conexão entre os pontos dos planos (s, θ) e (r, θ) ou (x, y) . Uma rede neural do tipo *feedforward* totalmente conectada é construída a partir de uma combinação linear tal como [26],

$$y_j = \sum_{i=0}^{N-1} w_{ji} x_i \quad (7)$$

onde cada saída y_j está conectada à entrada x_i através de um peso w_{ji} . Portanto, fazendo um paralelo desta equação com a fórmula de retroprojeção 6, a expressão genérica da total conexão entre os planos de projeções e imagem reconstruída

vem a ser

$$f(x, y) = \frac{1}{NM} \sum_{j=1}^N \sum_{i=1}^M w_{ji} q(s_j, \theta_i) \quad (8)$$

onde cada ponto na imagem final está totalmente conectado por pesos w_{ji} a todos os pontos do plano de projeção. O índice j se refere a valores de deslocamento no intervalo $[1, N]$ (onde N é o número de pontos da projeção) enquanto que o índice i está na faixa $[1, M]$ (onde M é o número de projeções ou ângulos).

Um caso especial sob esta solução genérica reflete a situação exposta pela equação 4, sem interpolação, com a seguinte fórmula,

$$f(x, y) = \frac{1}{M} \sum_{i=1}^M w_i q(s, \theta_i) \quad (9)$$

deduzida da equação anterior e tendo os pesos

$$w_i = 1 \quad (10)$$

e os valores de s dados¹ pela equação (4). A equação (9) fornece assim a base para a arquitetura em hardware do algoritmo de retroprojeção, exposto a seguir.

III. ARQUITETURA GENÉRICA

Para a implementação da rede em hardware, são necessários três estágios:

1. *Alimentação das Projeções Tomográficas* - neste estágio, o circuito é acionado para armazenar em memória RAM os valores de projeção para cada deslocamento de projeção s e cada ângulo θ . Um circuito contador alimenta o endereçamento da RAM, que deve estar sincronizado com o dispositivo externo, e armazena os dados das projeções.
2. *Retroprojeção* - o processamento das projeções é feito neste estágio para a produção dos valores de pixels da imagem final. Para cada pixel (x, y) e ângulo θ , o endereço da memória EEPROM é calculado para obter o valor do deslocamento s . Este valor por sua vez é usado para endereçar a RAM do senograma e obter o valor da projeção equivalente ao ponto (s, θ) . Os valores são acumulados na posição de memória RAM da imagem final.
3. *Gravação da Imagem Final* - a imagem final é disponibilizada para o dispositivo externo, de forma que seja possível transferir os dados para um circuito de vídeo ou para um barramento I/O de um PC para permitir a visualização da imagem reconstruída.

Antes de se proceder a estes estágios, porém, a EEPROM deve estar previamente gravada com o mapeamento dado pela equação (4), de acordo com o número de projeções e ângulos e o tamanho da imagem final. Na sequência, a explicação será dedicada ao estágio 2.

De acordo com a figura 4, pode-se notar os seguintes elementos que constituem a arquitetura da rede:

1. *UCE - Unidade de Controle de Endereçamento* - fornece o conjunto de sinais de controle de acesso e gravação das memórias da arquitetura, e sincroniza a acumulação dos valores para a memória da imagem final.

2. *ACC - Acumulador* - responsável por acumular os valores das projeções para produzir o valor final do pixel, de acordo com a equação (9).
3. *EEPROM - Mapeamento* - Armazena o mapeamento dado pela equação (4), e a partir do endereço combinado das coordenadas (x, y) e θ , endereça por sua vez a RAM do senograma.
4. *RAM - Senograma* - armazena os valores das projeções tomográficas $q(s, \theta)$ pré-alimentadas à rede, e alimenta o acumulador ACC para a obtenção da imagem final.
5. *RAM - Matriz de Imagem* - armazena os valores finais de pixels $f(x, y)$ fornecidos pelo acumulador ACC.
6. *I/O - Entrada-Saída* - dispositivo de entrada e saída responsável pela interface com módulos externos para manipulação das projeções e imagem reconstruída.

