CAPÍTULO II

ESTUDO DAS ARQUITETURAS SIMD

II.1 - Introdução

A primeira proposta de uma arquitetura com padrão SIMD surgiu ainda no final dos anos 50. Num artigo publicado em Outubro de 1958 por S. H. Unger [Unge58], pesquisador da Bell Telephone Labs., foi apresentada uma arquitetura especialmente concebida para resolver problemas de reconhecimento de padrões. Mais do que isto, ela visava servir de exemplo de como construir máquinas capazes de resolver vários problemas que não têm uma boa solução através dos computadores convencionais.

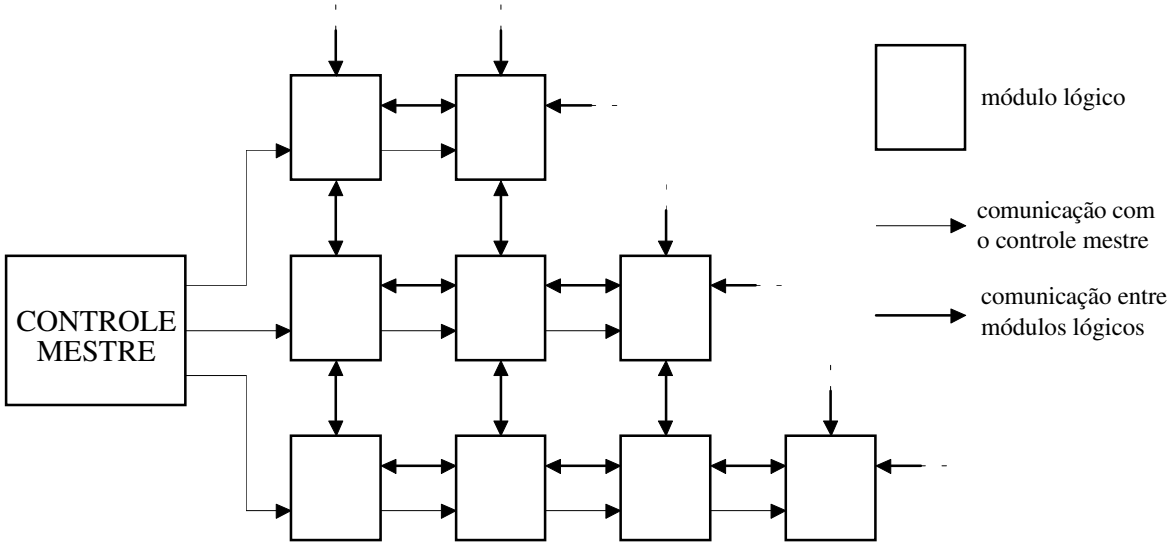

Figura II.1 - Organização espacial

Embora a máquina proposta jamais tenha sido construída, devido às restrições tecnológicas dos dispositivos eletrônicos disponíveis na época, o objetivo didático deste artigo foi perfeitamente alcançado, já que ele realmente serviu de inspiração para muitas arquiteturas posteriores. Pela proposta, a máquina é composta por um controle mestre e uma matriz de módulos. Cada módulo se comunica com os seus quatro vizinhos imediatos (cima, baixo, esquerda e direita) e recebe ordens do controle mestre (Figura II.1). O controle mestre não pode endereçar os módulos individualmente, somente enviar comandos gerais para todos os módulos. Ou seja, todos os módulos sempre executam exatamente a mesma operação, porém sobre dados diferentes.

Cada módulo consiste de um registrador de 1 *bit* utilizado como acumulador, uma pequena memória de acesso randômico (algo em torno de 6 *bits*) e alguma lógica (Figura II.2). Existe ainda um somador lógico (porta OU) dos acumuladores de todos os módulos, capaz de informar ao controle mestre se todos os acumuladores têm conteúdo zero. O controle mestre atua basicamente como um seqüenciador de Von Neumann, com uma exceção: a capacidade de tomar decisões baseado no resultado do somador lógico. Esta exceção é uma das inovações mais importantes da proposta de Unger. O controle mestre é composto por: uma memória de acesso randômico para o armazenamento de instruções, um relógio e circuitos de decodificação.

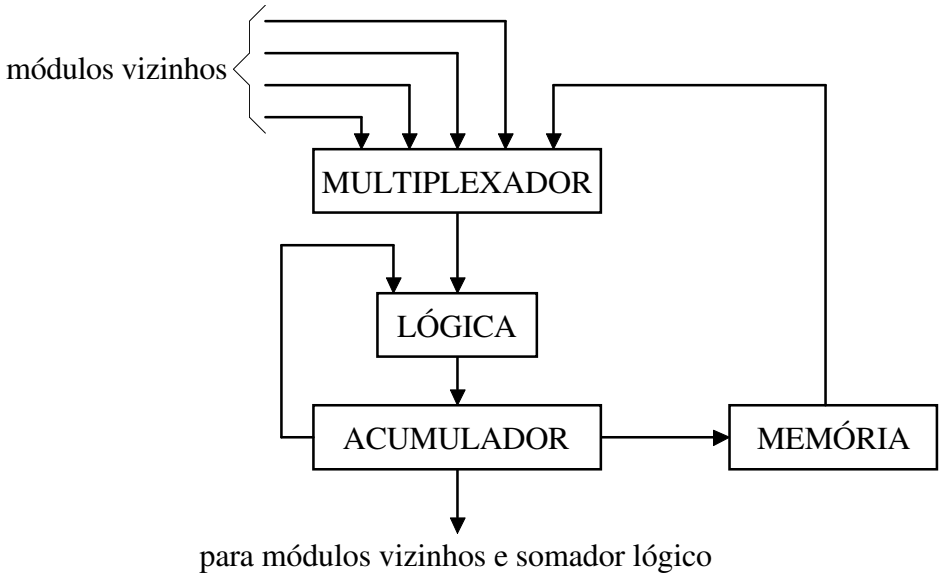

Figura II.2 - Estrutura lógica do módulo

A entrada e saída de dados na matriz é feita de forma serial. Um *bit* é transmitido ao módulo do canto inferior esquerdo da matriz pelo controle mestre. Ao mesmo tempo, o conteúdo do acumulador de cada módulo é transmitido para o módulo a sua direita. O módulo mais à direita de cada linha, transmite o conteúdo do seu acumulador para o módulo mais à esquerda da linha de cima. O módulo do canto superior direito transmite um *bit* para o controle mestre. Desta forma, pode-se carregar uma matriz simultaneamente ao processo de descarga de outra.

