PROPOSTA DE UM SISTEMA PARALELO COM APLICAÇÕES EM PROCESSAMENTO DE IMAGENS

Celso Luiz Mendes
Ministério da Ciência e Tecnologia - MCT
Instituto de Pesquisas Espaciais - INPE
Caixa Postal 515 - São José dos Campos, SP - 12201

SUMÁRIO

É feito um breve resumo sobre algumas famosas máquinas paralelas já desenvolvidas para o processamento de imagens digitais, situando o presente trabalho com uma extensão natural nesta direção. São mostradas a seguir as principais características do componente que está sendo estudado, com arquitetura altamente paralela e operando no modo SIMD, juntamente com as ferramentas já desenvolvidas para a sua programação e alguns exemplos de aplicações básicas simuladas, comentando-se as vantagens e desvantagens da sua utilização. Em seguida é proposta uma arquitetura para um sistema baseado neste componente e ligado a um microcomputador do tipo IBM-PC, com possíveis aplicações em processamento de imagens. Ao final são apontados possíveis desenvolvimentos futuros que viriam incrementar a funcionalidade do sistema.

1. INTRODUÇÃO

A área de processamento de imagens tem sido tradicionalmente visada como objetivo principal no desenvolvimento de diversas máquinas paralelas. Isto se deve ao fato de que, por um lado, há uma grande massa de dados a serem processados, e por outro as operações tipicamente realizadas podem, em geral, ocorrer simultaneamente em regiões distintas da imagem, já que cada ponto tende a influenciar apenas seus vizinhos mais próximos.

Um dos projetos mais antigos que merece ser citado é o da máquina denominada SOLOMON ("Simultaneous Operation Linked Ordinal Modular Network"), ainda na década de 60, e que apesar de nunca ter sido construído conforme a concepção original, constitui-se num marco por ter dado origem a vários desenvolvimentos de máquinas

com arquiteturas semelhantes, incorporando conceitos de paralelismo que não eram utilizados até então. Sua estrutura proposta consistia de uma matriz com 32x32 processadores, cada um deles tendo uma memória local com 128 palavras de 32 bits e uma unidade aritmética operando serialmente por bit, sob controle de uma unidade central.

Dentre as diversas máquinas originadas do SOLOMON, uma das mais populares foi o ILLIAC IV, desenvolvido na Universidade de Illinois (EUA). Ele era composto de quatro quadrantes interligados por um barramento de E/S paralelo; cada quadrante possuía sua própria unidade de controle e uma matriz de 8x8 processadores, havendo em cada um destes processadores uma memória com 2k palavras de 64 bits e a capacidade de realizar operações de ponto flutuante à velocidade de 240ns. Isto possibilitaria, teoricamente, um desempenho global da ordem de 1 GFLOP/s. Na prática, porém, apenas um quadrante foi construído e o desempenho máximo ficou em aproximadamente 50 MFLOP/s, devido principalmente à problemas tecnológicos de implementação. Apesar do aparente fracasso, já que o desempenho não chegou a 10% do previsto e o custo acabou quadriplicado em relação à estimativa inicial, este projeto teve um papel muito importante por servir de base para projetos atuais, que já podem contar com os avanços na área de componentes eletrônicos, o que torna viável a implementação prática a um custo accessível.

Outras máquinas derivadas do SOLOMON e com aplicações em processamento de imagens foram o DAP ("Distributed Array Processor") e o CLIP ("Cellular Logic Image Processors"), ambos desenvolvidos na Inglaterra na década de 70 e baseados numa matriz bidimensional de processadores interligados.

Um conceito bastante importante para a área de processamento paralelo foi introduzido por Shooman em 1960, consistindo na definição de um "processador ortogonal"; tal processador, ao acessar dados de uma memória, buscaria simultaneamente um certo bit de várias palavras, ao contrário do acesso convencional onde se busca todos os bits de uma dada palavra. Este novo modo de acesso é fundamental para processadores que trabalham no modo serial-por-bit, nos quais cada ponto da imagem tem seus bits tratados seqüencialmente, e vários pontos são processados ao mesmo tempo. Assim, pode ser preciso num dado instante acessar o bit menos significativo de todos os pontos da imagem, em seguida o segundo bit de todos os pontos, etc..