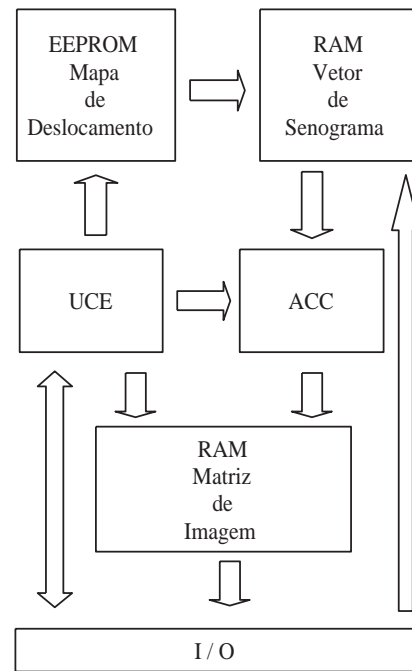


Fig. 4. Arquitetura Genérica para a Rede Parcialmente Conectada.

IV. DIMENSIONAMENTO DE MEMÓRIA

A implementação aqui descrita considera que $N = 16$ pontos e $M = 8$ ângulos. Portanto, tem-se um tamanho de imagem final de 16×16 pixels, sendo necessárias projeções tomográficas de dimensões 16×8 . Para cada ponto (s, θ) existe um valor q de projeção específico. Os valores dos pontos de projeções e da imagem final referem-se ao barramento de dados, tendo largura de 8 bits. Considerando que o processo de acumulação conta 8 ângulos, a faixa de trabalho dos valores deve permanecer no intervalo $[0, 31]$ para não ocorrer *overflow* no somatório. Assim, os valores dos pixels da imagem final ficarão no intervalo $[0, 248]$. A tabela 1 descreve os módulos de memória com o endereçamento necessário à implementação descrita.

Portanto, a EEPROM de Mapeamento deverá endereçar 2048 bytes de 4 bits de largura. A RAM de senograma deverá endereçar 128 bytes de 8 bits, e a RAM da Imagem Final deverá

¹ Com s sendo dependente de θ , r e ϕ , diminui-se o grau de liberdade para s proposto na equação 8, eliminando-se assim o índice j .

endereçar 256 bytes de 8 bits (ver figura 7).

TABELA 1: Dimensionamento dos Módulos de Memória

Módulo	Nº Bits	Descrição
EEPROM Mapeamento	11 bits	O endereçamento para uma imagem final de 16x16 exige que se tenha 256 pontos, requerendo 8 bits. Como existem 8 ângulos, e cada ângulo é combinado com cada pixel da imagem, adiciona-se mais 3 bits ao endereçamento. O valor do dado endereçado é o deslocamento s , com largura de 4 bits.
RAM Senograma	7 bits	Como os valores de projeções são fornecidos numa matriz de 16 pontos por 8 ângulos, são necessários 4 bits para o deslocamento s e mais os 3 bits necessários para os ângulos. O dado endereçado é o valor da projeção no ponto (s, θ) , com largura de 8 bits.
RAM Imagem Final	8 bits	Para a imagem final 16x16, 4 bits referem-se ao valor de x e mais 4 bits ao valor de y . O dado endereçado se refere ao pixel da imagem final com largura de 8 bits.

V. UNIDADE DE CONTROLE DE ENDEREÇAMENTO E ACUMULADOR

O diagrama da UCE mostrado na figura 5 mostra o circuito montado em lógica programável. Um contador de 11 bits monitora o barramento de endereços necessários para a EEPROM ($INPUT[10..0]$), fornecendo os bits referentes à posição (x, y) e o ângulo θ ($ANG0$ a $ANG2$). Pode-se verificar ainda os pinos de entrada EN , CLK e CLR_CNT . Estes pinos devem ser controlados por lógica externa, com EN habilitando a UCE, CLK fornecendo o trem de pulsos e CLR_CNT reiniciando o contador de 11 bits. O pino OEG habilita a memória EEPROM para endereçar a memória RAM do senograma. O pino CSP seleciona por sua vez a RAM do senograma para fornecer o dado de projeção; e o pino CSI habilita a RAM da imagem final para armazenar o dado proveniente do acumulador. Os pinos ACC e CLR_ACC controlam, respectivamente, o clock e o reset do acumulador. Na figura 6 vemos o símbolo em lógica programável do acumulador. Os pinos $IN_ACC[7..0]$ referem-se ao dado de entrada do senograma. Os pinos $OUT_ACC[7..0]$ referem-se à saída do acumulador. O pino CLK é o clock para sincronizar com a UCE, e o pino CLR_ACC zera o acumulador para uma nova contagem de ângulos. Internamente, o acumulador funciona como um somador completo síncrono. A precisão dos valores dos pixels depende apenas da largura em bits dos dados endereçados na RAM do senograma e RAM da imagem final, e da entrada-saída do acumulador. No diagrama da figura 7 vê-se a conexão entre os módulos de memória, a UCE e o acumulador. Este diagrama simboliza o circuito final de retroprojeção necessário para reconstruir as imagens de tamanho 16x16, a partir de projeções 16x8. O processo de reconstrução é bastante rápido, pois em menos de 1 ms têm-se a imagem final.

