

VETOR
DESCRIÇÃO DO PROJETO DE HARDWARE
DE UM SISTEMA MULTIMICROPROCESSADOR
PARA PROCESSAMENTO NUMÉRICO
por
FERNANDO ROSA DO NASCIMENTO
RP n. 115 AGOSTO/89

Trabalho realizado com o apoio do CNPq.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
Av. Osvaldo Aranha, 99
90.210 - Porto Alegre - RS, BRASIL

Endereço para correspondência:

UFRGS/CPGCC/Biblioteca

Caixa Postal 1501

90.001 - Porto Alegre - RS, BRASIL



RESUMO

Esta publicação apresenta o projeto de hardware do VETOR, um sistema multimicroprocessador para processamento numérico. O sistema faz uso de uma UCP de 32 bits (MC68020), de quatro processadores numéricos (MC68881) e de memória local compartilhada. Foi projetado para ser utilizado como processador auxiliar de microcomputadores com arquitetura IBM-PC. O trabalho faz parte do Projeto de Processamento Numérico de Alto Desempenho (PNAD).

PALAVRAS-CHAVES: multimicroprocessamento, processamento numérico, aplicações de microprocessadores.

ABSTRACT

This paper describes the hardware project of the VETOR, a multimicroprocessor system to work with numerical problems. This system is built with one 32 bits UCP (MC68020), four numeric data processor (MC68881) and one shared local memory. The VETOR is intended to be used with a microcomputer IBM-PC. This work is part of the Project of High Performance Numeric Processing (PNAD).

KEY-WORDS: multimicroprocessing, numeric processing, microprocessor applications.

SUMÁRIO

1. INTRODUÇÃO	01
2. DESCRIÇÃO GERAL	02
2.1 Unidade Central de Processamento	06
2.2 Coprocessadores Numéricos	09
2.3 Memória Principal	13
2.3.1 Banco de memória ROM	13
2.3.2 Memória Multiporta	15
2.3.3 Mapas de memória	21
2.4 Sistema de interrupção bidirecional	22
3. OBSERVAÇÕES E CONCLUSÕES	24
 BIBLIOGRAFIA	 29
 ANEXO 1 : Diagrama esquemático	 30
ANEXO 2 : Lista de componentes	34
ANEXO 3 : Lista de êrros elétricos	37

1. INTRODUÇÃO

A motivação para o projeto do sistema multimicroprocessador VETOR se deve a constatação de que existe uma necessidade de processamento numérico de alta velocidade, que atualmente não está disponível nos microprocessadores e minicomputadores do mercado nacional. As aplicações para tais computadores envolvem engenharia, matemática, física, meteorologia, controle de processos e robótica entre outros.

A solução encontrada para o projeto do VETOR foi do ponto de vista de Arquitetura de Computadores, um sistema multimicroprocessador com memória compartilhada, implementado numa "placa aceleradora". Esta escolha foi feita porque não se queria nesta fase do PNAD partir para a implementação completa de um computador; por motivos financeiros e técnicos e por não ser uma solução para uso geral.

O emprego do conceito de "placa aceleradora" foi facilitada visto que o computador hospedeiro escolhido, os IBM-PCs, apresentam uma arquitetura aberta e conhecida. Tais computadores dispõem de vários conectores num barramento comum de E/S, que podem ser utilizados pelos mais diversos dispositivos.

O VETOR emprega uma UCP de alto desempenho (MC68020), quatro Processadores Numéricos (NDPs) do tipo MC68881, 32 KB de RAM e de 64 KB de ROM.

A interface entre o VETOR e o computador (PCs XT,AT OU AT-386) se faz via um barramento de dados de 8 bits . A comunicação entre os dois se dá via uma memória compartilhada que pode ser acessada pelo computador hospedeiro. A sincronização

entre os processos a serem executados pelo VETOR e pelo computador hospedeiro é feita com auxílio de uma estrutura de interrupção bidirecional.

O relógio do MC68020 é de 12 Mhz, mas a troca de alguns componentes (memórias e UCPs) e a geração de ciclos de espera para o PC (se necessário) podem elevar a frequência de operação para 25 Mhz, duplicando pelo menos o desempenho do VETOR.

2. DESCRIÇÃO GERAL

A descrição do projeto do VETOR procura ser o mais abrangente possível, mas enfoca principalmente os aspectos de Arquitetura e Organização de Computadores e os de aplicações de microprocessadores. Não serão no entanto comentados os requisitos determinados para o sistema, sendo apenas citados os mais importantes, que são :

- processamento numérico de alto desempenho via técnicas de paralelismo;
- barramentos internos das UCPs e da RAM com 32 bits, da ROM com no mínimo 16 bits e do interface com o microcomputador hospedeiro (qualquer IBM-PC) com no mínimo 8 bits;
- processadores numéricos (NDPs) com funções trigonométricas, logarítmicas e exponenciais; operandos com precisão interna de 80 bits e funcionamento tipo coprocessador;
- UCP de alto desempenho para controlar a execução dos NDPs e comunicação com o microcomputador hospedeiro;
- memória multiporta que permita o acesso do microcomputador hospedeiro a qualquer instante;
- sistema de interrupção bilateral entre o VETOR e o microcomputador hospedeiro.

O projeto empregou os componentes citados na introdução. A família de microprocessadores escolhida foi a da MOTOROLA. Devido as necessidades dos tipos de instruções, precisão dos operandos e forma de operação; as alternativas viáveis eram os coprocessadores da INTEL e da MOTOROLA.

Tecnicamente os coprocessadores da segunda são mais velozes e têm melhores instruções, mas os da primeira (386+387) implementam de uma forma mais direta e simples um sistema. Quanto a fatores econômicos, os CIs da MOTOROLA tem se mostrado historicamente bem mais acessíveis. O sistema formado pode ser visto no diagrama em blocos da figura 2.1 .

É mostrado nesta figura os principais elementos, a ligação entre eles e a largura do barramento de dados usada em cada conexão. O VETOR emprega internamente um barramento comum (multiplexado no tempo) para interconectar todos os elementos. A

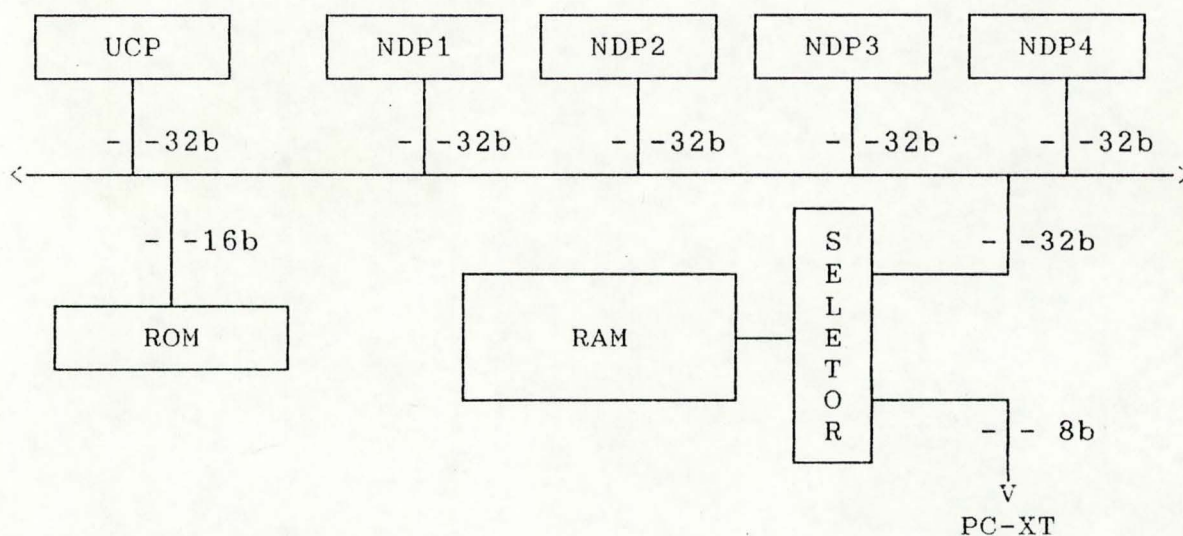


Figura 2.1 - Diagrama em blocos do VETOR

UCP em conjunto com os NDPs formam os elementos processadores do sistema, que a princípio executam os programas armazenados nas ROMs e eventualmente nas RAMs. Os dados a serem processados estão na RAM e foram ali colocados pelo computador hospedeiro via o interface, com barramento de 8 bits. Após serem processados serão

retirados pelo computador hospedeiro via o mesmo interface. O diagrama em blocos da figura 2.1 dá uma idéia da forma de interconexão empregada para acesso a memória compartilhada, mostrando que foi utilizado uma memória multiporta (MMP).

