

S. W. SONG *

SUMÁRIO

Na primeira parte do trabalho serão apresentadas disposições planares compactas para diversos tipos de estruturas de árvore, como árvores binárias com ligações especiais ("threaded trees") e árvores binárias com ligações horizontais ("x-trees"). Na segunda parte serão mostrados resultados de particionamento de árvores que têm aplicação na implementação de sistemas sistólicos em forma de árvore binária ou árvore com ligações especiais, que não cabem em uma única pastilha. Um método será proposto para dispor as pastilhas componentes de maneira eficiente numa placa de circuito impresso.

ABSTRACT

In the first part of this paper we present compact planar layouts for various kinds of tree structures, such as the threaded trees and the x-trees with brother links. In the second part, we show tree partitioning results with application in the implementation of systolic systems with the structure of binary trees or binary threaded trees, that do not fit into one single chip. It will be shown how the component chips can be laid out efficiently on PC boards.

* Engenheiro Eletrônico (EPUSP, 1970), mestre em Matemática Aplicada, modalidade Computação (IMEUSP, 1975), Ph.D. em Ciência de Computação (Carnegie-Mellon University, 1981); Computação Paralela e Computação VLSI; Professor Assistente-Doutor e atual Chefe do Departamento de Matemática Aplicada do Instituto de Matemática e Estatística da USP; IME/USP, Rua do Matão, 1010, Cidade Universitária, CEP 05508, São Paulo, SP; Patrocínio financeiro: CNPq, através da bolsa de pesquisa - processo 300438-82.

Sistemas sistólicos de alto desempenho, baseados em estruturas hexagonais de elementos de processamento, foram propostos por Kung e Leiserson [5] para computações matriciais. Outras estruturas, como a lista linear, matriz ortogonal e árvore binária, têm sido usadas para diversos outros problemas. A simplicidade e regularidade das interconexões dessas estruturas são propriedades desejáveis para implementação em VLSI [3]. Elas garantem a comunicação local entre elementos vizinhos e uma disposição geralmente compacta na pastilha. Além disso, tais estruturas são modulares, no sentido de permitir a extensão do tamanho da estrutura pela simples duplicação de estruturas menores.

Este trabalho é organizado em duas partes. Na primeira parte, constituída pelas seções 2 e 3, serão propostas disposições compactas planares de diversos tipos de árvore. Nessas disposições, procura-se minimizar a área ocupada e o comprimento das linhas de interconexão entre os elementos ou nós da árvore. Uma aplicação desses resultados é na implementação em pastilha de um sistema sistólico em forma de um dos tipos de árvores apresentados. Os tipos de árvores a serem considerados incluem, além da árvore binária, a árvore binária com folhas ou nós terminais ligados, a estrutura híbrida de árvore e lista linear, a árvore com ligações especiais ou "threads", a x-árvore ou árvore com ligações horizontais e a árvore com ligações especiais múltiplas.

Na segunda parte, seção 4, serão apresentados esquemas de particionamento da árvore binária e árvore binária com ligações especiais em regiões contendo subárvores e a subsequente interligação dessas regiões componentes no plano. Uma aplicação desses resultados é na implementação de um sistema sistólico constituído de uma grande árvore binária, com ou sem ligações especiais, que não cabe numa única pastilha. O particionamento em regiões corresponde à subdivisão do sistema sistólico em pastilhas componentes, que serão dispostas de forma compacta em placas de circuito impresso.

2. ESTRUTURAS DE ÁRVORES CONSIDERADAS

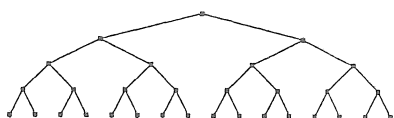


Figura 1

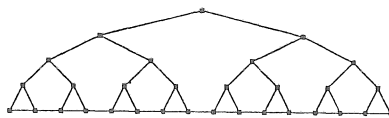


Figura 2

Nesta seção, vamos apresentar os diversos tipos de árvores consideradas. A estrutura da árvore binária da Figura 1 tem sido proposta em vários trabalhos, entre outros, por Bentley e Kung [1], Browning [2] e Song [12]. A Figura 2 mostra uma árvore binária com folhas ou nós terminais ligados. Essa estrutura foi proposta por Leiserson [6] e Magô [8].