A partir do conceito acima, foi concebida na década de 60 uma máquina denominada STARAN, que consistia de um conjunto de quatro módulos, com 256 processadores de um bit em cada módulo; todo o controle era exercido por um PDP-11, e a memória disponível (entre 64K bits e 64M bits) era acessada pelos processadores através de uma rede chaveada de acordo com um código de operação fornecido. Desta forma, tornava-se possível reconfigurar a região de memória "enxergada" por cada processador, podendo-se assim escolher a configuração mais adaptada para cada tipo de aplicação.

Os princípios utilizados no DAP e no STARAN serviram de base para o desenvolvimento de uma máquina extremamente poderosa para o tratamento de imagens orbitais, o MPP ("Massively Parallel Processor"), construído para a NASA já na década de 80. Com uma arquitetura bem avançada, ele contém uma matriz de 132x128 processadores, sendo que quatro processadores de cada fila atuam como reserva. Além dessa matriz, existe um módulo central de controle, um módulo para conexão a periféricos e a um computador convencional externo, e também circuitos de chaveamento destinados a transformar o formato dos dados que serão enviados aos processadores; externamente os dados estão no formato convencional (por palavras), enquanto que dentro da matriz de processadores eles devem estar no fórmato ortogonal (por planos de bits). Uma característica interessante do sistema é que, além das conexões habituais entre processadores, existe um barramento independente de E/S, o que permite que haja superposição entre processamento normal e operações de E/S. Outro aspecto importante é a existência de um registro de máscara em cada processador, que pode ser usado em certas ocasiões para se fazer com que alguns dos processadores deixem de operar durante alguns ciclos.

Com o atual avanço tecnológico, já estão surgindo no mercado internacional alguns novos componentes, desenvolvidos com técnicas VLSI, apresentando estruturas paralelas bastante interessantes para o tratamento de imagens. Um destes componentes, que inclusive foi a motivação principal do presente trabalho, é o GAPP ("Geometric Arithmetic Parallel Processor"), especificado pela Martin Marietta (EUA) e desenvolvido pela NCR (EUA), incorporando vários aspectos já presentes no MPP. A utilização deste componente torna viável o desenvolvimento de sistemas com alto desempenho e

baixo custo, pois podem contar com o uso de um microcomputador comum para a interação com os usuários e com os periféricos externos. Diante desta perspectiva, foi iniciado durante o ano de 1986 um trabalho visando o estudo deste componente e o desenvolvimento de um sistema que o utilizasse como elemento de processamento, conectado a um microcomputador do tipo IBM-PC e com aplicações em processamento de imagens. As principais características do componente, bem como do sistema proposto, são descritas a seguir.

2. CARACTERÍSTICAS DO PROCESSADOR GAPP.

O circuito integrado GAPP, projetado e construído pela Divisão de Microeletrônica da NCR (EUA), foi concebido tendo como principal objetivo a aplicação no reconhecimento de alvos em tempo real. Sua estrutura é constituída por várias células elementares idênticas interligadas entre si, com capacidade de funcionamento em paralelo no modo SIMD: todos os processadores executam a mesma instrução simultaneamente, cada um deles sobre um dado distinto. Há algumas das características típicas dos sistemas sistólicos, tais como: sincronismo fornecido por um relógio central, regularidade na estrutura, modularidade e ainda aumento linear de desempenho com o número de processadores. Existe, contudo, um desvio marcante do modelo sistólico, onde os processadores têm funções bem específicas: no GAPP os processadores são de propósito geral.

Na versão atual (GAPP II), que é a que está comercialmente disponível, cada componente é constituído por uma matriz de 6x12 células ou processadores elementares dispostos numa grade bidimensional regular. Pode-se agrupar vários componentes em cascata diretamente, a fim de se obter matrizes maiores, conforme a aplicação desejada. Idealmente, nas aplicações em processamento de imagens, esta matriz deveria ter as mesmas dimensões da imagem a ser tratada, para que cada célula ficasse responsável por um pixel da imagem; como isto, em geral, não é possível, devido ao excessivo número de componentes necessários, adota-se alguma forma de divisão da imagem em blocos menores, sendo então tais blocos processados um após o outro. Na próxima geração (GAPP III) cada componente deverá conter uma matriz de 8x16 células elementares, podendo-se então atingir matrizes maiores com um menor número de peças.