Os NDPs e a UCP estão ligados diretamente ao barramento comum. O paralelismo a ser obtido se deve ao fato de que os NDPs ficam processando internamente suas instruções, liberando o barramento comum (multiplexado no tempo) para os outros NDPs, a UCP e mesmo o computador hospedeiro. O controle dos NDPs é feito pela UCP, que tem ainda a tarefa de sinalizar qualquer evento necessário ao computador hospedeiro ou processar qualquer interrupção recebida.

Os barramentos de dados externo e interno da UCP são de 32 bits, o barramento de dados externo dos NDPs tem 32 bits e o interno 80 bits. Os barramentos da ROM, da RAM e da interface com o computador hospedeiro são respectivamente de 16, 32 e 8 bits de largura.

O uso do barramento de RAM com 32 bits permite explorar ao máximo a banda de passagem do barramento de dados da UCP e dos NDPs, fazendo por exemplo, com que a leitura/escrita de operandos com 8 bytes sejam realizados em apenas dois ciclos de acesso a memória, caso o deslocamento seja zero ($A_0=A_1=0$).

Já o banco de memória ROM tem um barramento de dados de 16 bits por questões de espaço físico (número de CIs na placa) e simplicidade, não se esperando a princípio que esta escolha afete muito o desempenho do sistema. Para tal previsão são duas as considerações: a UCP tem uma pequena cache de instruções (256

bytes) e se for necessário os programas da ROM poderiam ser transferidos para a RAM, que é muito mais rápida.

A largura do interface entre o VETOR e o computador hospedeiro é de 8 bits, no entanto pode ser facilmente alterado para barramento de 16 ou 32 bits, pois já existem transceptores de dados para tal fim, necessitando-se apenas a previsão de conectores na hora de ser projetada a placa de circuito impresso para o mesmo.

A fim de se evitar soluções de sincronização de processos do tipo "busy wait", foi implementado um sistema de interrupção bilateral entre o VETOR e o computador hospedeiro.

2.1 Unidade Central de Processamento

O emprego da UCP MC68020 trouxe vantagens e desvantagens. A maior desvantagem é a de requerer uma lógica adicional considerável para acesso as memórias. A figura 2.1.1 mostra os sinais utilizados pela UCP no VETOR.

Como pode ser visto, são utilizados todas as linhas de dados, parte das de endereço e parte das de controle. Nesta versão do VETOR, a frequência do relógio é de 12 Mhz, visto as dificuldades ^{da} de obtenção de CIs de alta velocidade (UCP, NDPs, RAMs e lógica discreta). A UCP só está utilizando um nível de interrupção, a entrada IPL0 está ativa, e IPL1, IPL2 estão desativadas. Com isto só se está usando um dos sete possíveis níveis de interrupção, no caso o de menor prioridade e que é mascarável. Este nível é acionado caso o computador hospedeiro,

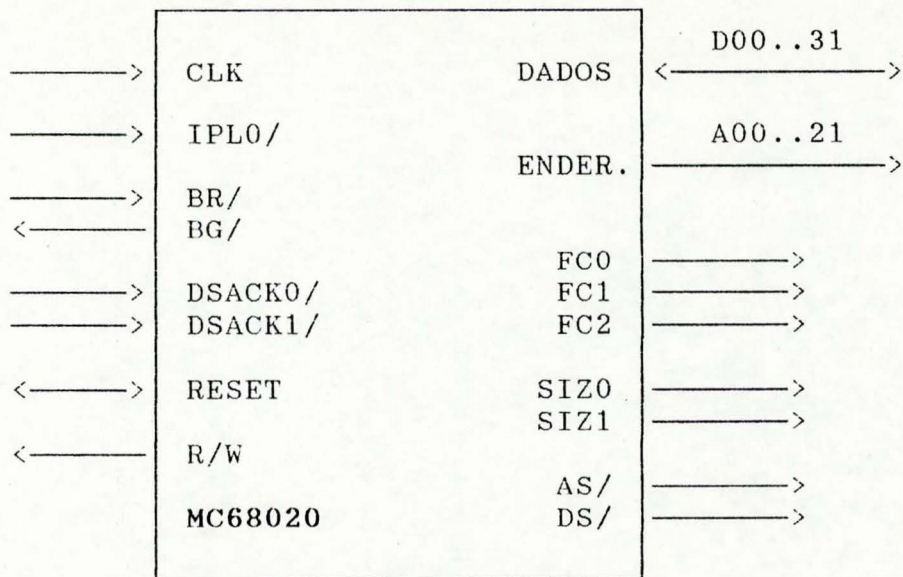
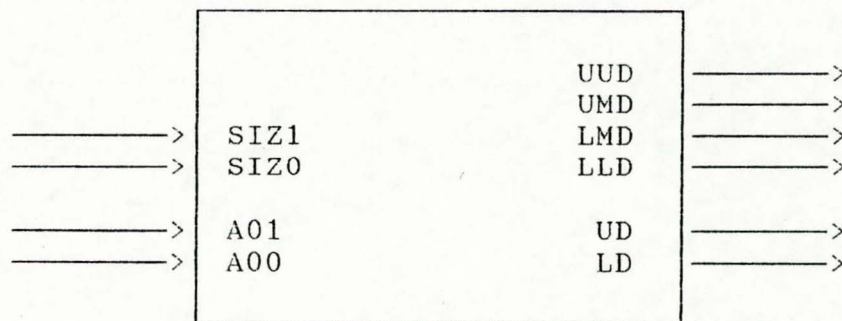


Figura 2.1.1 - Sinais da UCP utilizados pelo VETOR

um IBM-PC, enderece uma posição pré-determinada da sua própria memória. Os sinais de requisição e liberação de barramentos (BR,BG) são utilizados quando o computador hospedeiro deseja acessar a MMP.

Os sinais SIZ0,SIZ1 indicam se a UCP está querendo ler ou escrever um dado de 8/16/24/32 bits. Os sinais DSACK0,DSACK1 têm uma dupla função, sincronização com dispositivos externos e indicação da largura da porta que está sendo acessada (8/16/32 bits). A associação dos sinais SIZx, DSACKx e A00,A01 define de uma forma complicada e dispendiosa um protocolo que permite a UCP ler ou escrever dados de 8/16/24/32 bits em portas de 8/16/32 bits, dinamicamente, a cada ciclo de acesso (pgs. 5-2 à 5-15 em /MOT 84/). É necessário para tal, a implementação de uma lógica razoável e que consome bom tempo para se gerar os pedidos de acesso (CSs) dos integrados que formam o memória principal.

A figura 2.1.2 mostra o diagrama em blocos da geração dos sinais de seleção, por byte, para portas de 16/32 bits. Esta é uma lógica de três níveis, o que implica mesmo com o uso da família TTL, série 74S, num tempo de propagação típico de 10 ns, o que é significativo no tempo de acesso para o microprocessador usado. O uso de uma PAL na implementação de lógica da figura 2.1.2 reduz o número de CIs e possibilita a inclusão de mais sinais, e um exemplo bem detalhado desta técnica está no capítulo 12 em /MOT 87A/. A grande vantagem do microprocessador INTEL IAPX386 em relação aos MC68020/MC68030, é que ele já fornece diretamente os sinais de seleção por byte, não precisando da lógica mencionada acima, conforme pode ser visto em /INT 88/.