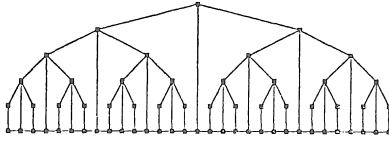


Figura 3

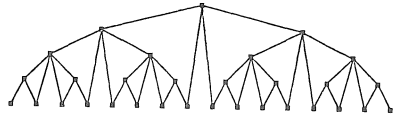


Figura 4

A estrutura híbrida, da Figura 3, é a combinação de uma árvore binária e uma lista linear, com as ligações verticais ligando seus nós. Se denominarmos de n um nó da árvore binária e n' a sua projeção vertical na lista linear, então um percurso em "in-order" (Knuth [4]) dos nós n da árvore binária corresponderá a um percurso dos nós da lista linear da esquerda para a direita. A árvore binária com ligações especiais (ou "threads"), mostrada na Figura 4, foi usada por Rosenberg, Wood e Galil [10], que deram o nome de "dree". A estrutura híbrida foi proposta por Song [13].

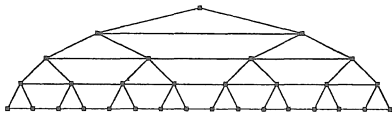


Figura 5

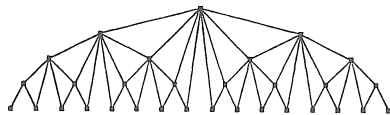


Figura 6

A árvore binária com ligações horizontais, mostrada na Figura 5, também é conhecida pelo nome de x -árvore, conforme o trabalho de Sequin, Despain e Patterson [11]. As ligações horizontais conectam os nós irmãos e primos de um mesmo nível. Finalmente, na Figura 6, é mostrada uma árvore binária com ligações especiais múltiplas.

3. DISPOSIÇÕES COMPACTAS PLANARES

As disposições das Figuras 1 a 6 não são adequadas para implementação em pastilhas. Sendo n o número de nós terminais ou folhas da árvore, a área ocupada naquelas disposições é proporcional a $n \log n$ (logaritmo na base 2). A pastilha teria um formato de um retângulo demasiadamente alongado, com um lado proporcional a n e o outro lado proporcional a $\log n$.

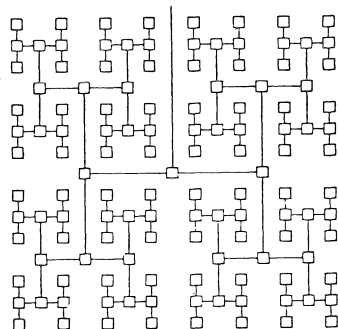


Figura 7

Uma disposição eficiente para a árvore binária da Figura 1 é proposta por Mead e Rem [9]. A chamada disposição-H, mostrada na Figura 7, requer uma área proporcional ao número de nós da árvore. Mostraremos agora como as árvores das Figuras 2 a 6 também podem ser dispostas eficientemente, gozando dessa mesma propriedade.

Mostremos antes que as estruturas das Figuras 2 e 3 são transformáveis à estrutura da Figura 4, e que a estrutura da Figura 5 é transformável à da Figura 6. Imaginemos que as ligações horizontais das Figuras 2, 3 e 5 sejam "elásticas". Se as ligações verticais da Figura 3 começam a encolher de comprimento até atingirem o comprimento nulo, a estrutura resultante será idêntica à da Figura 4. Para a árvore da Figura 2, imaginemos uma pressão aplicada de baixo para cima, elevando as ligações horizontais. Teremos novamente a estrutura resultante idêntica à da Figura 4. Analogamente, a elevação das ligações horizontais da estrutura da Figura 5 dará origem à estrutura da Figura 6. Iremos portanto apresentar disposições planares apenas para a árvore binária com ligações especiais (Figura 4) e a árvore binária com ligações especiais múltiplas (Figura 6), já que as demais são equivalentes a elas.