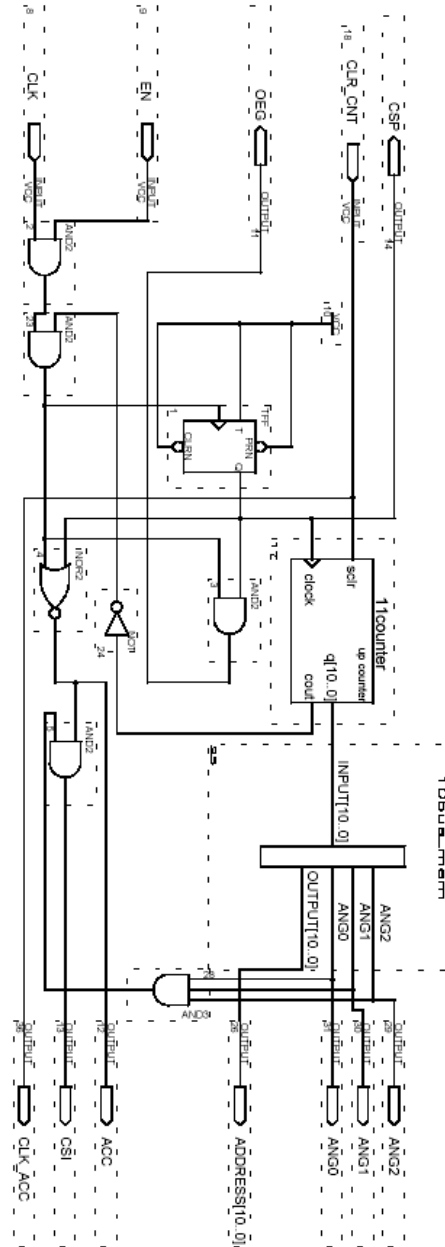


Fig. 5. Lógica Programável para a Unidade de Controle de Endereçamento.

VI. ASPECTOS DE SIMULAÇÃO

A simulação de sinais da figura 8 mostra o incremento do barramento de endereço $ADDRESS[10..0]$ acontecendo a cada $0.4\mu s$. Com EN alto, a cada subida de borda de OEG e CSP o barramento de endereços aumenta em uma unidade. Pode-se verificar também que o clock para o acumulador é alto antes desta subida de borda para o acumulador efetuar a soma anterior. Quando a contagem dos ângulos (que é a primeira parte do endereço enviado para a EEPROM), chegar a 8, o pino CSI é colocado em alto (não mostrado na figura), de forma a gravar o dado em $OUT_ACC[7..0]$. O endereçamento é concluído após $820\mu s$, quando se terá a imagem final gravada na RAM. A UCE e o acumulador foram simulados em lógica programável

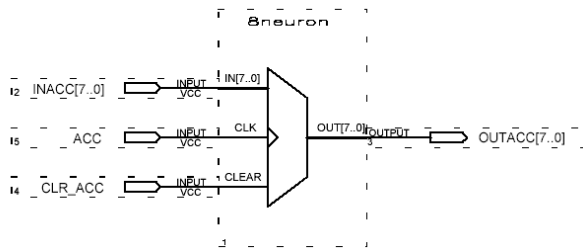


Fig. 6. Símbolo em Lógica Programável para o Acumulador.

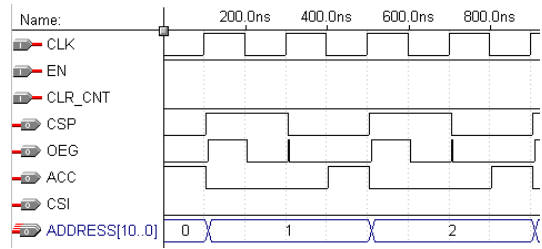


Fig. 8. Simulação do conjunto de sinais para endereçamento.