As células elementares já citadas têm a estrutura mostrada na

Fig. 1. Cada célula é na verdade um processador de bit, composto por: um somador completo de um bit, no qual são realizadas as operações aritméticas e lógicas; registros de um bit para armazenamento dos operandos, contendo na entrada multiplexadores que irão selecionar a origem dos dados a serem armazenados; e finalmente uma memória RAM com 128 posições de um bit, destinada a armazenar dados que estão sendo processados pela célula. O funcionamento do sistema é coordenado pelas linhas externas de controle $C_{\emptyset}C_1,\ldots,C_c$ e de endereço de memória $A_{\emptyset}A_1,\ldots,A_6$; todas as células da estrutura recebem o mesmo código, o que caracteriza o componente como sendo do tipo SIMD ("Single-Instruction/Multiple-Data"). Além disto, pode-se notar que o código de controle apresenta campos diferentes para cada um dos multiplexadores, o que torna possível a operação de vários deles simultaneamente. Desta forma, além do paralelismo intrínseco do componente (várias células processando ao mesmo tempo), existe a possibilidade de paralelismo dentro de cada célula (os registros podem ser carregados simultaneamente).

Como os processadores são capazes de operar apenas um bit, caso se queira realizar operações envolvendo dados de mais que um bit esta tarefa terá que ser feita por etapas, processando-se um bit a cada passo.

Um ponto bastante positivo na estrutura mostrada é a existência da memória de dados local em cada processador, a qual pode ser lida ou escrita a cada ciclo de operação, sem interferir com o processamento. Isto irá evitar o conhecido gargalo das máquinas tradicionais de Von Neuman, nas quais dados e instruções devem ser freqüentemente acessados numa memória externa, através de um barramento com largura de faixa limitada, sob a forma serial. Como no GAPP cada processador contém os dados de que necessita armazenados em sua própria memória local, não há o "overhead" que se encontra em sistemas típicos de multiprocessadores que disputam acesso a uma memória comum.

Cada célula pode comunicar-se com quatro células vizinhas: à esquerda, à direita, acima e abaixo, através de barramentos de comunicação N/S e E/W, responsáveis pelo fluxo de informações nas direções vertical e horizontal, respectivamente; estes barramentos estão ligados aos registros NS e EW de cada célula, e aos pinos externos do componente nas bordas da matriz, sendo bastante importantes para as transferências nas operações envolvendo vizinhanças.