A árvore binária com ligações especiais da Figura 4 pode ser disposta, como mostra a Figura 8, por um método recursivo. Nesse método, é aplicada uma operação de rotação a partes selecionadas em torno de um eixo apropriado, de tal maneira que as ligações especiais, ou "threads", têm comprimento constante. Tal não se verifica na Figura 4, onde as ligações especiais se tornam mais compridas à medida que elas se aproximam do centro.

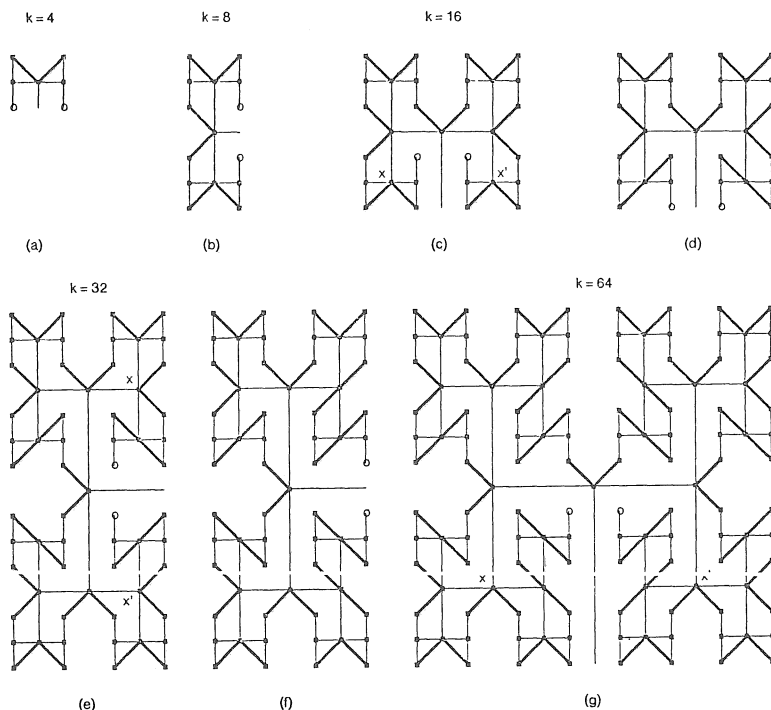


Figura 8

Seja k uma potência de 2, denotando o número de nós terminais de uma árvore binária com ligações especiais. A Figura 8 (a) mostra o caso de $k=4$. As linhas grossas representam as ligações especiais e os círculos pequenos denotam os dois nós terminais extremos, que não possuem ligações especiais. A Figura 8 (b), para $k=8$, é obtida combinando-se (a) com a sua imagem especular em relação a um eixo horizontal, e adicionando-se duas ligações especiais a mais. De maneira análoga, a Figura 8 (c), para $k=16$, é obtida combinando-se (b) com a sua imagem especular em relação a um eixo vertical, e adicionando-se duas ligações especiais.

Para conseguirmos a disposição para $k=32$, primeiro obtemos (d) a partir de (c), aplicando-se uma rotação de parte de (c) ao longo do eixo $x-x'$, a fim de trazer os nós terminais extremos (pequenos círculos) para perto da periferia. A Figura 8 (e), para $k=32$, é então obtida combinando-se (d) com a sua imagem especular em relação a um eixo horizontal e adicionando-se mais duas ligações especiais. Para $k=64$, primeiro obtemos (f) de (e) pela rotação de parte de (e) ao longo do eixo $x-x'$ e depois gerar (g). Assim,

a disposição que resulta tem a propriedade, por construção, de que todas as ligações especiais têm comprimento constante, ligando nós próximos.