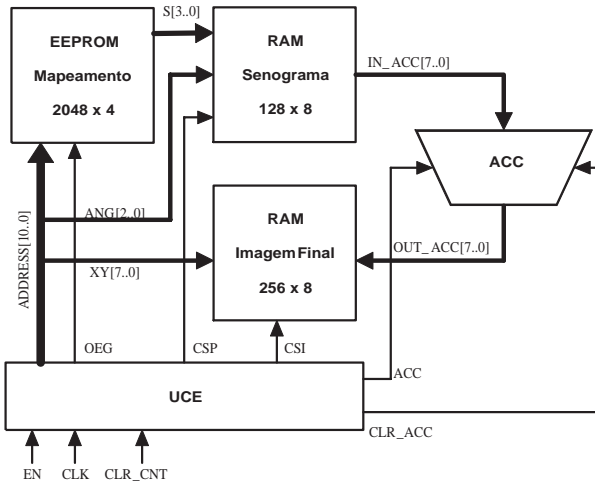


Fig. 7. Diagrama de conexão entre os módulos de memória e a UCE.

utilizando o software MAX+PLUS II da Altera Devices [27], sendo gravado em apenas um chip EPM3256ATC144-7, da família MAX3000A. O circuito utilizou 11 pinos de entrada e 26 pinos de saída, consumindo 124 células lógicas (referente a 48% do chip).

VII. ARQUITETURA EM PARALELO

Uma extensão da proposta aqui descrita é que o processo de retroprojeção pode ser representado por uma arquitetura em paralelo. Pode-se incluir vários conjuntos de memória EEPROM-RAM de senograma em paralelo, modificando o acumulador para aceitar mais entradas. Cada EEPROM endereça um ângulo específico, de forma que a referência ficaria apenas ao ponto de imagem (x, y) . A UCE apresentaria o mesmo circuito, não necessitando de modificações (apenas alterando-se a temporização). Na figura 9, pode ser visualizada a implementação em paralelo, apresentando 8 conjuntos EEPROM-RAM de senograma. Cada conjunto se refere a um ângulo específico, e as EEPROM devem ser gravadas de forma a possuírem os endereços dos pontos (x, y) para cada ângulo. O acumulador possui 8 entradas, de forma que com dois ciclos de clock o acumulador fornecerá o resultado à RAM de imagem final. Com esta arquitetura em paralelo, o tempo de execução pode cair para algo em torno de $100\mu s$.

Com a paralelização, as 8 memórias EEPROM de mapeamento deverão endereçar agora 256 bytes de 4 bits de largura.

As 8 memórias RAM de senograma deverão endereçar 16 bytes de 8 bits, e a RAM da imagem final deverá endereçar 256 bytes de 8 bits (ver figura 9). Uma desvantagem é que deve ser prevista a gravação de 8 memórias EEPROM e a alimentação do senograma a 8 memórias RAM exigindo um hardware diferenciado, em contrapartida à arquitetura simples.

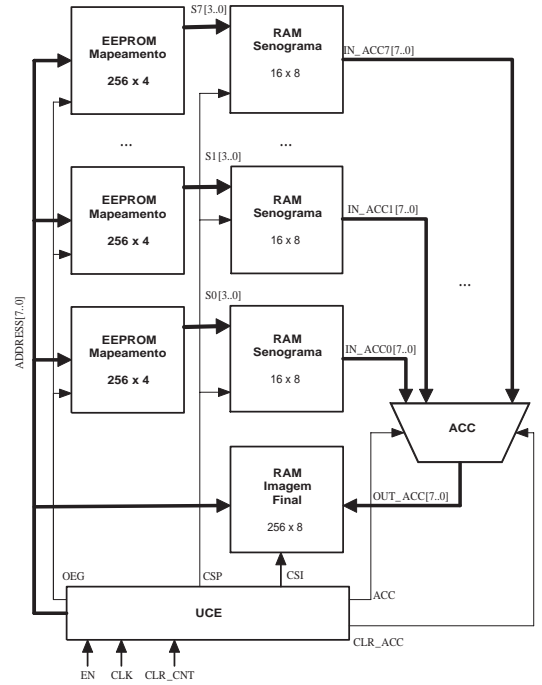


Fig. 9. Diagrama representando a implementação em paralelo.