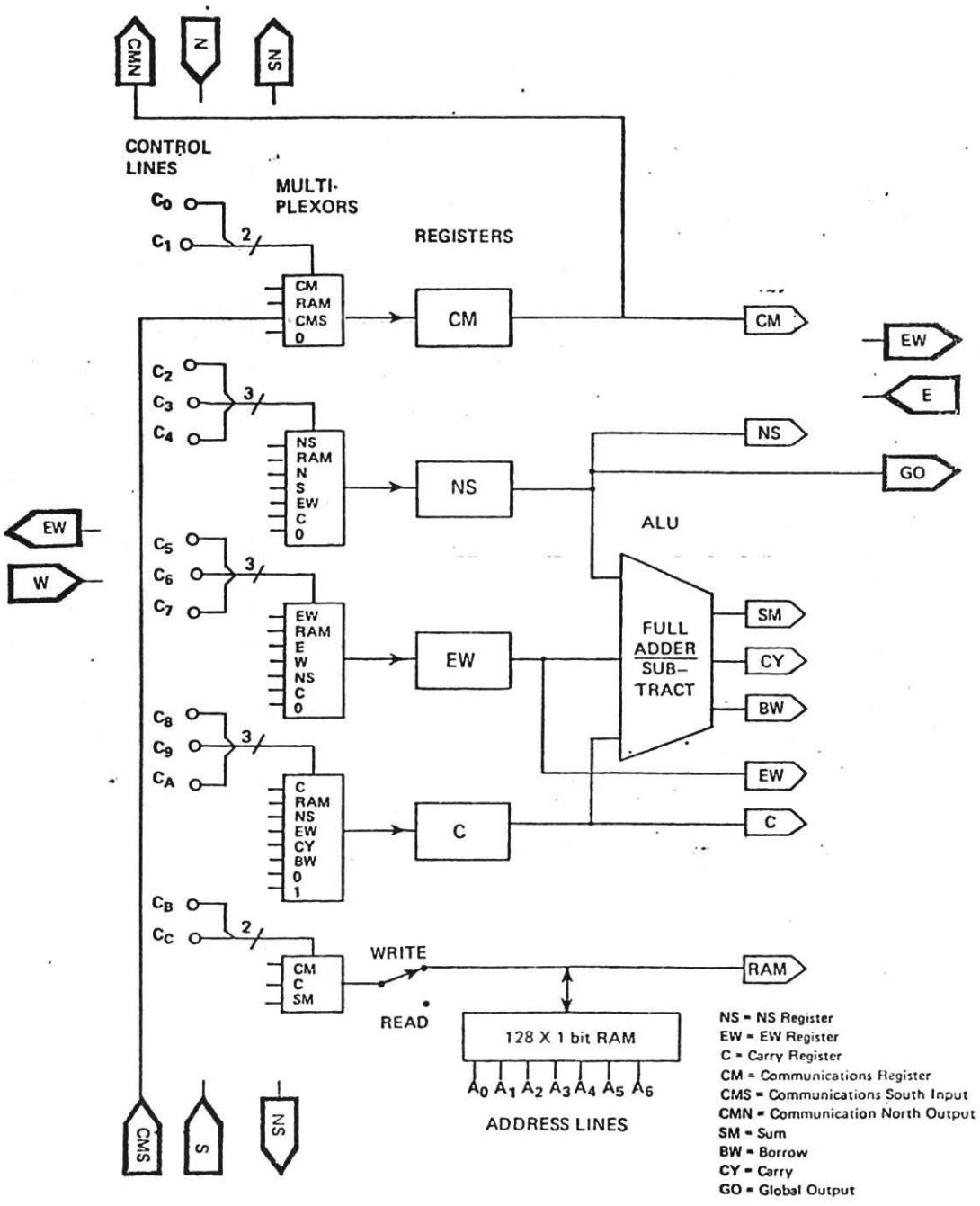

Fig. 1 - Estrutura do processador elementar.

Outro barramento existente na direção vertical é o de comunicação CMS/CMN, unidirecional (sentido Sul - Norte), interligando os registros CM de cada coluna de processadores, e que é utilizado para entrada/saída, sem influenciar o processamento no interior da célula; com isto, existe potencialmente a capacidade de paralelismo de atividades (processamento e entrada/saída simultâneos). Todos estes barramentos podem operar em paralelo (transferências de dados simultâneas), dependendo apenas dos códigos de controle selecionados. Durante as transferências por estes barramentos, os respectivos pinos de saída do componente receberão os dados do último nível de registros, a fim de alimentar os pinos de entrada de um componente vizinho, garantindo assim a modularidade do sistema geral.

Existe ainda a ligação de todos os registros NS de um componente (72 registros no caso do GAPP II) numa porta do tipo "OU", fornecendo, através de inversão, uma saída externa global "GO"; este sinal de saída pode ser bastante útil em determinadas aplicações, pois poderá fornecer de imediato uma indicação do comportamento de todo um conjunto de células, evitando que se tenha que consultar cada uma das células individualmente.

Quanto à velocidade de operação, há dois modelos disponíveis, com taxas máximas de 5 MHz e 10 MHz, obtendo-se assim ciclos de execução de 200 e 100 ns, respectivamente. A cada ciclo, as linhas de controle devem receber um certo código para seleção dos multiplexadores, o qual tem que ser fornecido por alguma estrutura externa.

De forma geral, pode-se dizer que o GAPP é um componente que torna possível o paralelismo em larga escala, como foi visto: paralelismo de células operando simultaneamente, de operações concorrentes dentro de cada célula, simultaneidade de entrada/saída e processamento e ainda paralelismo por trocas de informação através de barramentos independentes. Tudo isto indica que a sua utilização pode levar a altos níveis de desempenho, desde que se implemente um sistema cuja arquitetura seja conveniente para a aplicação a ser tratada.

Um aspecto bastante interessante derivado da modularidade do GAPP é que o desempenho aumenta na proporção direta do número de processadores que compõem o sistema, desde que se consiga manter todos os processadores realizando trabalho útil; vale lembrar ain-

da que o aumento de número de componentes não requer nenhuma modificação no software já desenvolvido.

A Tabela 1 mostra alguns índices de desempenho, os quais podem dar uma idéia comparativa entre sistemas de processamento tradicionais e sistemas baseados no GAPP. Deve-se ressalvar que estes índices referem-se apenas a taxas de processamento puro, sem considerar tempos de entrada/saída, já que estes últimos dependerão fortemente da arquitetura de cada sistema em particular.