A arquitetura tem ainda uma memória de ligações. Entre cada módulo e cada um de seus oito vizinhos (cima, baixo, esquerda, direita e diagonais) existe um circuito lógico de ligação que, quando ativado, armazena o valor 1 (um) se o módulo e seu vizinho têm seus acumuladores com conteúdo 1 (um). Esta memória de ligações tem a sua origem na aplicação específica da máquina, que é o reconhecimento de padrões.

A arquitetura proposta é bastante limitada: a capacidade de armazenamento (memória) associada a cada módulo é muito pequena, as operações lógicas implementadas são muito restritas, os módulos não possuem qualquer grau de independência (não têm a capacidade de tomar qualquer tipo de decisão) e a entrada e saída de dados é muito lenta. Porém, a importância do artigo encontra-se nas idéias da estrutura da arquitetura propostas pelo mesmo. Nas arquiteturas paralelas SIMD atuais, são encontradas algumas idéias propostas por Unger, como: um controle central e vários elementos processadores simples interligados executando a mesma instrução (estrutura básica de uma arquitetura SIMD) e os resultados de operações nos elementos processadores influenciando a tomada de decisões pelo controle central.

A seguir são apresentadas diversas arquiteturas SIMD propostas nos últimos 30 anos. Cada uma delas apresenta inovações e repetições de idéias das anteriores, mostrando a evolução histórica deste tipo de arquitetura.

II.2 - SOLOMON (*Simultaneous Operation Linked Ordinal MOdular Network*)

O SOLOMON [Slot62, Greg63] foi concebido no início da década de 60, por pesquisadores da Westinghouse Electric Corporation. Da mesma forma que o computador proposto por Unger, o SOLOMON jamais foi construído, mas suas idéias influenciaram arquiteturas posteriores. O SOLOMON é composto por uma Unidade de Controle e uma Matriz de Processadores contendo 1024 Elementos Processadores (sistema básico), organizados em 32 linhas e 32 colunas.

Os Elementos Processadores (EPs) são interligados fisicamente na configuração de uma matriz. Cada EP tem dois canais de comunicação de 1 *bit* com os seus vizinhos de cima (Norte), de baixo (Sul), da direita (Leste) e da esquerda (Oeste). As bordas da matriz permitem vários modos de interconexão. É importante ressaltar que o modo de interconexão é geral para toda a matriz. Existem três esquemas básicos de configuração permitidos. O primeiro permite a conexão entre o primeiro e o último EP de cada linha (configuração cilíndrica "horizontal"). O segundo permite a conexão entre o primeiro e o último EP de cada coluna (configuração cilíndrica "vertical"). O último esquema permite ligar os EPs na forma de um vetor unidimensional (último EP de cada linha ligado ao primeiro EP da linha seguinte); formando um circuito fechado, caso desejado. É permitida a configuração por um ou mais dos esquemas apresentados acima. A escolha da configuração depende do problema em questão.

Além da comunicação entre processadores, todos os EPs são ligados a barramentos comuns, através dos quais eles podem receber dados comuns, simultaneamente. Existe um barramento global (B) e barramentos para cada linha da matriz (LB). Estes canais de comunicação entre os processadores e os barramentos comuns são de 1 *bit*. Estes barramentos

comuns são uma idéia simples, porém muito útil. Sem o barramento global, o carregamento de um dado comum a ser utilizado em uma operação seria muito mais lento. Os barramentos para cada linha permitem o carregamento de dados diferenciados para cada linha da matriz de processadores.

Cada EP é um processador serial de 1 *bit*, desprovido de seqüenciador de instruções ou de decodificador lógico de instruções. A Unidade de Controle do SOLOMON se encarrega de enviar os controles da instrução a ser executada, diretamente, a todos os EPs.

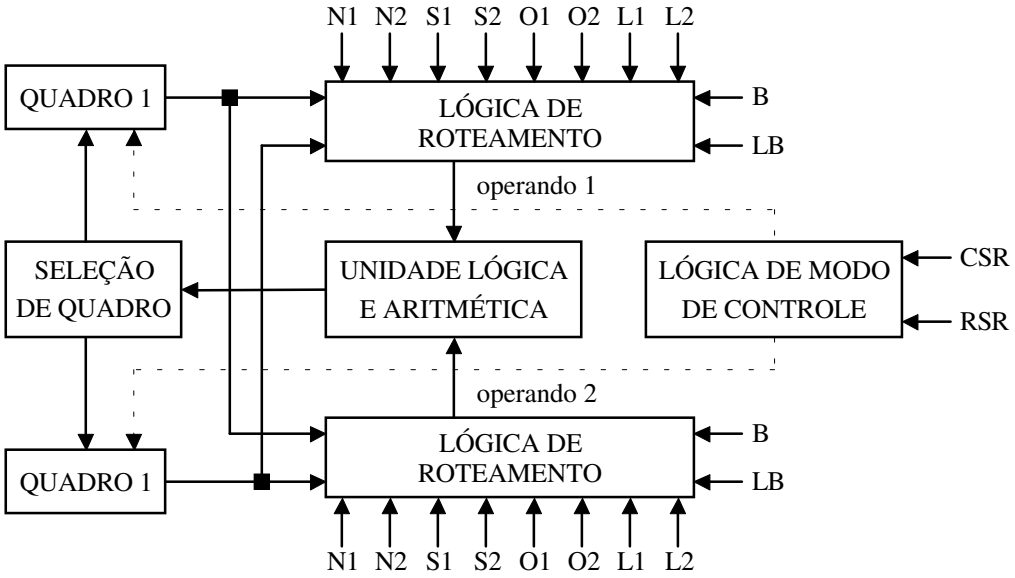

Figura II.3 - Diagrama funcional do elemento processador do SOLOMON

Cada EP (Figura II.3) contém dois módulos de memória de acesso randômico, denominados quadros, usados para o armazenamento de dados. Cada quadro é composto por uma memória com capacidade para o armazenamento de 4096 *bits* (expansível para 16384 *bits*). Além disso, cada EP possui uma lógica de execução de instruções, uma Unidade Lógica e Aritmética, duas lógicas de roteamento, uma lógica de seleção de quadro e um registrador de modo. Um protótipo de um EP do SOLOMON foi construído numa placa de 5 por 7 polegadas.

A Unidade Lógica e Aritmética é responsável pela execução das operações aritméticas, booleanas e de comparação entre dois operandos. Cada lógica de roteamento controla a origem de um dos operandos, que podem ser: dados lidos de um dos quadros do próprio EP, dados lidos de um dos quadros de um dos 4 vizinhos do EP, ou dados vindos por um dos barramentos comuns. A lógica de seleção de quadro controla em que quadro será gravado o resultado da operação. O grande problema desta estrutura é que o operando contido no quadro onde o resultado será armazenado é perdido, já que o endereço de leitura e escrita de cada quadro é o mesmo. Além disso, a existência de dois canais de comunicação para cada par de

vizinhos, aumenta muito a complexidade das conexões e do roteamento, trazendo como única vantagem a possibilidade de transmitir 2 *bits*, simultaneamente, para cada vizinho.