UUD = D24..31, porta de 32 bits
 UMD = D16..23, porta de 32 bits
 LMD = D08..15, porta de 32 bits
 LLD = D00..07, porta de 32 bits
 UD = D08..15, porta de 16 bits
 LD = D00..07, porta de 16 bits

Figura 2.1.2 - Bloco gerador de sinais de seleção para portas de 16/32 bits

O sinal AS indica que os sinais Axx, FCx, SIZx, e R/W

estão válidos, e é utilizado pelos NDPs e pelas RAMs. Os sinais DS (strobe de dado) e R/W (leitura/escrita) são utilizados pelos NDPs, RAMs e ROMs. Os sinais de código de função (FC0-FC2) são utilizados para identificar o espaço de endereçamento dos NDPs.

O barramento de dados é utilizado em toda sua extensão para acesso aos NDPs e as RAMs, e parcialmente aos demais dispositivos. A diversidade da largura das portas de dados das diversos unidades do sistema é possível graças a capacidade de reconfiguração dinâmica de barramentos que a UCP dispõe. Os sinais de dados não são bufferizados, reduzindo assim o número de CIs na placa. Nem todas as linhas de endereços são aproveitadas, sendo usadas apenas as de A0 à A21. As linhas de A16 à A21 vão para decodificadores e seletores. Houve um pequeno aproveitamento da capacidade efetiva de endereçamento porque se optou nesta versão inicial do VETOR, por uma memória pequena, implementada com RAMs estáticas. As linhas de endereço A0-A15 são bufferizadas por um conjunto de seletores usados na implementação da MMP, que será descrita adiante. Não é necessário bufferização

2.2 Coprocessadores Numéricos

Os NDPs MC68881 são os processadores principais do VETOR. O projeto inicial fixou um número de quatro NDPs, suficientes para por exemplo realizar confortavelmente operações vetoriais em múltiplos de quatro elementos, embora o MC68020 possa suportar até oito NDPS. A forma de ligação da UCP com os

NDPS é bastante simples, conforme é mostrado na figura 2.2.1 .

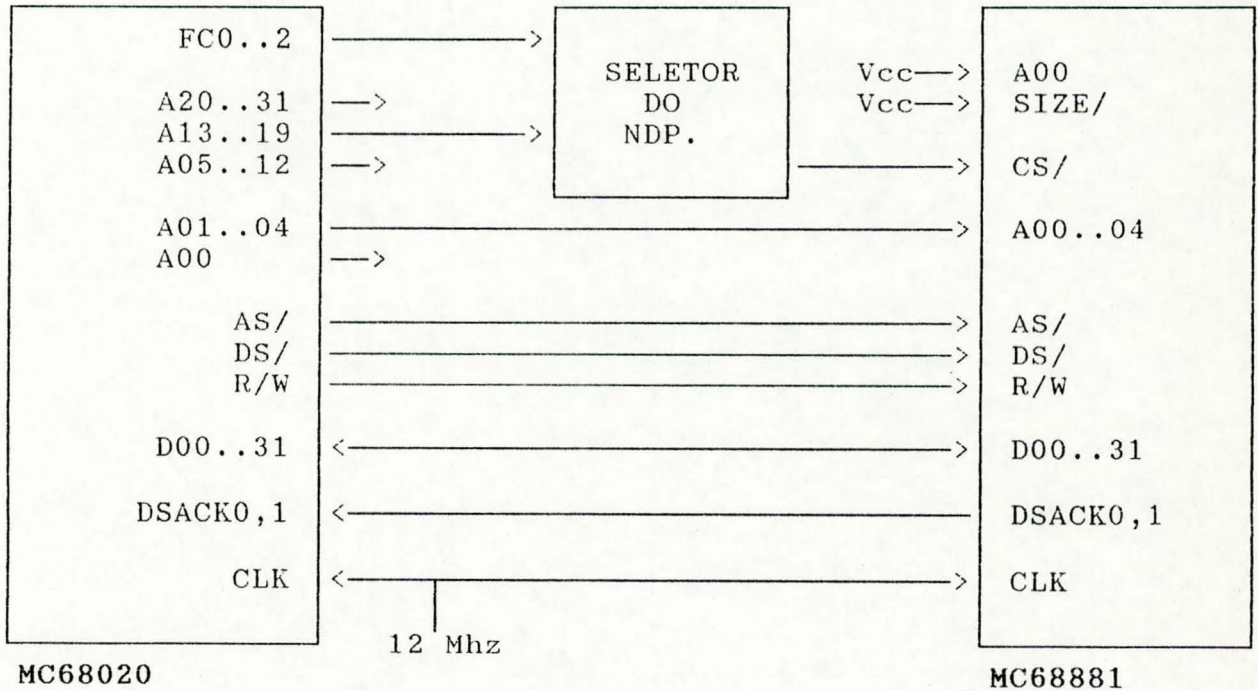


Figura 2.2.1 - Ligação CPU - NDP, por via de 32 bits.

O barramento de dados tem 32 bits, o que possibilita uma grande banda de passagem entre os processadores (UCP - NDPs), necessária porque os operando são de até 96 bits e têm-se que otimizar o máximo possível as operações de carga e descarga na pilha dos NDPs. Isto é importante também porque o barramento único pode ser utilizado por um dos seis possíveis processadores mestres: UCP, NDPs (4) e pela UCP do computador hospedeiro. Como a conexão é de 32 bits, a configuração do MC68881 é feita via os sinais SIZE e A00, que devem permanecer sempre em "1".

Os demais sinais de controle interconectados são: AS/, DS/, R/W, CLK e DSACK0,1. Existe ainda um sinal, SENSE/, que foi

utilizado para detectar a ausência de um determinado NDP quando o mesmo é endereçado. A figura 2.2.2 mostra que caso o NDP não esteja presente quando foi endereçado, o MC68020 receberá um sinal de erro de barramento (BERR/), que pode ser tratado pelo software. Isto possibilita a pesquisa e a implementação de algoritmos auto-reconfiguráveis em função do número de NDPs presentes.

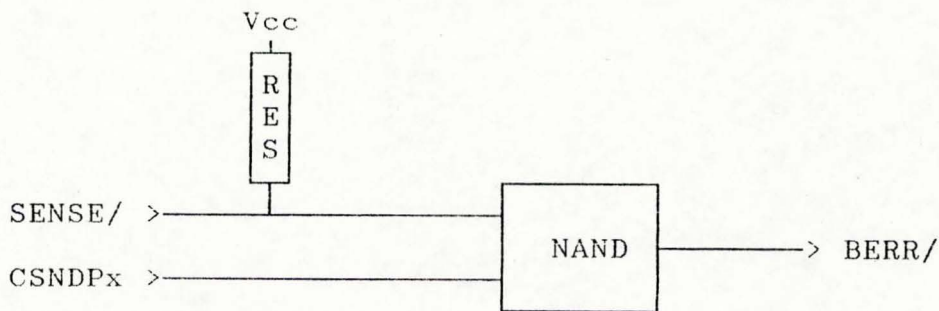


Figura 2.2.2 - Lógica de detecção do NDP.

A comunicação entre o MC68020 e os MC68881 se faz através de instruções usuais, não sendo portanto dependente da arquitetura (não depende de endereços de memória). Os NDPs não são dispositivos potencialmente mestres, como um canal de ADM, e nem geram interrupções. No entanto permitem operações em paralelo. Um NDP, nas fases de leitura ou escrita, só recebe ou entrega operandos, sendo funções da UCP nesta relação tipo coprocessador, a de calcular o endereço dos operandos e a de controlar as transferências. Pois é justamente durante esta fase de transferência de operandos para o NDP, que a UCP fica "presa", sendo após liberada para "disparar" instruções para os outros

NDPs; ou verificar o andamento ou término das instruções em outros NDPs /MOT 87B/.

As operações com os coprocessadores são feitas via instruções do tipo "F-line code", onde os quatro bits mais significativos estão em "1". A figura 2.2.3(a) mostra ainda que os bits 09,10 e 11 indicam um dos oito possíveis coprocessadores (Núm. Copr.). Os bits 6,7 e 8 codificam o tipo de instrução e os bits restantes complementam a instrução de acordo com o tipo. As instruções para acesso a coprocessadores, usam um dos oito espaços de endereçamento da UCP, é denominado de "CPU Space", e é indicado através das linhas de código de função (FC0..2) do MC68020 com valor "111".