PROCESSADOR	MULTIPLICAÇÃO PONTO FLUT./32 BITS	ADIÇÃO INTEIRO/8 BITS
VAX, 80286, 68020	10 a 200 MFLOPs	1 a 50 MIPS
CRAY X-MP, CYBER 205	100 a 400 MFLOPS	
1 GAPP (12x6)	0.2 MFLOPs	28 MIPS
32 GAPP's(48x48)	7.6 MFLOPs	900 MIPS
3 700 GAPP's (512 x 512)	870 MFLOPs	100 000 MIPS

TABELA 1 - ÍNDICE DE DESEMPENHO.

3. FERRAMENTAS DE PROGRAMAÇÃO.

Sendo o GAPP um dispositivo microprogramado, deve haver ferramentas de programação tanto para a compilação de programas como para o auxílio a testes de execução, as quais serão comentadas a seguir.

Por tratar-se de uma arquitetura bastante não-convencional, a programação do componente deve tentar aproveitar ao máximo os recursos de hardware disponíveis, a fim de se obter um bom desempenho em termos de tempo de rpocessamento. Para atingir tal objetivo, existem dois caminhos principais: um deles seria criar ferramentas que forneçam ao programador meios de trabalhar no nível mais interno possível, controlando até, por exemplo, as transferências de dados entre registros internos de cada célula. O outro caminho seria desenvolver compiladores para linguagens de alto nível com "inteligência" suficiente para explorar de forma eficaz todos os recur-

sos fornecidos pelo componente. Do ponto de vista da programação de aplicações, sem dúvida a segunda alternativa seria mais atraente, porém, devido a ser este um trabalho que visa também fornecer meios para uma maior familiarização com o componente, será abordada apenas a primeira linha de trabalho, ou seja, a implementação de ferramentas de baixo nível.

Seguindo assim a linha proposta, foram inicialmente desenvolvidos dois programas: um montador para geração de código objeto do GAPP e um simulador de uma estrutura com até 64x64 células elementares. Com estes programas, tornou-se possível o desenvolvimento de alguns algoritmos básicos, o que causou vários benefícios importantes: foi possível verificar os pontos positivos e negativos do GAPP em diversos tipos de aplicações; tornou-se viável a reconstrução de programas normalmente utilizados em máquinas seriais, dando uma idéia dos desempenhos comparativos; e como tais ferramentas podem ser usadas por outras pessoas, passou a estar disponível um ambiente prático para o aprendizado inicial sobre o funcionamento do componente.

Os dois programas foram desenvolvidos em linguagem C, num microcomputador de 16 bits compatível com o IBM-XT, e a estrutura de funcionamento está mostrada na Fig. 2; a entrada principal é um arquivo com o programa-fonte a ser simulado. Como saídas do montador são gerados os arquivos com o código objeto do programa, um arquivo indicando possíveis "break-points" assinalados pelo usuário e uma listagem do programa montado, com o respectivo código binário gerado. As duas primeiras saídas citadas servem como entrada do simulador, o qual deve receber ainda um terceiro arquivo contendo os dados externos de entrada (barramento N/S, E/W e CMS/CMN), além de um possível arquivo contendo dados para inicialização das memórias internas de cada processador. A saída da simulação consta de dois arquivos: um deles contém o estado interno dos processadores nos instantes solicitados, enquanto que o segundo contém os dados dos barramentos de saída nestes mesmos instantes.

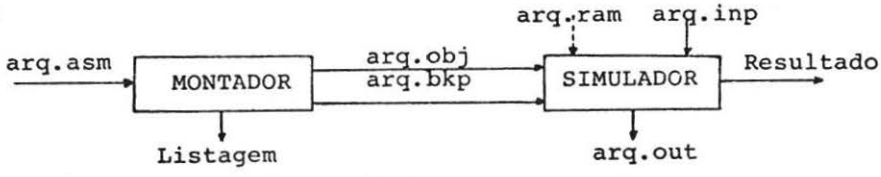

Fig. 2 - ESTRUTURA DO CONJUNTO MONTADOR-SIMULADOR.