Todos os EPs recebem o mesmo controle, porém eles possuem a capacidade de executar ou ignorar uma instrução. Isto confere uma independência maior ao EP, que faltava na proposta de Unger. O SOLOMON permite duas formas de inibição de uma instrução. A primeira forma é através do registrador de modo (2 *bits*), que contém o "estado interno" do EP. O seu conteúdo pode ser alterado de acordo com o resultado de uma operação. A Unidade de Controle pode especificar para que valores do registrador de modo a instrução deve ser executada. A segunda forma é através de sinais de seleção de linha (RSR) e de coluna (CSR). Esses sinais permitem a inibição de linhas de EPs e de colunas de EPs. Logo, pode-se inibir a execução de uma instrução tendo como base a posição do EP dentro da matriz ou o seu estado interno (resultado de operações anteriores). Esta capacidade de tomar decisões conferida ao elemento processador é, provavelmente, a mais importante inovação encontrada no SOLOMON. Através dela é possível executar instruções condicionais, ou seja, fica fácil criar uma estrutura do tipo SE expressão ENTÃO tarefa, onde a expressão pode ser tanto a posição do EP quanto o resultado de uma operação envolvendo dados próprios de cada EP.

II.3 - ILLIAC IV

O ILLIAC IV [Barn68], apesar de ter sido concebido ainda na década de 60, é uma proposta bem ousada, até mesmo para os dias de hoje. Ao contrário das duas arquiteturas apresentadas anteriormente, que não chegaram a ser implementadas, o ILLIAC IV foi construído em 1972 para a NASA Ames. Este protótipo único foi utilizado com sucesso por pesquisadores da época, através de um serviço oferecido pela rede ARPANET. Embora a especificação completa do projeto jamais tenha sido construída, o ILLIAC IV foi de grande importância para o desenvolvimento das máquinas SIMD, principalmente no que se refere às linguagens paralelas. Diversas linguagens paralelas de programação foram concebidas originariamente para o ILLIAC IV.

O ILLIAC IV é composto por 256 EPs (quatro vezes menos do que o SOLOMON), organizados em 4 vetores de 64 EPs. Os EPs podem ainda ser unidos, formando 2 vetores de 128 EPs ou um único vetor de 256 EPs. Isso traz duas vantagens: uma melhor adaptação do vetor de processadores ao tamanho do problema a ser resolvido e uma melhor tolerância a falhas. Em relação à tolerância a falhas, esta ainda é uma solução precária, pois a falha de um único processador exige a inutilização de um bloco de 64 processadores (um quarto do total). Na verdade, apenas um vetor de 64 EPs foi construído.

A maior inovação do ILLIAC IV encontra-se no elemento processador, que é um processador que opera dados de 64 *bits*. Para não perder muita flexibilidade, já que nem todos

os problemas necessitam de 64 *bits* de precisão, cada processador pode ser partido em 2 processadores de 32 *bits* ou 8 processadores de 8 *bits*, produzindo 512 processadores de 32 *bits* ou 2048 processadores de 8 *bits*. Nestes casos, por uma restrição do projeto, os processadores perdem parte de sua independência. Isto é, ao dividir um processador, ganha-se flexibilidade no tamanho dos operandos, mas perde-se potencialidade no roteamento e na capacidade individual de tomar decisões.

Cada EP de índice i tem conexões com os EPs de índice i+1, i-1, i+8, i-8. Estas conexões, fazem com que cada vetor do ILLIAC IV atue, na verdade, como uma matriz. Desta forma, estas conexões correspondem as ligações Norte, Sul, Leste e Oeste encontradas no SOLOMON. Como desvantagem, o ILLIAC IV possui apenas um barramento comum de dados.

Cada EP possui 4 registradores de 64 *bits*, um somador/multiplicador, uma Unidade Lógica e um registrador de deslocamento. A cada EP está associada uma memória de acesso randômico com capacidade para 2048 palavras de 64 *bits*. Além disso, para controle e modificação do endereço de acesso à memória, cada EP possui um registrador de 16 *bits* e um somador. Essa inovação do ILLIAC IV permite que cada EP tenha acesso a uma posição de memória diferente.

Da mesma forma que o SOLOMON, o ILLIAC IV também possui um registrador de modo, mas de 8 *bits*. O *flip-flop ENABLE*, permite a habilitação ou não de certas instruções.

II.4 - MPP (*Massively Parallel Processor*)

O MPP [Batc80] foi projetado pela Goodyear Aerospace Corporation. O projeto foi encomendado pela NASA em 1979 e obteve grande sucesso.

A Unidade Vetorial (ARU) do MPP é composta por 16.384 elementos processadores organizados na forma de uma matriz com 128 linhas e 128 colunas. O MPP apresenta uma solução prática para uma importante preocupação em relação as máquinas SIMD, que é a tolerância a falhas. Fisicamente, ele possui 128 linhas e 132 colunas. Essas colunas extras servem para reconfigurar a ARU no caso de falha de algum processador. Cada grupo de 4 colunas possui uma linha de controle de desabilitação do grupo, ativada pela Unidade de Controle Vetorial (ACU). Para controlar o perfeito funcionamento das memórias associadas a cada EP, cada grupo de 2 linhas e 4 colunas possui um *bit* de paridade. Em caso de detecção de erro, um *flip-flop* associado a esse grupo é carregado com valor 1. Uma "árvore" lógica formada com portas OU-inclusivo, que recebe dados de todos esses *flip-flops*, fornece à Unidade de Controle do Vetor (ACU) a indicação de erro. Através de operações de desabilitação de grupos, a ACU pode localizar o grupo que contém o erro e desabilitá-lo.

Cada EP se comunica com seus vizinhos de: cima, baixo, direita e esquerda. Nas extremidades da matriz, as conexões são configuráveis, da mesma forma que o SOLOMON. Assim como na arquitetura proposta por Unger, todos os processadores tem uma saída para uma "árvore" OU-inclusivo. O acesso à saída dessa árvore é feito através da ACU. Isso permite que a ACU tome decisões baseada em resultados obtidos pelos EPs.