O número do coprocessador presente na instrução é mapeado na hora nos bits 13,14 e 15 das linhas de endereço, quando a UCP acessa o NDP, conforme mostra a figura 2.2.3(b). São

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

1	1	1	1	Núm.Copr.	Tipo	Complemento
---	---	---	---	-----------	------	-------------

(a)

31 20 19 .. 16 15 14 13 12 05 04 00

0	0	...	0	0	0	0	0	0	0	1	0	Núm.Copr.	0	0	...	0	0	0	0	Sel. Reg.
---	---	-----	---	---	---	---	---	---	---	---	---	-----------	---	---	-----	---	---	---	---	-----------

(b)

Figura 2.2.3 - Formato da instrução (a) e formato do endereço (b) para um coprocessador.

as linhas de endereço que associadas ao código de função, servem para gerar o sinal de seleção do NDP. A figura 2.2.1 mostra o bloco lógico denominado de "Seletor do NDP", que realiza esta função. Para tanto é considerado ainda os bits de endereço A19..16 com valor "0010".

O MC68881 dispõe internamente de um conjunto de registradores mapeados no "CPU Space" via os bits de endereço A00..04 (Sel. Reg. na figura 2.2.3). No VETOR foi usado um decodificador de 2-4, que seleciona um dos quatro primeiros coprocessadores, dos oito possíveis no MC68020. O MC68881 usa o mesmo relógio da UCP, 12 Mhz.

2.3 Memória Principal

A memória principal do VETOR está dividida em dois bancos: um de REEPROM com 64 KB e um de RAM com 32 KB. Os bancos têm barramentos de dados com largura de 16 e 32 bits respectivamente. O banco de memória RAM é organizado sob a forma de uma memória multiporta (MMP), com duas portas, permitindo tanto o acesso dos processadores locais (UCP e NDPs) como do processador do computador hospedeiro, um computador IBM-PCXT.

2.3.1 Banco de memória ROM

O espaço de endereçamento de MC68020 vai até 4 gigabytes, mas a pesquisa inicial do VETOR prevê aplicações que envolvem pequena quantidade de memória. A proposta inicial é

dotar o VETOR com uma capacidade de ROM de 64 KB, suficiente para desenvolvimento de um ambiente de trabalho que permita a pesquisa de algoritmos básicos (operações vetoriais elementares), de um pequeno Sistema Operacional para viabilizar comunicações com o computador hospedeiro e de algoritmos auto-reconfiguráveis.

O ideal seria o uso de um banco de ROM com um barramento de dados de 32 bits, o que implicaria no uso de quatro CIs de 28 pinos, e que ocuparia um grande espaço na placa. Optou-se por um barramento de dados menor, de 16 bits, levando-se em conta que o MC68020 dispõe de uma cache de 256 bytes e de que uma possível redução no desempenho do VETOR ~~pode~~ este motivo, é perfeitamente controlável. Neste caso utilizou-se duas REPROMs de 256 Kbits.

As memórias do tipo REPROM têm a vantagem de serem não voláteis, mas devido a tecnologia usada para tal fim, não apresentam um bom tempo de acesso. No caso do VETOR com relógio de 12 Mhz, o uso de CIs modelo 27C256-135V05 permitem que a UCP realize ciclos de leitura sem precisar inserir ciclos de espera para sincronização. Mas no caso de versões mais velozes, de até 33 Mhz, será necessário a inclusão de uma lógica de sincronização, que inclua um ou até dois ciclos de espera. Por estes motivos, este banco de ROM, serve bem para programas de inicialização, programas residentes que se transferem para um banco de RAM (que não precise de ciclos de espera) ou como nesta versão inicial em que a redução de desempenho prevista não mascara os resultados.

2.3.2 Memória Multiporta

A MMP do VETOR tem as possíveis finalidades: de memória de rascunho, memória de dados, canal de comunicação entre a placa aceleradora e o computador hospedeiro e memória de programas. Os programas residentes podem ser programas em teste carregados sob o controle de um pequeno S.O. residente em ROM, ou programas já testados mas que precisam de uma memória com ótimo tempo de acesso.

A implementação do módulo de RAM teve que ser feita com CIs de memória estáticos porque se quis evitar a necessidade de multiplexadores de endereços e de lógica de "refresh" no caso do uso de CIs de memória dinâmica. Além destas razões, não ficou evidente que para os propósitos iniciais do VETOR, cálculos elementares sobre pequenos vetores e matrizes, seriam necessários uma área de dados e de comunicação grande. Certamente a análise das reais possibilidades do VETOR é que poderão determinar o tamanho da RAM em função de futuras aplicações.

A largura do barramento de dados do banco de memória RAM com a UCP do VETOR é de 32 bits, pelas razões já expostas. Os barramentos de dados são conectados diretamente a UCP, diminuindo assim o número de CIs e minimizando os tempos de acessos.

A figura 2.3.2.1 mostra num diagrama em blocos os detalhes da parte operacional da MMP do VETOR. Nela estão presentes os seguintes elementos: os CIs de memória estática de 8 Kbx8 (6164), o multiplexador de endereços e os transceptores de dados bidirecionais de 8 bits (74LS245).

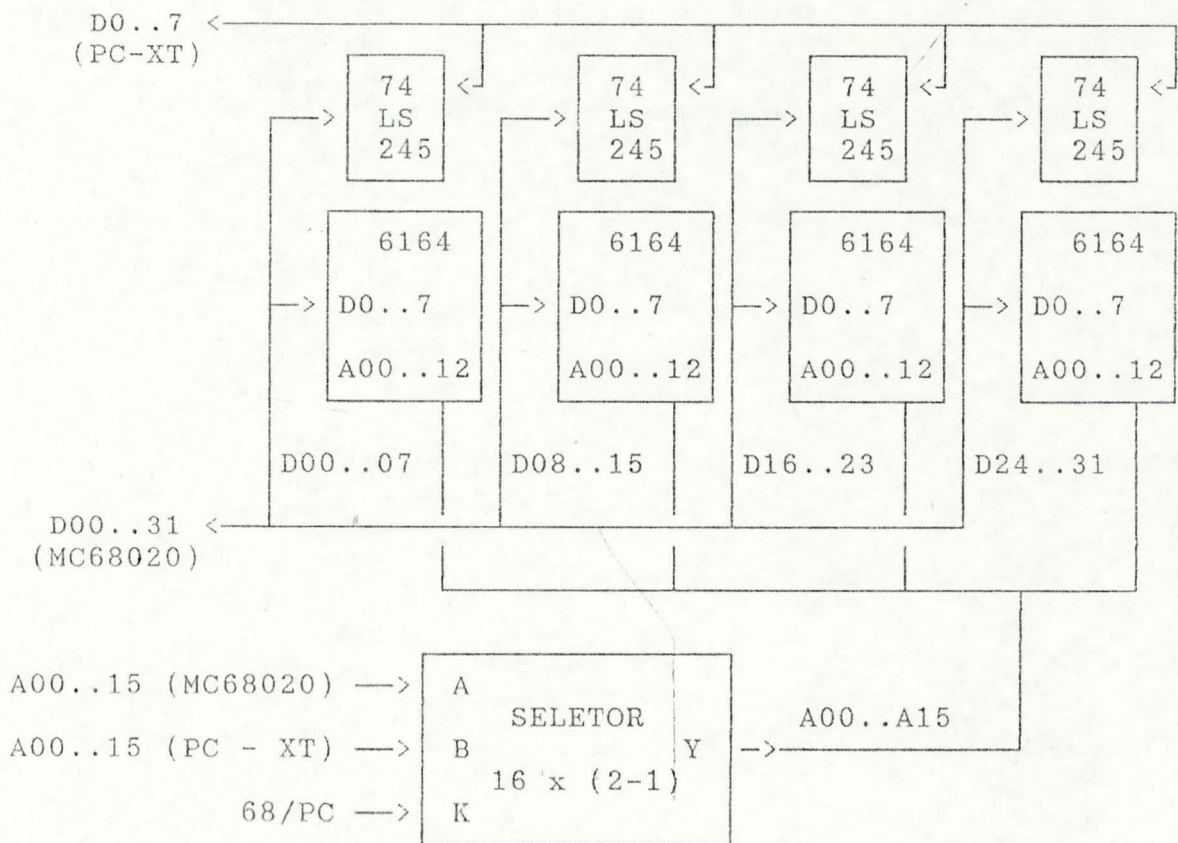


Figura 2.3.2.1 - Blocos operacionais da MMP do VETOR.