VIII. CONCLUSÃO

Este artigo tratou da implementação em hardware de redes neurais parcialmente conectadas para reconstrução de imagens de tomografia a partir de projeções. Apresentando uma arquitetura genérica, descreveu também aspectos como dimensionamento de memória a partir dos parâmetros de imagem e senograma, bem como uma proposta de circuito para a unidade de controle de endereçamento e o acumulador. Pôde-se verificar que nenhum cálculo foi necessário para obter os endereços, sendo todo o processo controlado de forma centralizada. A única operação aritmética utilizada foi a soma via acumulação dos valores de projeções para os ângulos. Foi explorado também uma proposta para arquitetura em paralelo,

evidenciando as vantagens da abordagem conexionista para o problema de reconstrução de imagens de CT implementada em hardware. Com base nisto, está se montando um protótipo da implementação para a retroprojeção apresentada aqui e em trabalhos futuros pretende-se mostrar os resultados conseguidos.

No caso de um circuito para a fase de retroprojeção utilizando interpolação, será necessário introduzir, além do processo de acumulação mencionado aqui, a operação de multiplicação dos pesos que representam os valores para a interpolação dos pontos do senograma.

Os estágios de alimentação de projeções e gravação da imagem final em dispositivos externos serão abordados em trabalhos subsequentes, assim como a fase de filtragem prevista na teoria das redes neurais parcialmente conectadas.