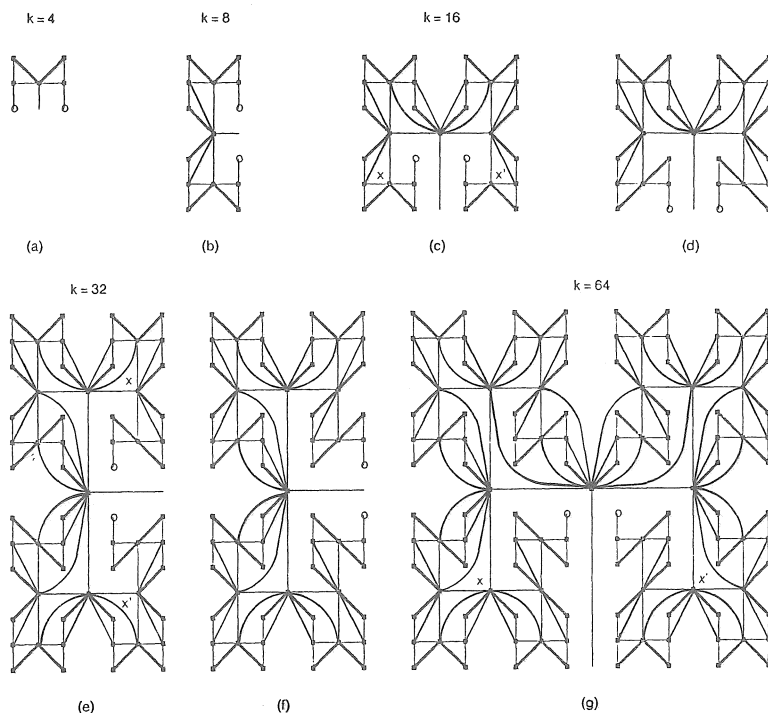


Figura 9

Para obtermos uma disposição compacta para a estrutura de árvore binária com múltiplas ligações especiais, da Figura 6, basta observar que cada árvore parcial desta estrutura, constituída pelos nós desde a raiz até os nós de algum nível, contém a árvore com ligações especiais, do tipo da Figura 4, que já sabemos como dispor. Aplicando assim o mesmo método, teremos a disposição da Figura 9.

4. PARTICIONAMENTO DE ÁRVORES

Consideremos uma estrutura de árvore binária grande, que não cabe numa única pastilha. Queremos particioná-la em regiões ou pastilhas componentes. As propriedades desejáveis num esquema de particionamento de uma estrutura em pastilhas componentes são:

1. Usa poucos tipos diferentes de pastilhas componentes.
2. Usa a área de cada pastilha componente de maneira efetiva (sem desperdício).
3. Permite uma disposição compacta das pastilhas componentes, que serão montadas em placas de circuito impresso.
4. Usa linhas curtas para interligar pastilhas componentes.

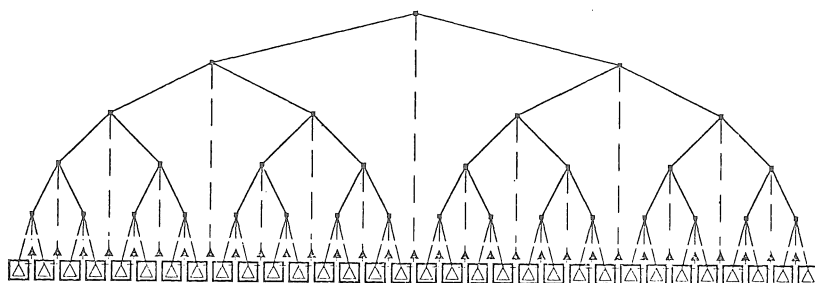


Figura 10

Na Figura 10, cada quadrado representa uma pastilha contendo uma subárvore binária (representada por um triângulo dentro da pastilha). Chamemos tais pastilhas de pastilhas terminais. Para uma árvore binária de k pastilhas terminais, temos exatamente $k-1$ nós internos da árvore binária que ainda precisam ser alocados em pastilhas. Para acomodar tais nós internos, vamos fazer com que cada pastilha terminal, com exceção da pastilha mais à direita, receba mais um nó, que é o seu sucessor em "in-order", indicado pela linha tracejada da Figura 10. Temos assim a Figura 11.