Esta implementação da MMP não foi feita da forma tradicional, onde em cada porta de acesso a memória estão presentes os barramentos de endereço e os de dados (cap. 4 em /NAS 81/). O fato de não haver uma porta de dados, isto é, um multiplexador bidirecional de dados, resultou numa economia de espaço, tempo e componentes. Isto foi possível graças a forma como foram utilizados os recursos do microprocessador MC68020, que será detalhado na descrição do bloco de controle da MMP.

Se a implementação da MMP fosse a tradicional, a vantagem resultante seria a de que a UCP do VETOR poderia acessar o banco de ROM, simultaneamente ao acesso do computador

hospedeiro ao banco de RAM. Esta implementação tradicional, neste caso não parece melhorar significativamente a relação custo/benefício pois: a UCP dispõe de cache de instruções e a frequência de acesso por parte do computador hospedeiro ao VETOR é limitada a carga e descarga de operandos (a sincronização de processos entre os dois computadores é feita via interrupção bidirecional).

Os CIs de memória são quatro, a fim de formarem uma porta de 32 bits de dados.

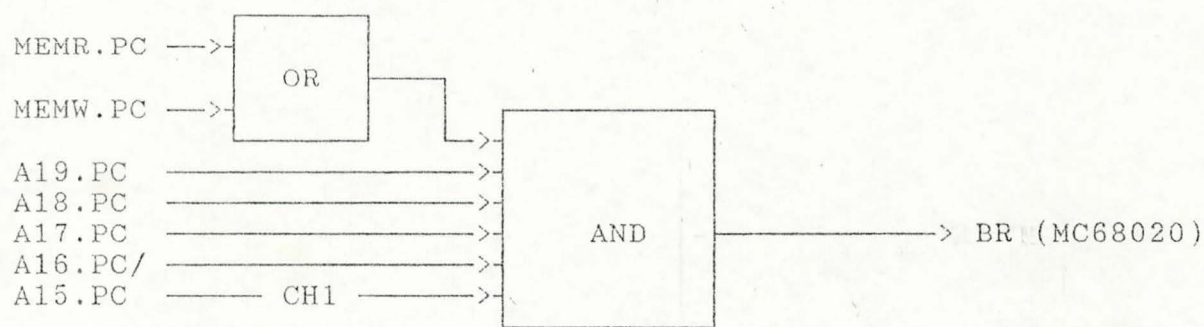
A descrição da MMP se confunde muito com a do interface do VETOR com o computador hospedeiro, pois a não ser alguns sinais de controle que são trocados entre ambos, a interface se faz realmente via uma das portas da memória compartilhada.

Na parte operacional, como é mostrado na figura 2.3.2.1, a interface de dados entre o computador hospedeiro e o VETOR se faz operacionalmente via um conjunto de quatro transceptores de dados bidirecionais de 8 bits. Existe um mapeamento dos endereços do computador hospedeiro ao acessar a RAM do VETOR, visto que a porta de dados do têm tamanhos de dados diferentes, 8 e 32 bits. Portanto quando o acesso a RAM é externo, somente um dos transceptores de dados é ativado, juntamente com seu respectivo CI de memória. Este mapeamento é obtido pela mesma lógica mostrada na figura 2.1.2, onde neste caso as entradas SIZ1, SIZ0 são forçadas para definirem uma porta de 8 bits (valor "01").

As linhas de endereço também devem ser multiplexadas. As opções possíveis com o uso da família 74 são: "buffers tri-

state" ou multiplexadores. Optou-se por CIs multiplexadores 74LS245, que proveêm uma ligação mais segura, com o mesmo número de CIs. As linhas multiplexadas são A00..15, embora só sejam necessárias para a RAM as linhas A00..12, isto porque os multiplexadores são de quatro bits, e a ROM usa até A14.

O controle da MMP é relativamente complexo, pois envolve dois computadores. Um bloco operacional desta MMP, a lógica de seleção de byte já foi descrita acima. O pedido de acesso de computador hospedeiro ao VETOR, se dá através de uma pequena região de memória. A lógica de requisição está mostrada na figura 2.3.2.2(a), onde ainda existe uma chave (CH1), que pode ser usada para dar uma pequena flexibilidade no bloco de endereçamento a ser endereçado pelo computador hospedeiro. A tabela 2.3.2.2(b) mostra as possíveis regiões que o computador



(a)

CH1	Faixa (hexa)	Tamanho
ON	[E8000,EF000]	32 KB
OFF	[E0000,EF000]	64 KB

(b)

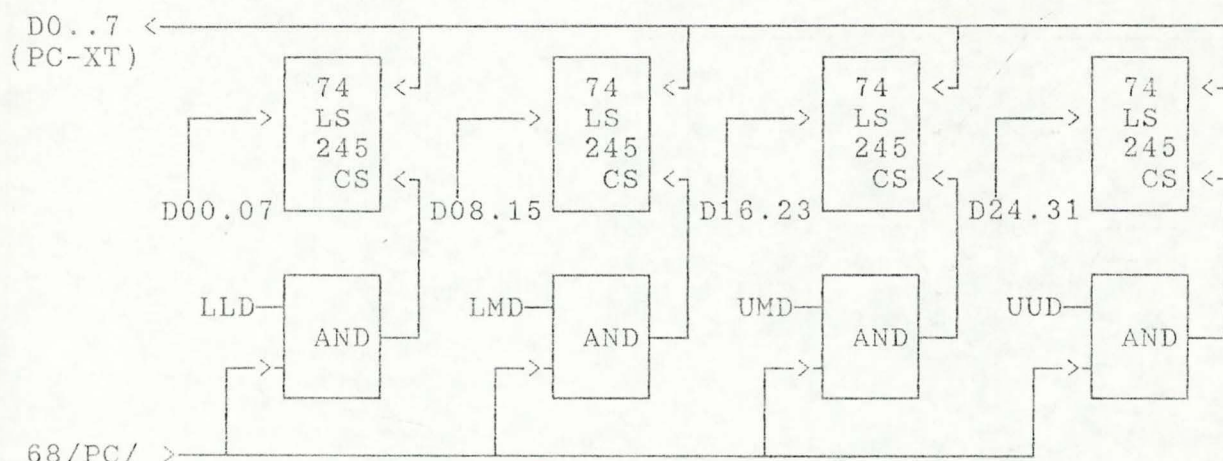
Figura 2.3.2.2 - Geração de pedido de acesso (a) e mapeamento de memória no VETOR (b).

hospedeiro pode utilizar para acessar a MMP do VETOR.

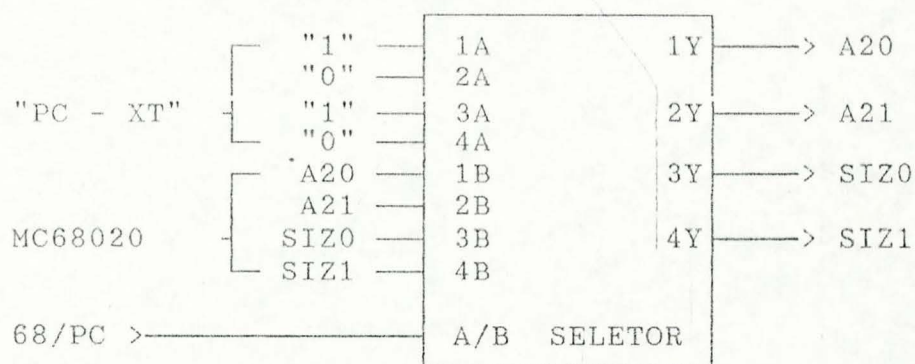
A lógica de requisição de barramentos é utilizada como arbitrador da MMP, para tanto o sinal gerado na lógica da figura 2.3.2.2(a) é encaminhado diretamente ao pino BR do MC68020. Quando a UCP responde com BG/ (68/PC), o seletor da figura 2.3.2.1 é comutado e o acesso a MMP efetivamente começa. O sinal 68/PC controla toda a comutação de porta da MMP, e é gerado a partir de BG, de acordo com a figura 2.3.2.3(c) .