Existem algumas facilidades embutidas na estrutura dos programas, visando uma maior operacionalidade dos mesmos: no montador é possível inserir comentários junto com o programa-fonte, assim como instruções de "break-point"; no simulador pode-se escolher o tamanho da matriz a ser simulada, o modo de inicialização das memórias internas (zeradas, por arquivos externos ou com dados aleatórios), e ainda o modo de simulação: se com interrupções após cada instrução simulada ou apenas nos "break-points" indicados no programa-fonte; a cada interrupção, são apresentados os conteúdos de todos os registros internos e todas as posições de memória, assim como os sinais gerados nos barramentos externos nestes instantes.

4. EXEMPLOS DE APLICAÇÃO.

A seguir são descritos alguns programas simples que foram implementados para servir de exemplo sobre como o componente pode ser programado, e ainda possibilitar uma avaliação preliminar sobre suas características.

A - Soma de dois números:

Este programa tem a finalidade de somar dois números inteiros de oito bits, gerando um resultado de até nove bits; supõe-se que os operandos estejam na memória interna dos processadores, na qual também será guardado o resultado. A soma é feita passo a passo, sendo a cada passo somados os bits de uma certa ordem. No total são gastos 25 ciclos, o que corresponderia ao tempo de 2,5 μs na versão mais rápida do componente (10 MH).Como cada componente tem 72 processadores operando simultaneamente, pode-se ter 72 somas em 2,5 μs, ou $28,8.10^6$ somas por segundo, isto é, mais de 28 MIPS por componente. É claro que tal valor trata-se apenas de um limite teórico, o qual dificilmente será atingido nas aplicações reais, já que não está sendo contado o tempo de E/S necessário para alimentar a matriz com os dados externos a serem processados.

B - Limitação de valores ("Clampeamento")

Nesta operação os valores de cada pixel são comparados com um limiar de decisão; caso sejam menores ou iguais, os valores originais são mantidos; caso contrário, são substituídos por este valor de decisão, que será assim o máximo valor da imagem resultante.

Esta operação serve como exemplo das chamadas operações "ponto-a-ponto", já que o processamento de cada pixel independe dos valores vizinhos. Um detalhe importante é que a decisão ocorre em instantes distintos para os vários processadores, porém, como todos eles executam o mesmo programa, o algoritmo deve ser tal que os passos finais não "perturbem" o resultado dos processadores que já tenham feito sua decisão. O tempo total de execução é de 50 ciclos, isto é, 5μs, devendo-se novamente lembrar que não foi considerado o tempo de E/S.

C - Erosão Binária.

Este tipo de operação constitui um procedimento básico na área de análise de imagens binárias, sendo bem semelhante ao processo de convolução em duas dimensões: a imagem original é varrida por uma máscara escolhida previamente e a cada posição a região de imagem correspondente é comparada ponto-a-ponto com a máscara; dependendo do resultado da comparação, o ponto original da imagem é modificado ou não. Como o valor final de cada ponto depende dos pontos vizinhos, torna-se necessário transferir dados entre processadores, o que pode ser facilmente realizado com os barramentos E/W e N/S. O número de ciclos necessários foi de 66, correspondendo ao tempo de 6,6 μs.

Deve ser notado que nos casos em que o tamanho da matriz de processadores disponíveis for menor que a imagem a ser analisada, esta deve ser dividida em várias sub-imagens com sobreposição de dois pontos de borda, já que tais pontos serão necessários para o processamento do vizinho imediato. As diversas sub-imagens seriam então processadas em seqüência.

COMENTÁRIOS.

Pelos exemplos mostrados, pode-se ver que a estrutura do componente GAPP é bastante flexível, caracterizando este componente como sendo de propósito geral: é possível realizar-se tanto as operações "ponto-a-ponto" como as operações envolvendo vizinhanças. Existem alguns pontos bastante interessantes na sua arquitetura, apesar de haver também algumas deficiências, causadas talvez pela necessidade de implementação física do componente.

Dentre os pontos positivos, destaca-se o alto grau de paralelismo existente, permitindo que as operações de E/S tenham um im

pacto mínimo sobre o processamento. A possibilidade de comunicar dados entre processadores vizinhos mostra-se bastante conveniente em certas aplicações. Apesar da simplicidade da unidade lógica e aritmética, é possível realizar um variado conjunto de operações, através da progr mação adequada. Outro ponto forte da arquitetura é a sua modularidade, podendo-se conectar vários componentes entre si para formar matrizes com tamanhos desejados.