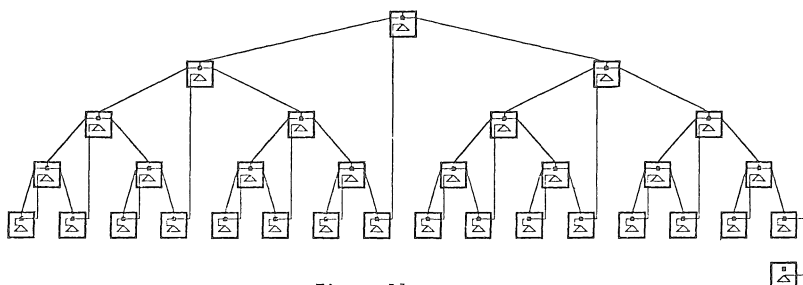


Figura 11

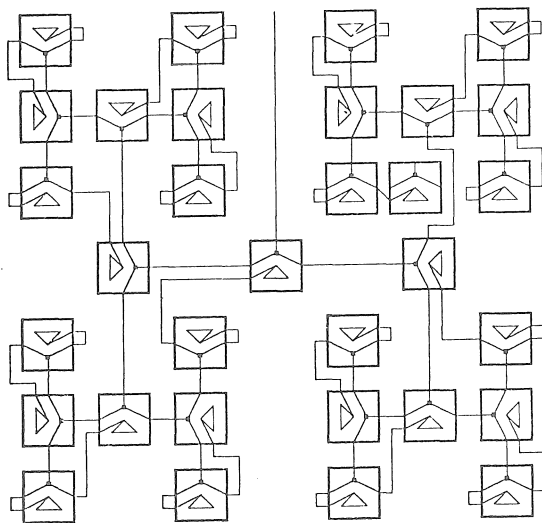


Figura 12

Na Figura 11, conseguimos dispor a árvore binária original em pastilhas componentes, todas de um mesmo tipo. Cada pastilha componente tem sua área usada de maneira efetiva. Como as pastilhas componentes estão dispostas em forma de um tipo particular de árvore binária com ligações especiais (uma só ligação especial ao invés de duas), podemos usar o resultado da seção 3 para produzir a disposição final da Figura 12.

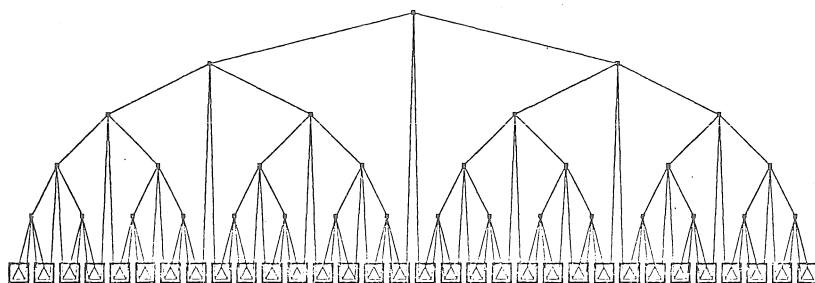


Figura 13

A Figura 13 mostra uma árvore binária com ligações especiais. Novamente, vamos supor que ela deva ser decomposta em pastilhas componentes. Cada quadrado da Figura 13 representa uma pastilha componente, contendo uma subárvore denotada pelo triângulo.