O diagrama em blocos completo da porta adaptadora da interface com o computador hospedeiro está mostrada na figura 2.3.2.3(a), onde o transceptor de dados bidirecional só pode ser ativado após receber os sinais de seleção corretos, isto é, o seu seletor de byte (LLD,LMD,UMD ou UUD) com a liberação dos barramentos pela UCP (68/PC/). Quando o computador hospedeiro acessa a MMP, o mesmo espera até que ela esteja disponível, o que lhe é informado pelo sinal IOCHRDY, cuja geração é mostrada na figura 2.3.2.3(c). Este sinal faz parte do barramento do PC-XT, e serve para sincronizar a UCP 8088 com dispositivos mais lentos.

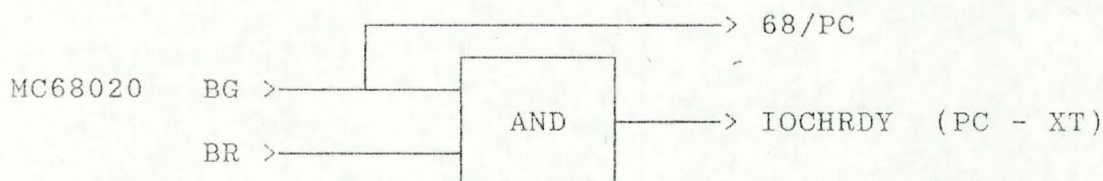
Os sinais A20 e A21 são utilizados por um seletor para endereçar a memória (ROM ou RAM) ou para pedir interrupção via um determinado endereço. Quando o MC68020 é o mestre no VETOR, ele mesmo gera estes bits de endereço, mas quando o mestre é o computador hospedeiro, há a necessidade de se simular tais bits. Isto é necessário pois um PC-XT não gera tais bits. A simulação é feita por um seletor, conforme mostra a figura 2.3.2.3(b), e que é selecionado pelo sinal 68/PC. Este seletor ainda multiplexa os sinais SIZ1 e SIZ0, e no caso do computador hospedeiro ser o



(a)



(b)



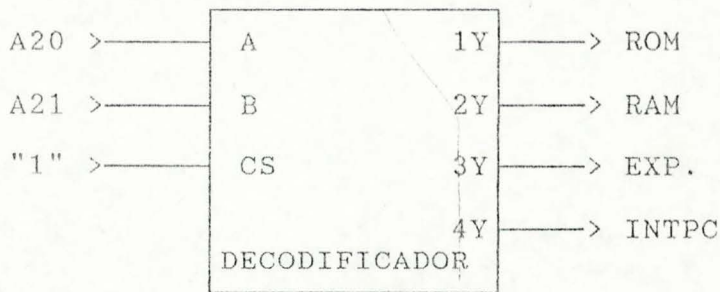
(c)

Figura 2.3.2.3 - Seleção do byte do banco da RAM para a interface (a); seletor das linhas A20,A21,SIZ0 e SIZ1 (b) e geração dos sinais 68/PC e IOCHRDY.

mestre, o tamanho da porta de dados é forçado para um byte ("01"). Os sinais SIZx vão então para a lógica da figura 2.1.2 .

2.3.3 Mapas de memória

Os endereços de memória foram gerados à partir dos bits A20 e A21, conforme mostra a figura 2.3.3.1(a). É usado um decodificador de 2-4, sendo na realidade os sinais A20 e A21 provenientes do seletor da figura 2.3.2.3(b), conforme explicado no capítulo anterior.



(a)

ENDEREÇO	DISPOSITIVO	TAM. REAL
[0 - 1M)	ROM	64 KB
[1M - 2M)	RAM	32 KB
[2M - 3M)	EXP.	
[3M - 4M)	INTPC	0

(b)

Figura 2.3.3.1 - Geração dos sinais de seleção e de interrupção (a) e mapa de endereços (b).

O mapa de endereços gerados é mostrado na figura 2.3.3.1(b). Como não se faz uso extensivo de memória, os bits A22..A31 são desconsiderados. Os blocos de seleção são grandes, 1 MB, dando liberdade de expansões futuras. Ainda resta um sinal de

seleção, EXP., para um eventual bloco de memória ou outra função qualquer. O sinal INTPC gera um pedido de interrupção ao computador hospedeiro, um PC-XT, e deve ser ligado portanto a um dos níveis disponíveis no mesmo.

2.4 Sistema de interrupção bidirecional

A forma escolhida para que os processos que são executados no VETOR e no computador hospedeiro se comuniquem foi via um sistema bidirecional de interrupções. Isto objetiva com que os dois sistemas realmente possam executar seus processos em paralelo, de acordo com a aplicação, evitando que um dos sistemas tenha que ficar esperando pelo outro num laço de programa improdutivo.

O VETOR pede interrupção ao computador hospedeiro quando endereça qualquer posição do quarto mega ([3M - 4MB)) da RAM, como mostram as figuras 2.3.3.1 (a) e (b).

O computador hospedeiro pede interrupção ao VETOR quando escreve numa determinada posição de memória, como pode ser visto na figura 2.4.1 .

O sinal BR é o gerado na figura 2.3.2.2(a), isto é, quando o computador hospedeiro, um PC-XT, endereça a região tabelada na figura 2.3.2.2(b). As linhas de endereço A03..14 devem estar em "1" e a operação deve ser de escrita. Portanto a região que pode gerar um pedido de interrupção, num ciclo de escrita, são os últimos oito bytes de uma das alternativas da

tabela mencionada. A região é pequena para não consumir memória,

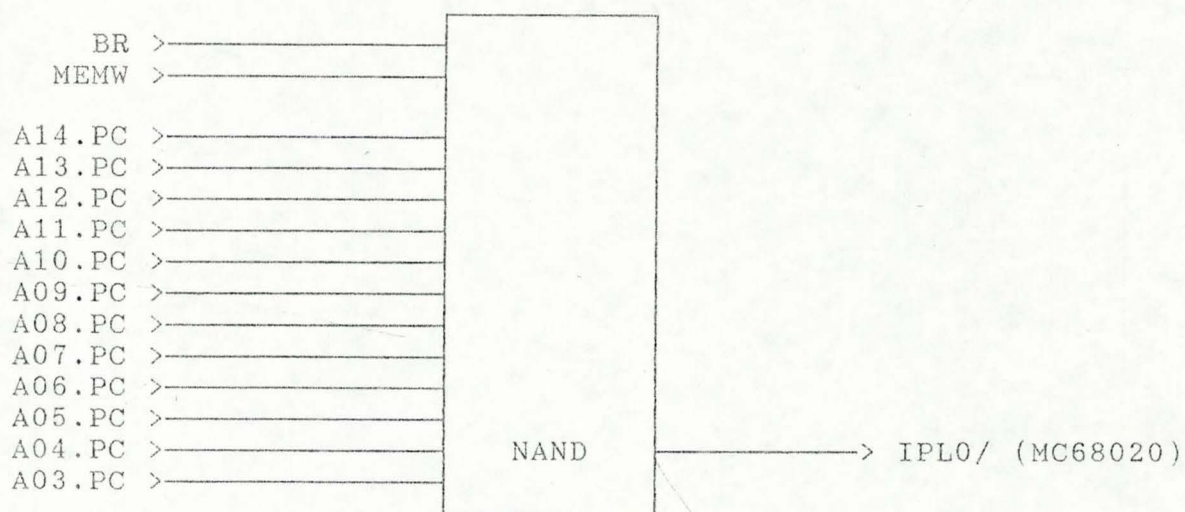


Figura 2.4.1 - Geração de interrupção a partir do computador hospedeiro, para o VETOR.

mas poderia ser só um byte. Quando do pedido de interrupção, a acesso eventualmente já poderia conter uma palavra de comando ao VETOR.