REFERENCES

- [1] M. T. et al, "An artificial neural network approach to quantitative single photon emission computed tomographic reconstruction with collimator, attenuation and scatter compensation," *Med. Physics*, vol. 12, no. 21, 1994.
- [2] J. P. Kerr and E. B. Bartlett, "A statistical tailored neural network approach to tomographic image reconstruction," *Med. Physics*, vol. 5, no. 22, 1995.
- [3] R. G. S. Rodrigues, *Desenvolvimento e Aplicação de um Algoritmo de Reconstrução Tomográfica com Base em Redes Neurais Artificiais*, Ph.D. thesis, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Ribeirão Preto, SP, 2000.
- [4] L. F. de Medeiros, H. P. da Silva, and E. P. Ribeiro, "Reconstrução de imagens tomográficas usando redes neurais parcialmente conectadas," in *Proceedings of the V International Conference on Neural Networks*, Rio de Janeiro, RJ, 2001, pp. 289–294.
- [5] L. F. de Medeiros, H. P. da Silva, and E. P. Ribeiro, "Reconstrução de imagens tomográficas geradas por projeções fan-beam com detectores equidistantes usando redes neurais parcialmente conectadas," in *Proceedings of the VI International Conference on Neural Networks*, São Paulo, SP, 2003, pp. 301–306.
- [6] L. F. Medeiros, H. P. da Silva, and E. P. Ribeiro, "Tomographic reconstruction images using partially connected neural networks," *Learning and Non-Linear Models, SBRN*, vol. 2, no. 2, pp. 134–143, 2003.
- [7] L. F. de Medeiros, "Reconstrução de imagens tomográficas com redes neurais parcialmente conectadas," M.S. thesis, Universidade Federal do Paraná, Curitiba, PR, 2001.
- [8] A. Perez-Urbe, *Structure Adaptable Digital Neural Networks*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne, France, 1999.
- [9] I. Aleksander and E.H. Mamdani, "Microcircuit Learning Nets: Improved Recognition by Means of Pattern Feedback," *Electronic Letters*, vol. 4, no. 20, pp. 425–426, October 1968.
- [10] M.C. Fairhurst and I. Aleksander, "Natural Pattern Clustering in Digital Learning Nets," *Electronic Letters*, vol. 7, no. 24, pp. 724–726, October 1971.
- [11] J.S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *Journal of Dynamic Systems, Measurement and Control*, vol. 97, pp. 220–227, September 1975.
- [12] I. Aleksander and H. Morton, *An Introduction to Neural Computing*, International Thomson Computer Press, second edition, October 1995.
- [13] P. Inne, T. Cornu, and G. Kuhn, "Special-purpose digital hardware for neural networks: An architectural survey," *Journal of VLSI Signal Processing*, pp. 13(1):5–25, 1996.
- [14] H. Speckmann, P. Thole, M. Bogdan, and W. Rosentiel, "Coprocessor for special neural networks KOKOS and KOBOLD," in *Proc. ICNN'94, International Conference on Neural Networks*, Piscataway, NJ, 1994, pp. 1959–1962, IEEE Service Center.
- [15] J. Liu and M. Brooke, "A fully parallel learning neural network chip for real-time control," 1999.
- [16] H. Speckmann, P. Thole, and W. Rosentiel, "Hardware synthesis for neural networks from a behavioral description with VHDL," in *Proc. IJCNN-93, International Joint Conference on Neural Networks*, Nagoya, Piscataway, NJ, 1993, vol. II, pp. 1983–1986, IEEE Service Center.
- [17] J. Cloutier and P.Y. Simard, "Hardware implementation of the backpropagation without multiplication," in *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems Microneuro'96*, September 1994, pp. 46–55, IEEE.
- [18] C. Cox and E. Blanz, "GANGLION: A Fast Field Programmable Gate Array implementation of a Connectionist Classifier," *IEEE Journal of Solid State Circuits*, vol. 27, no. 3, pp. 288–299, March 1992.
- [19] A. Hamilton, S. Churcher, P.J. Edwards, G.B. Jackson, A.F. Murray, and H.M. Reekie, "Pulse stream VLSI circuits and systems: the epsilon neural network chipset," in *Proceedings of the Third International Conference on Microelectronics for Neural Networks*, December 1993, vol. 4, pp. 395–405.
- [20] P. Inne, *Programmable VLSI Systolic Processors for Neural Network and Matrix Computations*, Ph.D. thesis, Swiss Federal Institute of Technology-Lausanne, Lausanne, EPFL, 1996, Thesis 1525.
- [21] G. de Trémiolles, P. Tannhof, B. Plougonven, C. Demarigny, and K. Madani, "Visual probe mark inspection, using hardware implementation of artificial neural networks in VLSI production," in *Biological and Artificial Computation: From Neuroscience to technology*, R. Moreno-Díaz J. Mira and J. Cabestany, Eds., Lecture Notes in Computer Science, 1997, vol. 1240, pp. 1374–1383, Springer Verlag.
- [22] C. Mead, "Neuromorphic electronic systems," in *Neuro-Vision Systems*, M. M. Gupta and G. K. Knopf, Eds., pp. 463–470, IEEE, New York, 1994.
- [23] J. C. Lee, B. J. Sheu, W. C. Fang, and R. Chellappa, "Vlsi neuroprocessors for video motion detection," in *Neuro-Vision Systems*, M. M. Gupta and G. K. Knopf, Eds., pp. 461–483, IEEE, New York, 1994.
- [24] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.
- [25] G. T. Herman, *Image Reconstruction from Projections*, Academic Press, Orlando, Florida, 1980.
- [26] S. Haykin, *Neural Networks - A Comprehensive Foundation*, Macmillan Coll. Pub., 1994.
- [27] Altera, *Data Book*, Altera Corporation, Santa Clara, 1996.



Luciano Frontino de Medeiros nasceu em Porto Alegre-RS em 1966. É mestre em Informática pela Universidade Federal do Paraná, tendo cursado Administração e Física Licenciatura pela Universidade Federal de Santa Maria. É autor de livro e artigos na área de Redes Neurais. Atualmente é professor de Inteligência Artificial, Banco de Dados e Sistemas de Informação nos cursos de graduação e pós-graduação da UNINTER-PR.



Hamilton Pereira da Silva nasceu em Santa Maria-RS em 1964. É mestre em Informática Industrial pelo CEFET-PR e cursou Física pela Universidade Federal do Paraná. Desenvolveu trabalhos e publicações na área de imagens de Tomografia Computadorizada. Atualmente é professor nos cursos de graduação e pós-graduação da UNINTER-PR.



Eduardo Parente Ribeiro nasceu no Rio de Janeiro em 1967. Recebeu o título de Mestre e Doutor em Engenharia Elétrica em 1992 e 1996 pela PUC-Rio. Suas áreas de interesse incluem redes neurais aplicada ao processamento de sinais. Desde 1996 é professor no Departamento de Engenharia Elétrica da Universidade Federal do Paraná tendo orientado e publicado diversos trabalhos na área.