Os EPs são processadores de 1 *bit*, com ciclo de 100 ns. Cada grupo de 2 linhas e 4 colunas, com exceção da memória associada a eles, é encapsulado em um único *chip*. Para processadores de 1 *bit*, os EPs do MPP são bastante poderosos (Figura II.4). Eles possuem 6 registradores de 1 *bit* (A, B, C, G, P e S), um somador completo, lógica combinatória capaz de executar qualquer função de duas entradas, uma memória de acesso randômico de 1024 *bits* (expansíveis para 65536 *bits*), um registrador de deslocamento de tamanho variável e uma porta OU-exclusivo ligando os registradores P e G ao barramento principal. Para as memórias, são usados *chips* de memória contendo 1024 palavras de 4 *bits*, servindo a 4 EPs. Cada placa de 8 1/2 por 12 polegadas do MPP contém 192 EPs, configurados em 16 linhas e 12 colunas. Para construir a Unidade Vetorial do MPP são necessárias 88 placas como as descritas acima.

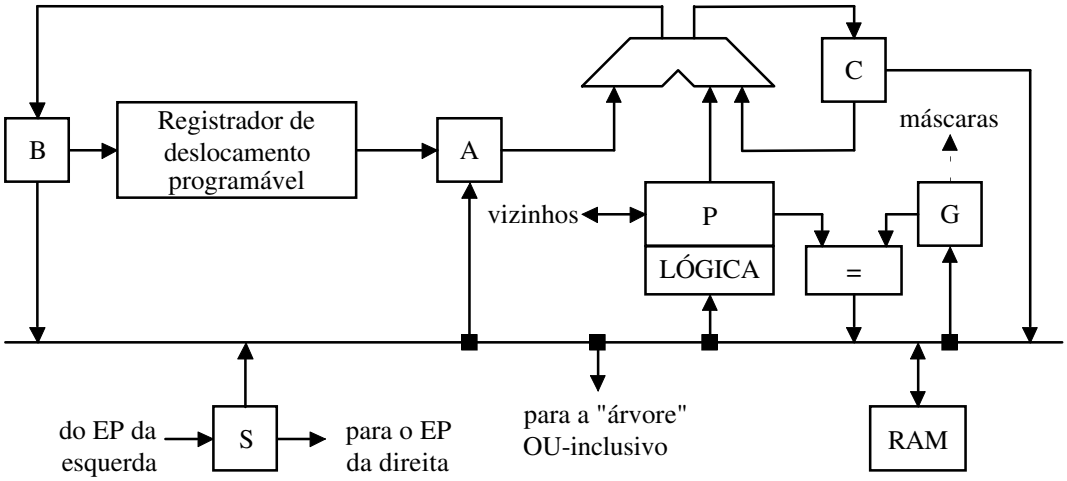

Figura II.4 - Diagrama funcional do elemento processador do MPP

Um dos registradores (G) tem a mesma função do registro de modo do SOLOMON. As instruções "mascaradas" só são executadas nos EPs cujo conteúdo do registrador G é 1. Esse registrador pode ser carregado com o valor de qualquer outro dispositivo que tenha uma saída ligada ao barramento principal do EP (registradores B, C, S, P ou conteúdo de alguma posição da memória).

Em relação à aritmética, a grande inovação do MPP é o uso de um registrador de deslocamento de tamanho variável para o armazenamento dos resultados parciais de algumas operações, como multiplicações e divisões, diminuindo o número de acessos à memória. Este registrador pode ter largura de: 2, 6, 10, 14, 18, 22, 26 ou 30 *bits*. Considerando-se que existe

um registrador de 1 *bit* na entrada do registrador de deslocamento (B) e um na saída (A), a capacidade de armazenamento de resultados parciais pode ser de: 4, 8, 12, 16, 20, 24, 28 ou 32 *bits*. Ou seja, os valores foram projetados para otimizar a operação dos dados com tamanho múltiplo de 4, embora não inviabilize a operação de dados com outros tamanhos. A largura do registrador de deslocamento é especificada através de uma instrução enviada pela ARU. O registrador variável é composto por cinco registradores de deslocamento de: 16, 1, 8, 1, e 4 *bits*, ligados em série. Com exceção dos dois registradores de 1 *bit*, todos os demais podem ser desativados, através de uma ligação direta entre os registradores anterior e posterior, como mostra a Figura II.5. É desta forma que a largura total do registrador é variada. As operações de deslocamento só ocorrem nos registradores ativos, o que significa que os dados armazenados nos registradores inativos se mantêm inalterados.

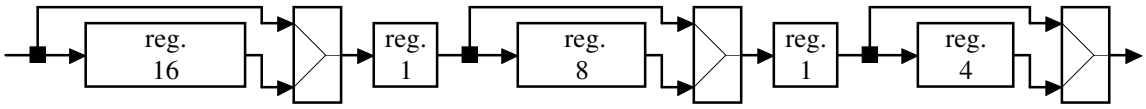

Figura II.5 - Estrutura do registrador de deslocamento programável

Outra importante inovação do MPP se refere à entrada e à saída de dados na Unidade Vetorial (ARU). Para isso é utilizado um registrador específico (S). Os registradores S são interligados da seguinte forma: cada registrador recebe dados do EP a sua esquerda e envia dados ao EP a sua direita. Além disso, cada registrador S pode enviar e receber dados do barramento principal do EP. Essa configuração dos registradores S permite que operações de entrada e saída de dados da ARU possam ser executadas simultaneamente com outras instruções do EP. Nos extremos esquerdo e direito da ARU são encontradas memórias denominadas *Staging Memories*. Essas memórias são diretamente ligadas aos registradores S dos extremos da ARU. A da esquerda é utilizada para a entrada de dados e a da direita é utilizada para a saída de dados. Elas têm duas funções: armazenamento e reordenação dos dados. Elas são implementadas com grandes memórias de acesso multidimensional. Esse esquema de entrada e saída permite um fluxo de até 256 *bits* por ciclo (128 de entrada e 128 de saída).

II.5 - DAP (*Distributed Array Processor*)

O DAP [Redd79, Hock88] foi desenvolvido em 1979 na ICL (International Computer Ltd. of Stevenage - Inglaterra). Ele tem um número variável de processadores, mas a sua primeira versão foi formada por 4096 EPs organizados na forma de uma matriz com 64 linhas e 64 colunas.

Os EPs são processadores de 1 *bit* com ciclo de 200 ns. Cada grupo de 4 EPs (com exceção das memórias associadas a eles) é encapsulado em um único *chip*, com 28 pinos. A

matriz de processadores do DAP é composta por placas de 12 por 7 polegadas. Cada placa do DAP contém uma submatriz com 4 linhas e 4 colunas. Isto significa que, para compor toda a matriz de processadores, são necessárias 256 placas.