3. OBSERVAÇÕES E CONCLUSÕES

O projeto de um sistema multimicroprocessador para processamento numérico é muito interessante, pois abre caminho para pesquisa em vários ramos da informática, embora o paralelismo procurado seja só em função de processamento numérico.

A pesquisa em processamento paralelo é muito necessária no atual estágio de desenvolvimento do país. As necessidades de alto desempenho numérico são grandes em: indústrias, pesquisas científicas e para o próprio desenvolvimento da informática. Mas é importante que tal desempenho seja feito a um custo aceitável. Atualmente, devido ao modelo empregado para a política nacional de informática, as opções na aquisição de computadores para a grande maioria de usuários se situam principalmente na família da microcomputadores IBM-PC (XT, AT e AT386). Estes sistemas têm um desempenho já bem determinados, com ou sem o uso de coprocessador numérico. Embora o desempenho de um computador PC-AT386 à 25 MHz e com NDP, frente a um PC padrão também com NDP, seja de um fator de até vinte (20), as necessidades de certos usuários tem subido acima deste fator. Portanto a solução deste problema não é simplesmente solucionada com melhores tecnologias em microeletrônica, isto é, com CIs mais rápidos, mas sim o substancial desenvolvimento de arquiteturas paralelas.

As soluções de baixo custo envolvem necessariamente produtos com larga escala de produção. Dentro desta visão, o uso de processadores numéricos com operação tipo coprocessador foi a

melhor solução encontrada. Os NDPs coprocessadores começaram a ser implementados nos fins dos anos 70, primeiramente foi o 8087 da INTEL. Isto era necessário, visto que sempre existe uma necessidade de processamento numérico que não é coberta nem pelas melhores UCPs. Esta solução, a implementação de um processador numérico num CI separado, foi utilizada porque não se tinha escala de integração suficiente para a implementação do mesmo, dentro da UCP original. A filosofia de coprocessador foi uma evolução dos processadores numéricos que funcionavam como periféricos, a fim de que a integração de um NDP ao sistema fosse entendida pelos usuários como apenas a inclusão de novas instruções e novos registradores. Os NDPs de hoje são bastante velozes, têm um rico conjunto de instruções, um bom número de registradores, uma precisão suficiente para a maioria das aplicações científicas e tem um custo moderado. Por estes atributos eles foram os escolhidos para implementar o primeiro sistema do PNAD, o VETOR.

Surge uma incógnita a partir da seguinte constatação: hoje em dia, quando a escala de integração ultrapassa um milhão de transistores, já é possível a incorporação do NDP à UCP. Exemplo disto é o microprocessador 486 da INTEL que incorpora a UCP, o NDP, a gerência de memória e a memória cache num unico CI. Para a próxima década, a tendência nas UCPs da alta desempenho é implementação de microprocessadores com os recursos do 486. Mas aí se verifica que desenvolvimento de novos coprocessadores numéricos implementados em CIs separados vai terminar, ou pelo menos se reduzir. Restará aos grupos de Arquitetura de Computadores desenvolverem seus próprios CIs, com NDPs ou vetores

de NDPs, já que a escala de integração permitirá tais pesquisas.

O projeto do VETOR enfrentou muitos obstáculos, principalmente os de suporte para projeto. O porte da lógica envolvida no VETOR exige ferramentas adequadas de CAD, de simulação, de emulação, e de desenvolvimento, entre outros. A ferramenta de CAD utilizada foi o OrCAD, que ainda traz alguns utilitários bastante úteis, como pode ser visto em /ORC 87/. Foi necessário o uso de um PC-AT com placa EGA, monitor colorido e um mouse para se utilizar o OrCAD, visto que era exigida uma UCP de bom desempenho, e uma tela de boa resolução para o usuário melhor se localizar e se movimentar no diagrama esquemático. Caso fosse utilizado um PC-XT, o tempo de execução dos utilitários não seria menor do que vinte minutos, tornando um trabalho interativo insuportável para um usuário. A resolução oferecida pelo padrão EGA força o uso de um mouse para se indicar e se realizar rapidamente um grande deslocamento dentro do esquemático.

O VETOR se propõe inicialmente a operar como um executor de macro-instruções gravadas na sua ROM, que ativam de forma ordenada os NDPs necessários, que recebem seus operandos via a MMP, que colocam os resultados (operandos e estado) na mesma MMP para que o computador hospedeiro os retire e que se sincroniza com o computador hospedeiro via um sistema bidirecional de interrupções.

O VETOR é na realidade um bom laboratório para ser estudada a proposta de um sistema multimicroprocessador para processamento numérico que utiliza como elementos processadores numéricos do tipo coprocessador. Foi constatada uma pequena

diferença entre a forma de operação do conjunto UCP+NDPs da MOTOROLA e o da INTEL, a primeira não trabalha com interrupções e a segunda trabalha. Porém é mostrado em /MOT 87B/ que a UCP só fica conectada ao NDP pelo tempo necessário as transferências de operandos. Aparentemente resulta que o desempenho do sistema de comunicação entre as UCPs e o resultado final para o sistema seriam os mesmos, mas só uma pesquisa mais aprofundada poderá tirar melhores conclusões.

Um trabalho muito interessante no VETOR é o desenvolvimento de algoritmos com paralelismo, primeiramente enfocando exemplos particulares, depois desenvolvendo-se algoritmos mais gerais e adaptáveis aos recursos do sistema e finalmente desenvolvendo-se compiladores para linguagens paralelas.

Quanto ao projeto lógico do VETOR existem duas melhorias que podem ser feitas, com vantagens e riscos: a troca da lógica de seleção de memória mostrada na figura 2.1.2 por uma PLA e a troca do seletor de endereços da figura 2.3.2.1 por um "buffer tri-state". A proposta de uso de quatro NDPs foi feita prevendo-se operações matriciais de 4x4 elementos, suficientes para pequenos calculos de trajetórias de um braço simples de robo. No entanto, podem ser utilizados até oito NDPs, com a expansão da lógica de decodificação correspondente e desde que exista espaço físico suficiente na placa à ser utilizada.

Atualmente, já existe um microprocessador da MOTOROLA mais ^econveniente para substituir a UCP do VETOR, é o MC68030, pois principalmente já incorpora uma memória cache de dados. Um

grande problema para os microprocessadores atuais de alto desempenho é o fato de necessitarem de uma memória com tempo de acesso rápido (sem ciclos de espera). Tais microprocessadores operam com frequências elevadas (20/25/33 MHz), exigindo muito da memória principal; mas o MC68020 e melhor ainda o MC68030 tentam minimizar este problema com auxilio de suas memórias caches internas. No caso do VETOR, o tipo de aplicação proposto inicialmente não exige memória de grande capacidade, mas certamente sugerirão aplicações em que vai se precisar de uma memória de grande porte. Neste caso as RAMs estáticas não podem ser empregadas pois tem pequena capacidade, e então serão necessários RAMs dinâmicas. Ai aparece uma defasagem tecnológica existente entre os microprocessadores e as memórias dinâmicas, a velocidade das ultimas são menores que os primeiros. Será necessário então recorrer ao uso de uma memória cache externa, de tamanho adequado a aplicação, que diminua o tempo de acesso médio a memória principal.