Quanto aos pontos negativos, deve ser citada a falta de um registro de máscara, existente no MPP, que serviria para fazer parar o funcionamento de um processador sob determinadas condições; sem isto, alguns algoritmos, como o de "Grampeamento", devem ser aumentados de modo a garantir que algumas das instruções não afetem resultados já calculados; isto sempre irá ocorrer nos casos em que o processamento de cada pixel tiver um tempo variável para cada processador. Outra questão duvidosa é o tamanho da memória local de cada célula, que pode ser pequena para aplicações que necessitem armazenar números reais, ou nos casos em que o processamento de cada pixel dependa de vários parâmetros que devam ser armazenados, como no caso de uma classificação estatística. O número de processadores por componente, apesar de razoável, faz com que o número de componentes necessários para tratar as imagens comuns (512x512) em tempo real seja ainda excessivamente elevado (da ordem de 3 700 componentes).

De qualquer forma, parece bastante conveniente a construção de sistemas baseados no GAPP, já que as futuras versões do componente deverão superar algumas das deficiências atuais, esperando-se que sejam compatíveis ao menos a nível de programação com as versões atuais.

5. PROPOSTA DE ARQUITETURA.

A seguir é apresentada a proposta de uma arquitetura para um sistema baseado no GAPP e que possa ser conectado a um microcomputador do tipo IBM-PC. O sistema proposto, apesar de ser de propósito geral, deverá ser capaz de atender às aplicações já mostradas em processamento de imagens, podendo ser incorporado a futuros membros da família SITIM que vem sendo desenvolvida no INPE.

O diagrama de blocos do sistema está mostrado na Fig. 3. A matriz de processadores teve seu tamanho escolhido de modo a facilitar a implementação física, contando com seis GAPPs numa organiza

ção de duas linhas de três componentes cada. Desta forma, para se processar imagens normais de tamanho 512x512 será necessário dividí-las em sub-imagens de tamanho 24x24, as quais serão processadas em seqüência. Nas operações ponto-a-ponto deverá haver $22^2 = 484$ sub-imagens, enquanto que nos processos envolvendo vizi

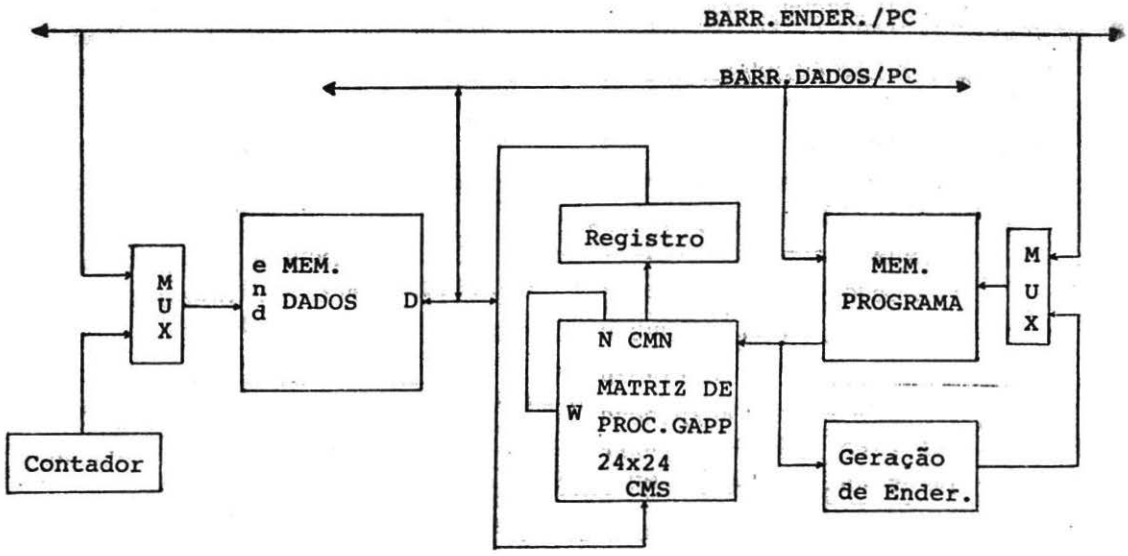

Fig. 3 - Arquitetura proposta.

nhos este número deverá subir para $23^2 = 529$, devido à sobreposição de dois pontos entre sub-imagens vizinhas como já foi comentado.