Cada EP se comunica com os seus quatro vizinhos diretos e com dois barramentos comuns denominados: *Column Highway* e *Row Highway*. Através desses barramentos comuns, que são um aprimoramento dos barramentos do SOLOMON, pode-se carregar ou ler dados de uma linha inteira ou coluna inteira, simultaneamente. Todos os canais de comunicação são de 1 *bit*. As conexões nas bordas da matriz são configuráveis para cada instrução. As bordas das linhas têm dois tipos de configuração: planar (todas as bordas tem entrada igual a zero) e cíclica (primeiro elemento de cada linha ligado ao último elemento da mesma linha). Da mesma forma, as bordas das colunas também possuem dois tipos de configuração: planar e cíclica.

Cada EP (Figura II.6) possui dois multiplexadores, uma Unidade Lógica e Aritmética (ULA) e uma memória de acesso randômico (4 *kbits* na primeira versão e 16 *kbits* posteriormente). Um dos multiplexadores seleciona a entrada da ULA, que pode vir de um dos 4 vizinhos ou da saída do outro multiplexador, chamada saída do EP. O outro multiplexador seleciona, portanto, a saída do EP, que pode vir da saída da ULA, de um barramento comum ou pode ser um dado lido da memória. A saída do EP pode ser armazenada na memória ou pode ser transferida para o controle central, através dos barramentos comuns.

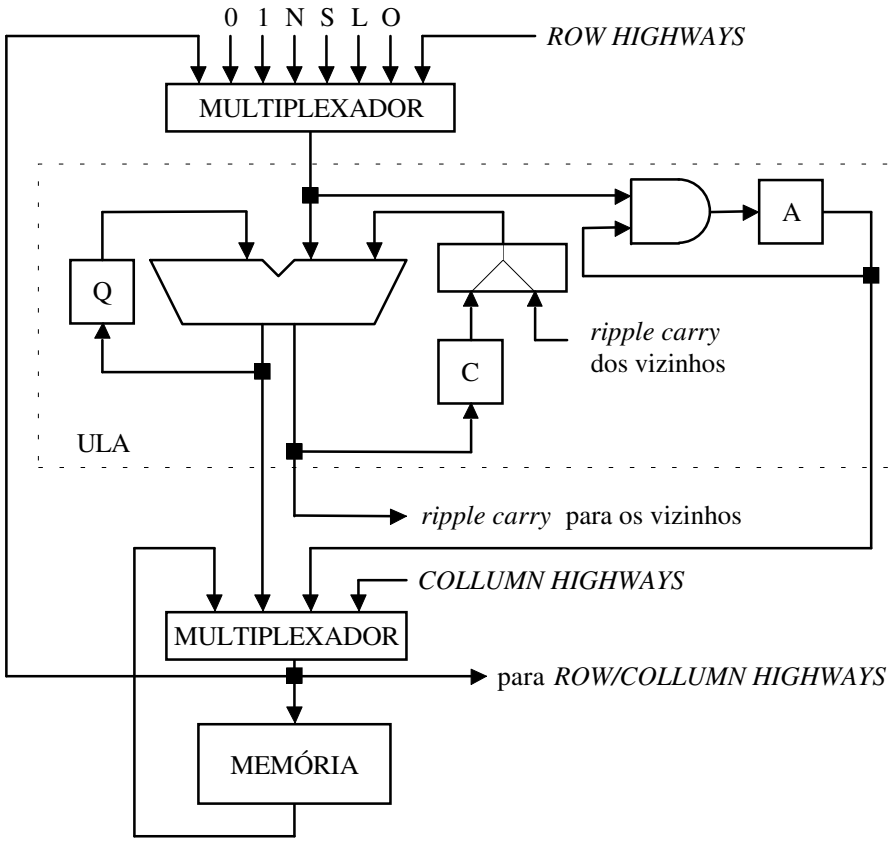

Figura II.6 - Diagrama funcional simplificado do elemento processador do DAP

A ULA possui três registradores de 1 *bit* (A, Q e C) e um somador completo. O registrador A funciona como o registrador de modo do SOLOMON. Ele permite a inibição da escrita na memória. Para maior facilidade, o registrador A possui uma porta lógica E na sua entrada, permitindo a combinação do seu valor atual com a entrada da ULA. O registrador Q age como um acumulador e o registrador C com um *carry*. A saída da ULA vêm do registrador Q. Existe ainda uma saída vinda do registrador C que permite o envio de dados para os vizinhos. Esta segunda saída permite a execução de operações aritméticas paralelas sobre operandos com largura de n *bits*, usando n processadores e transferindo os *carries* para os processadores vizinhos.

II.6 - *Connection Machine*

A *Connection Machine* [Tuck88] foi desenvolvida a partir de 1980 no MIT (Massachusetts Institute of Technology), num projeto em que o principal pesquisador foi W. D. Hillis. O sucesso do projeto desencadeou a fundação da Thinking Machines, em 1983. Através dessa nova empresa, a *Connection Machine* foi aprimorada e produzida, tornando-se uma máquina comercial de sucesso. O primeiro protótipo, denominado CM-1 foi desenvolvido entre os anos de 1983 e 1984. A primeira máquina completa foi construída em 1985 e a comercialização do CM-1 começou em 1986. A segunda versão da *Connection Machine* - denominada CM-2 - começou a ser comercializada em 1987.

O objetivo principal do projeto foi o de construir uma máquina paralela capaz de melhor resolver problemas associados a Inteligência Artificial. A associação entre os processadores na forma de uma matriz (com cada processador se comunicando com os seus vizinhos mais próximos), encontrada até então na maioria das arquiteturas SIMD, é adequada para resolver problemas em que as estruturas de dados são naturalmente organizadas na forma de vetores. Para outros tipos de problemas, no entanto, essa estrutura não é eficiente. A maior diferença entre a *Connection Machine* e as arquiteturas apresentadas anteriormente se refere, como o próprio nome já diz, às conexões entre os processadores. As conexões são programáveis e bastante flexíveis, adaptando-se melhor a diferentes estruturas de dados.

A *Connection Machine* é composta por até 64k processadores, divididos em até 4 blocos de 16k processadores. Cada bloco possui um seqüenciador independente.

A *Connection Machine* possui cinco diferentes mecanismos de comunicação: transmissão de dados comuns para todos os processadores (como os barramentos comuns vistos em arquiteturas anteriores), OU-inclusivo das saídas de *carry* das ULAs de todos os processadores (como o somador lógico proposto pelo SOLOMON), hipercubo, roteador e grade *NEWS* (NLOS - Norte, Leste, Oeste, Sul). O hipercubo é uma rede física com topologia de um n-cubo. No caso de uma configuração completa, o hipercubo é um cubo de 12

dimensões ligando 4096 grupos de 16 processadores. O roteador permite o envio de pacotes de mensagens entre quaisquer processadores. Cada pacote contém o endereço do processador de destino e os dados a serem enviados. Através de uma interface de roteamento e usando a rede do hipercubo é possível armazenar e combinar mensagens de mesmo destino. A grade NEWS, que implementava a tradicional comunicação com os quatro vizinhos imediatos na CM-1, foi um pouco modificada na CM-2. Nesta versão da *Connection Machine*, a grade bidimensional foi substituída por uma grade n-dimensional, capaz de suportar desde comunicações unidimensionais até 16-dimensionais. Este mecanismo de comunicação é implementado através da interface de grade n-dimensional e da rede do hipercubo.