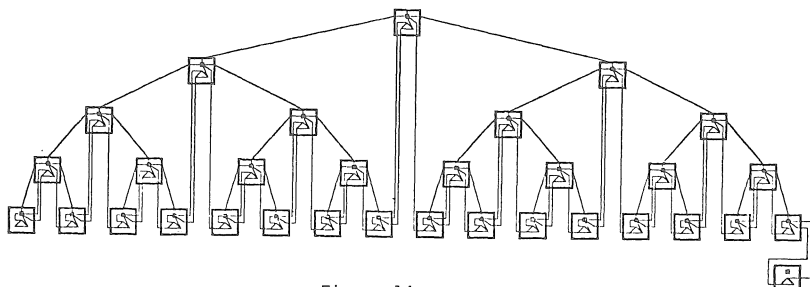


Figura 14

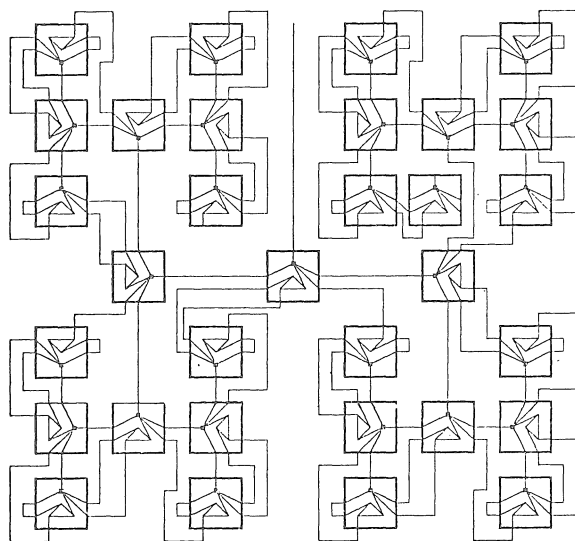


Figura 15

Como antes, fazemos com que cada pastilha terminal receba mais um nó interno, resultando assim na Figura 14. A disposição das pastilhas componentes é exatamente a de uma árvore binária com ligações especiais, onde os nós são as pastilhas. Podemos então apresentar a disposição final da Figura 15, usando novamente resultado da seção 3.

5. CONCLUSÕES

Mostramos na primeira parte deste trabalho que a disposição-H pode ser estendida a diversos outros tipos de estruturas de árvore, onde há ligações especiais, ou ligações horizontais entre nós irmãos e primos. Sendo todas árvores binárias, a idéia de tentar essa extensão não é extraordinária. A importância do trabalho reside principalmente em mostrar que as ligações adicionais podem ser acomodadas dentro da disposição-H de uma forma sistemática e eficiente.

Quanto à segunda parte, a idéia de incorporar um nó interno da árvore binária em cada pastilha terminal é devida a Leiserson [7]. O método proposto aqui difere do dele na escolha do nó a ser combinado, além da utilização da operação de rotação para conseguir a propriedade desejável de comprimento constante nas ligações especiais. No que se refere ao comprimento total das linhas de interconexão entre pastilhas componentes, o método aqui proposto usa essencialmente um quarto do comprimento no método de Leiserson. Além disso, o método proposto é mais geral no sentido de poder estender-se a outros tipos de árvores, como a árvore binária com ligações especiais.

BIBLIOGRAFIA

1. Bentley, J. L., & Kung, H. T., "A Tree Machine for Searching Problems", *Proceedings of 1979 International Conference on Parallel Processing*, pp. 257-266. IEEE. August, 1979.
2. Browning, S. A., "Computations on a Tree of Processors", *Proceedings Conference on Very Large Scale Integration: Architecture, Design, Fabrication*, pp. 453-478. January, 1979.
3. Foster, M. J., & Kung, H. T., "The Design of Special-Purpose VLSI Chips", *Computer*, pp. 26-40. January, 1980.
4. Knuth, D. E., *The Art of Computer Programming. Volume 1: Fundamental Algorithms*, 634 p. Addison-Wesley, 1973.
5. Kung, H. T., & Leiserson, C. E., "Systolic Arrays for VLSI", *Sparse Matrix Proceedings*, pp. 256-282. SIAM. 1979.
6. Leiserson, C. E., "Systolic Priority Queues", *Proceedings Conference on Very Large Scale Integration: Architecture, Design, Fabrication*, pp. 199-214. January, 1979.

7. Leiserson, C. E., *Area-Efficient VLSI Computations*, 132 p. Ph. D. Thesis, Carnegie-Mellon University, Computer Science Department. 1981.
8. Magō, G. A., "A Network of Processors to Execute Reduction Languages", *International Journal of Computer and Information Sciences*, pp. 349-385. March, 1979.
9. Mead, C., & Rem, M., "Cost and Performance of VLSI Computing Structures", *IEEE Journal of Solid State Circuits*, pp. 455-462. April, 1979.
10. Rosenberg, A. L., Wood, D., & Galil, Z., "Storage Representations for Tree-like Data Structures", *Mathematical Systems Theory*, pp. 105-130. Volume 13, Number 2, 1979.
11. Sequin, C. H., Despain, A. M., & Patterson, D. A., "Communication in X-trees, a Modular Multiprocessor System", *ACM Proceedings 1978 Annual Conference*, pp. 194-203. December, 1978.
12. Song, S. W., "A Highly Concurrent Tree Machine for Database Applications", *Proceedings of the 1980 International Conference on Parallel Processing*, pp. 259-268. August, 1980.
13. Song, S. W., *On a High-Performance VLSI Solution to Database Problems*, 161 p. Ph. D. Thesis, Carnegie-Mellon University, Computer Science Department. 1981.