O funcionamento dos processadores é comandado por instruções geradas por uma unidade de controle, composta por uma memória de microprograma e um módulo de geração de endereço de microinstruções; cada microinstrução é formada por 32 bits, dos quais treze poderão fornecer um código de operação à matriz de processadores, sete formarão o endereço da memória local dos processadores e os demais bits irão conter um código de operação para a estrutura de controle, de modo a controlar a execução de um programa, permitindo a operação seqüencial ou desvios comandados por certas condições, tais como o estado da saída global da matriz de processadores. A memória de programa poderá ser acessada também pelo microcomputador, que deverá carregar o código dos programas a serem executados; o código-objeto de tais programas será gerado por um montador a ser

implementado no microcomputador.

O outro bloco importante do sistema é a memória de dados, destinada a armazenar pedaços da imagem a ser processada; o acesso a esta memória também é duplo, pelo microcomputador e pela matriz de processadores, a qual fará operações de entrada/saída para acessar tal memória, com o endereço podendo ser incrementado ou decrementado automaticamente. Como tais operações de entrada/saída inserem uma fila de dados no lado inferior da matriz (vindos de uma leitura da memória) e fornecem uma fila de dados no lado superior, pelo barramento CMS/CMN, é preciso haver um registro que armazene tais dados para que possam ser escritos na memória num ciclo seguinte.

Existe ainda na matriz de processadores uma conexão externa entre os barramentos N/S e E/W; esta conexão tem a finalidade de permitir a reformatação dos dados no interior da matriz, segundo o conceito de "ortogonalidade" já citado, permitindo assim que na memória externa seja utilizado o formato convencional, isto é, todos os bits de um pixel em cada posição, e nas memórias internas o formato de planos de bit, ou seja, os bits de mesma ordem de todos os pixels em cada endereço. Embora isto acarrete num processamento a mais a ser realizado pela matriz, esta opção foi feita para dispensar o uso de circuitos complexos que permitissem o acesso à memória de dados nos dois modos, como ocorre no MPP.

Encontra-se em desenvolvimento atualmente um simulador do sistema proposto, com a finalidade de depurar a estrutura apresentada e permitir o desenvolvimento de programas antes mesmo da montagem de um protótipo. No futuro, poderia-se pensar em desenvolver compiladores de alguma linguagem de alto nível utilizando de forma eficiente os recursos disponíveis nesta estrutura, de modo a facilitar a programação do sistema. Outra possível atividade relacionada no trabalho seria o desenvolvimento de um componente similar, utilizando as técnicas de projeto VLSI já disponíveis no país.

6. CONCLUSÕES.

As técnicas de processamento paralelo, principalmente a utilização do modo SIMD, têm se mostrado como a mais eficiente alternativa para o processamento da quantidade cada vez maior de dados presentes em imagens. Com o surgimento de componentes comerciais empregando esta arquitetura, já é possível o desenvolvimento de módulos de processamento bastante rápidos, os quais podem ser associa-

dos a computadores hospedeiros de propósito geral para formar um conjunto de alto desempenho e flexibilidade suficiente para ser utilizado em várias aplicações.

O componente utilizado neste trabalho, apesar das limitações em sua versão atual, mostra-se como bastante atraente para o tratamento de imagens em geral, onde grande parte das operações são feitas com números inteiros; ele é a opção ideal para o tratamento de imagens binárias, já que o tempo de processamento depende do número de bits por pixel.

7. AGRADECIMENTOS.

Ao Departamento de Eletrônica Aplicada do ITA, e em particular ao Dr. Edward W. Davis (North Carolina State University, convênio ITA/IBM) pela valiosa orientação e incentivo, sem os quais este trabalho não teria sido possível.

8. REFERÊNCIAS.

SOUZA, R.C.M., et Al. - "Evolução da Família de Sistemas de Tratamento de Imagens do INPE", Simpósio Latino-Americano de Sensoriamento Remoto, Gramado, RS, 1986.

DAVIS, R. & THOMAS, D. - "Systolic Array Chip Matches the Pace of High-Speed Processing", Electronic Design, October 31, 1984.

STRONG, J.P. - "Basic Image Processing Algorithms on the Massively Parallel Processor", em Multicomputers and Image Processing, Academic Press, 1982.

HOCKNEY, R.W. & JESSHOPE, C.R. - "Parallel Computers", Adam Hilger Ltd., Bristol, 1981.

KUNG, H.T. - "Why Systolic Architectures?", IEEE Computer, Jan. 1982.