Cada grupo de 16 EPs (com exceção das memórias associadas a eles) é encapsulado em um único *chip*. Cada placa da CM-1 contém 512 EPs e memórias associadas. Logo, 128 placas são necessárias para compor a configuração completa da *Connection Machine*.

Cada EP é composto por: uma memória de acesso randômico (4 *kbits* na CM-1 e 16 *kbits* na CM-2), uma ULA com registradores associados, registradores de 1 *bit* para *flags* (8 registradores na CM-1 e 4 registradores na CM-2), uma interface de roteamento e uma interface de grade (bidimensional na CM-1 e n-dimensional na CM-2). Além disso, a CM-2 possui uma interface de entrada e saída, um circuito de detecção de erro e um acelerador de ponto flutuante opcional.

A ULA presente em cada EP (Figura II.7) é capaz de executar quaisquer duas funções envolvendo três entradas. Duas entradas da ULA são fornecidas pela memória e a terceira vem de um dos registradores de *flag*. Um dos resultados gerados pela ULA é armazenado na memória e o outro é armazenado em um dos registradores de *flag*. Um registrador de *flag* pode ser usado como condicionante do armazenamento dos resultados da operação.

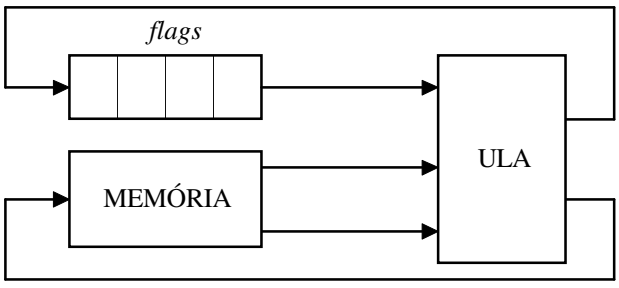

Figura II.7 - Diagrama funcional simplificado do elemento processador da CM-2

II.7 - BLITZEN

O BLITZEN [Blev] foi desenvolvido por pesquisadores americanos no final da década de 80. Ele é um aprimoramento do MPP, tendo a estrutura dos seus EPs baseada neste sistema.

A evolução da tecnologia de integração tornou possível o encapsulamento de 128 EPs (organizados numa matriz com 8 linhas e 16 colunas) em um único *chip*. Além disso, a memória associada a cada EP (1024 *bits*/EP) também pode ser incorporada ao *chip*. O número total de EPs foi mantido: 16.384 EPs organizados na forma de uma matriz com 128 linhas e 128 colunas.

Os EPs continuam sendo processadores de 1 *bit* (Figura II.8). Cada EP possui 6 registradores de 1 *bit* (A, B, C, G, K e P). O registrador S, utilizado no MPP para a entrada e a saída de dados na matriz de processadores foi eliminado. Por outro lado, o registrador K foi acrescentado à arquitetura.

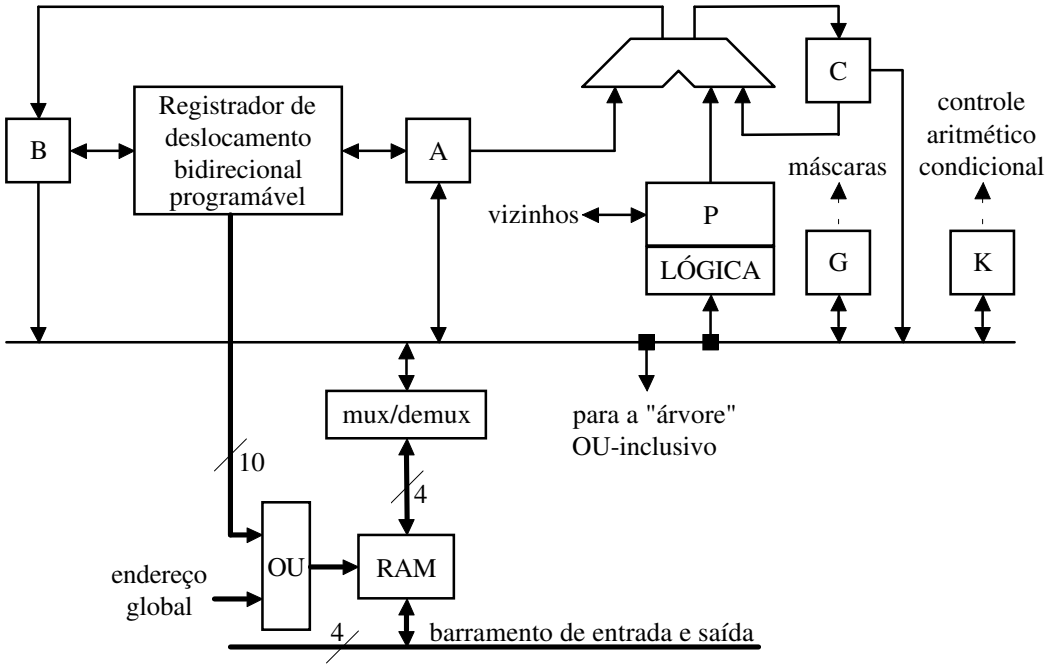

Figura II.8 - Diagrama funcional do elemento processador do BLITZEN

O registrador K é um aprimoramento no controle local, restrito ao registrador de máscara G no MPP. Uma operação "mascarada" permite que alguns EPs executem uma tarefa (se G=1) enquanto outros EPs não executam esta tarefa (se G=0), de acordo com o valor da máscara G. Conceitualmente, uma operação "mascarada" suporta uma instrução de alto nível do tipo **SE** condição, **ENTÃO** execute tarefa1. As seguintes operações podem ser "mascaradas": atribuição de valor ao registrador P através de operação lógica, inclusão do valor de P na soma, atribuição de valores aos registradores A, B e C, deslocamento de 1 *bit* (para a esquerda ou para a direita) do registrador de deslocamento e escrita na memória.

O registrador K veio implementar o controle condicional, que permite que alguns EPs executem uma tarefa enquanto outros EPs executam outra tarefa. Nas operações com controle condicional local, se K=0, o resultado é normal. Porém, se K=1, a operação produz como

resultado o complemento do resultado normal. Conceitualmente ela suporta uma operação de alto nível do tipo **SE** condição, **ENTÃO** execute tarefa 1, **SENÃO** execute tarefa2; sendo que ambas as tarefas são executadas simultaneamente. As seguintes operações suportam controle condicional local: atribuição de valor inicial ao registrador C e atribuição de valor ao registrador P através de operação lógica. O objetivo do BLITZEN ao criar operações condicionais, foi o de permitir que alguns EPs executem uma soma, ao mesmo tempo em que outros EPs estão executando uma subtração, o que é muito útil na implementação de um algoritmo de divisão sem restauração (*non-restoring*), como será visto no Capítulo III.

Outra modificação em relação ao MPP foi a exclusão da lógica P=G ($P \oplus G$), disponível no barramento de dados.

O registrador de deslocamento de tamanho variável - existente no MPP - também foi aprimorado, passando a ser bidirecional (incluindo os registradores A e B). Além disso, foram acrescentadas instruções para a atribuição do valor 0 (zero) tanto para o registrador de deslocamento (apenas para os estágios ativos), quanto para os registradores A e B.

Problemas inerentes ao projeto em VLSI das memórias de 1 K*bits* associadas a cada EP levaram a opção pela implementação dessas memórias com largura de 4 *bits* (256X4 *bits*). Apesar disso, cada memória de 256X4 *bits* é vista pelo EP ao qual ela está associada como se ela fosse uma memória de 1024X1 *bits*. Isto é feito através do uso de um multiplexador/demultiplexador.

Em relação a entrada e a saída de dados na matriz de processadores, a memória de cada EP é vista como uma RAM de 256X4 *bits*. Para cada linha da matriz, cada conjunto de 16 processadores (isto é, 16 colunas da linha) possui um barramento próprio de 4 bits para entrada/saída de dados. Um endereço de 4 bits seleciona qual das 16 colunas será utilizada para a leitura/escrita de dados. Existe ainda uma operação de entrada (*broadcast input*) que grava o conteúdo do barramento de 4 *bits* nas memórias de todas as 16 colunas de EPs.

Ainda em relação à memória, uma importante inovação do BLITZEN foi a criação de um mecanismo simples de modificação local do endereço global de acesso à memória. O ILLIAC IV já permitia uma modificação local do endereço, mas através de um registrador e um somador, ambos de 16 *bits*. O custo de um somador da mesma largura do endereço (10 *bits* no caso do BLITZEN) é muito alto, principalmente se for levado em consideração que cada EP é um processador de apenas 1 *bit*. A solução adotada pelo BLITZEN foi a substituição do somador por um OU entre o endereço global e os 10 *bits* mais significativos do estágio de 16 *bits* do registrador de deslocamento de cada EP. Embora esta estratégia seja mais restritiva do que a adotada pelo ILLIAC IV, ela pode ser bastante útil. O OU pode simular uma soma,

desde que a base do endereço seja escolhida de forma que os *bits* menos significativos sejam iguais a zero.

A maior inovação do BLITZEN encontra-se na rede de roteamento entre os processadores vizinhos. Essa rede prevê uma comunicação de 1 *bit* de largura entre cada EP e seus oito EPs vizinhos: norte (N), nordeste (NE), este (E), sudeste (SE), sul (S), sudoeste (SO), oeste (O) e noroeste (NO). Porém, não são usadas oito linhas de conexão bidirecionais para cada EP, como era de se esperar, mas apenas quatro. A cada EP estão associadas quatro linhas direcionadas aos vizinhos: NE, SE, SO e NO. Na interseção entre duas linhas que se cruzam é posta uma conexão ligando as duas linhas, como mostra a Figura II.9. Numa operação de roteamento, cada EP envia dados em uma direção (NE, SE, SO ou NO) e recebe dados de uma das outras direções restantes (NE, SE, SO ou NO). Estas duas direções (de envio e recebimento) definem a direção do fluxo de dados (N, NE, E, SE, S, SO, O ou NO). Por exemplo, para rotear dados na direção norte, basta que cada EP envie dados para a linha NE e receba dados da linha SE.

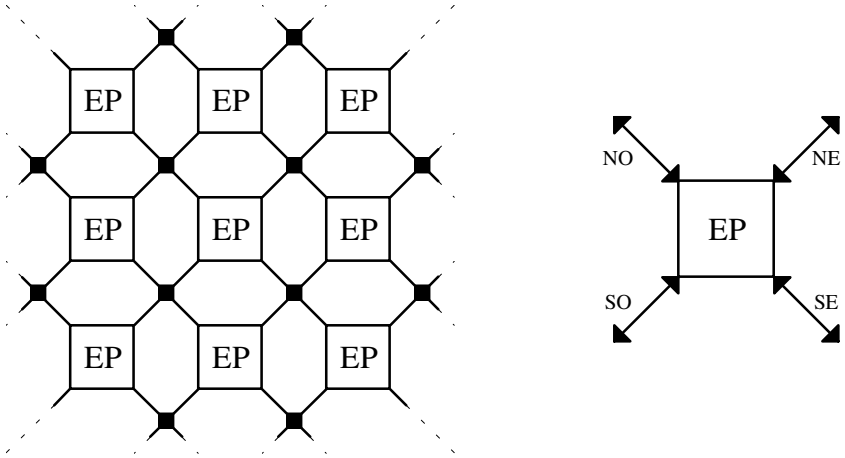

Figura II.9 - Configuração da rede de interconexões entre EPs no BLITZEN

II.8 - MasPar

O MasPar MP-1 [Blan90, Nick90] é uma máquina SIMD desenvolvida pela MasPar Computer Corporation. O seu projeto teve por objetivo construir um sistema com um desempenho da ordem de GigaFLOPS ao custo de um minicomputador. A comercialização do MP-1 teve início em Janeiro de 1990 e em Outubro de 1992 foi lançado o MP-2, um aprimoramento do MP-1.

A Unidade de Controle Vetorial (ACU) do MP-1 é um processador escalar RISC de 14 MIPS, com um ciclo de relógio de 70ns. A ACU é responsável pelo controle do vetor de EPs e pela execução de instruções escalares. A ACU ocupa uma placa de circuito impresso.

O vetor de EPs pode ser configurado com 1 a 16 placas de processadores. Cada placa possui 1024 EPs e memórias associadas. Sendo assim, o MP-1 pode ter de 1024 a 16384 EPs. Cada placa tem dimensão aproximada de 14 polegadas por 19 polegadas e consumo menor do que 50 watts. Os EPs são organizados em cada placa através de *clusters*. Cada placa possui uma matriz com 8 linhas e 8 colunas de *clusters*, totalizando 64 *clusters* de 16 EPs cada um. Para a integração dos processadores no MP-1 foi utilizada tecnologia CMOS VLSI, o que permitiu o encapsulamento de 2 *clusters* por *chip*. Dentro de cada *cluster*, os EPs são organizados na forma de uma matriz com 4 linhas e 4 colunas. A cada EP está associada uma memória (PMEM) de 16 kbytes.

Ao contrário da maior parte dos sistemas apresentados anteriormente, os EPs do MasPar não são processadores de 1 *bit*, mas sim processadores de 4 *bits*, o que aumenta o desempenho individual de cada EP. Seguindo a filosofia dos processadores RISC atuais, cada EP tem um grande conjunto de registradores internos (PREG), onde são executadas as operações. A memória externa PMEM só é utilizada para carregar e armazenar dados no conjunto de registradores PREG. Cada EP possui 40 registradores de 32 *bits* disponíveis para o programador e 8 registradores de 32 *bits* de uso interno. Os registradores ocupam 30% da área do *chip* e correspondem a 75% do total de transístores do *chip*.

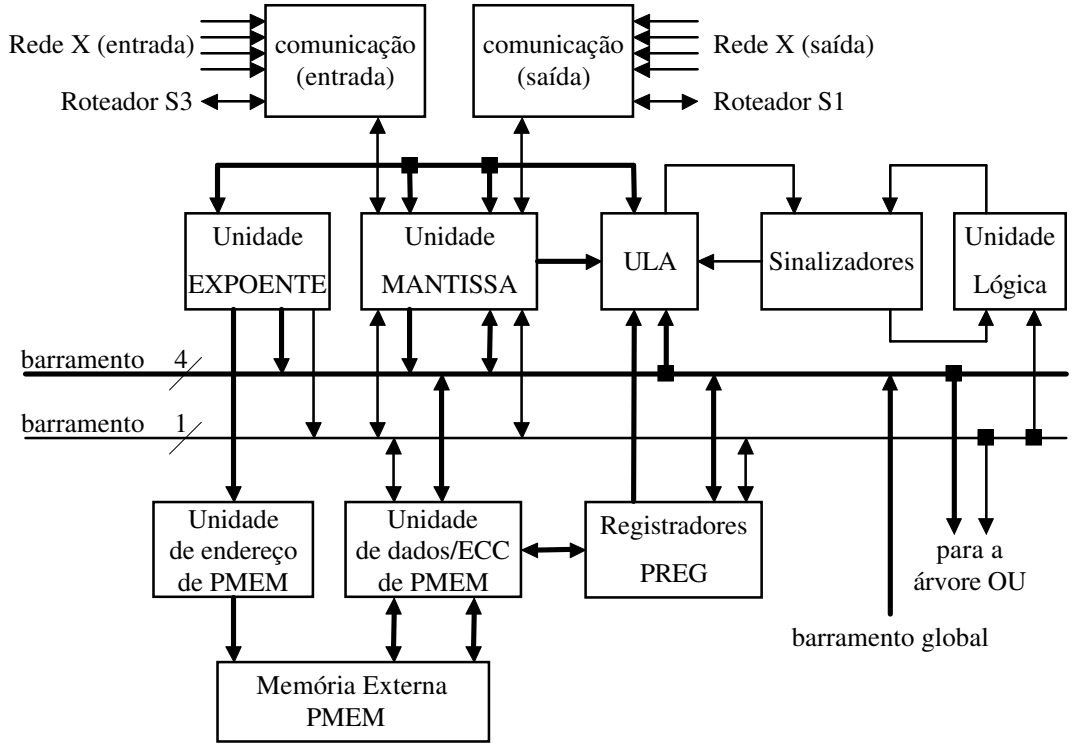

Figura II.10 - Diagrama funcional do elemento processador do MasPar MP-1

A ACU provê operações sobre booleanos (1 *bit*), inteiros (1, 8, 16, 32 e 64 *bits*) e reais representados em ponto flutuante (32 e 64 *bits*). As operações sobre reais são oferecidas para

dois padrões: o IEEE (formatos *single* e *double*) e o VAX (formatos D, F e G). Para prover essas operações, os registradores podem ser endereçados por *bit*, *byte* ou palavras de 16, 32 ou 64 *bits*. A estrutura de um EP do MP-1 é mostrada na Figura II.10. Cada EP possui um barramento principal de 1 *bit*, um barramento principal de 4 *bits*, uma unidade de 64 *bits* denominada MANTISSA, uma unidade de 16 *bits* denominada EXPOENTE, uma unidade de sinalizadores (*FLAGS*), uma unidade lógica e aritmética de 4 *bits* e uma unidade lógica de 1 *bit*. Entre os sinalizadores, existe um *bit* denominado E capaz de inibir algumas operações enviadas pela ACU. Ele funciona da mesma forma que o registrador G do MPP.

Em relação à comunicação entre a ACU e o vetor de EPs, não existem grandes inovações. Há um barramento comum, que permite à ACU enviar instruções e dados (4 *bits*) para todos os EPs, ou apenas para EPs selecionados. Há também uma árvore OU que transmite para a ACU o estado dos barramentos globais de 1 e 4 *bits* de todos os EPs.

O MP-1 executa simultaneamente operações envolvendo PMEM e cálculos nos EPs. Isso é possível porque PMEM só é utilizada para trocar dados com PREG. A ACU previne conflitos no uso dos registradores PREG, impedindo leituras ou escritas prematuras. O endereçamento de PMEM pode ser direto ou indireto. No endereçamento direto é utilizado o endereço global enviado pela ACU. Já o endereçamento indireto utiliza o endereço armazenado na unidade de endereço de PMEM, existente em cada EP. Esse endereço já deve ter sido previamente calculado e armazenado nessa unidade. Além das facilidades oferecidas por um endereçamento local, o endereçamento indireto ainda é um terço mais rápido do que o endereçamento direto.

Em relação às comunicações de dados entre EPs, o MasPar possui dois mecanismos. O primeiro é a rede de comunicação X, que provê um canal de comunicação de 1 *bit* entre cada EP e seus oito vizinhos mais próximos, seguindo a mesma filosofia do BLITZEN. O segundo mecanismo é a rede de roteamento global, implementada através de uma rede *crossbar* hierárquica de três estágios, que permite uma comunicação bidirecional ponto a ponto entre qualquer par de processadores de uma mesma placa. A rede de roteamento é dividida em três estágios: S1, S2 e S3. Para estabelecer uma comunicação, o EP de origem calcula o número do EP alvo e envia esse número para o roteador S1. Cada estágio do roteador seleciona uma conexão com o próximo estágio baseado no número do EP alvo. Uma vez estabelecida, a conexão é bidirecional, permitindo a transmissão de dados entre o EP de origem e o EP alvo. Quando a conexão é fechada, o EP alvo retorna uma aceitação de fim de conexão (*acknowledgment*).