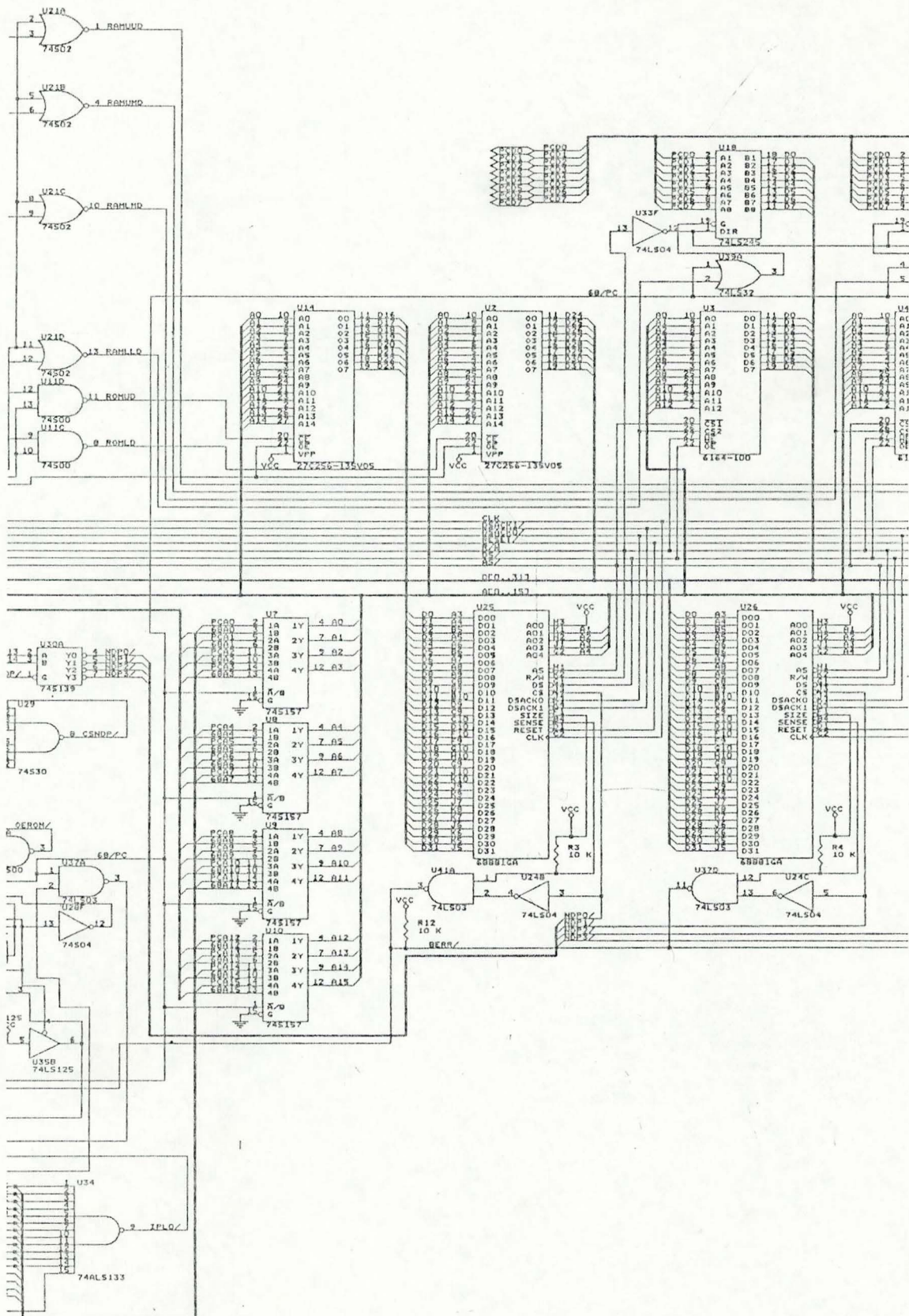
No VETOR, o tamanho de ROM pode ser reduzido para conter apenas um programa de inicialização, com isto reduzindo-se o número de tais CIs para um. Finalmente, um computador hospedeiro poderia suportar diversos sistemas multimicroprocessadores VETOR, a fim de realizar mais calculos quando necessário. Um computador com arquitetura IBM-PC pode suportar pelo menos seis placas, e neste caso seria necessário dar uma flexibilidade maior ao mapeamento da porta de oito bits da MMP do VETOR, a fim de que o espaço de cada uma não entre em conflito com a outras.

BIBLIOGRAFIA

- /INT 88/ INTEL. Microprocessor and Peripheral Handbook. Santa Clara, 1988. v. 1 .
- /MOT 84/ MOTOROLA. MC68020 32-Bit Microprocessor User Manual. Englewood Cliffs, Prentice Hall, 1984.
- /MOT 87A/ MOTOROLA. MC68030 Enhanced 32-Bit Microprocessor User Manual. 1987.
- /MOT 87B/ MOTOROLA. MC68881/MC68882 Floating Point Coprocessor User Manual. Englewood Cliffs, Prentice Hall, 1987.
- /NAS 81/ NASCIMENTO, F. R. Um Estudo Sobre Multimicroprocessamento. Porto Alegre, CPGCC/UFRGS, 1981 (Dissertação de Mestrado).
- /ORC 87/ ORCAD, Systems Corporation. Schematic Design Tools. 1987.

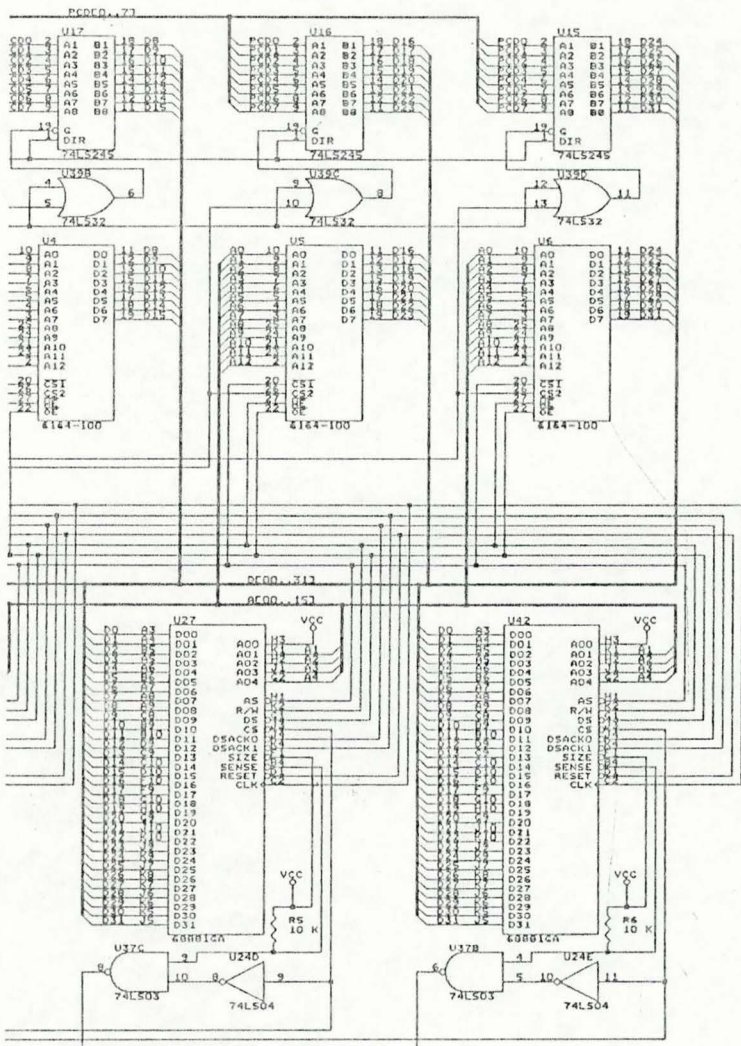
ANEXO 1

DIAGRAMA ESQUEMÁTICO



←-- continua na pg. 31

continua na pg. 33 -->



←- continua na pg. 32

UFRRS/CPD/DEPARTAMENTO DE INFORMÁTICA/PROCC			
Titulo	SISTEMA MULTIMICROPROCESSADOR V E T O R		
Serie/Documento	Numero	REV	
0	00HEC/CH-0-08.89	01	
Data:	July 26, 1989	Sheet	1 of 1

ANEXO 2

LISTA DE COMPONENTES

Item	Quantity	Reference	Part
1	1	U1	68020
2	2	U2,U14	27C256-135V05
3	4	U3,U4,U5,U6	6164-100
4	5	U7,U8,U9,U10,U40	74S157
5	1	U11	74S00
6	3	U12,U13,U19	74S64
7	4	U15,U16,U17,U18	74LS245
8	1	U20	74S10
9	1	U21	74S02
10	2	U22,U28	74S04
11	1	U23	74LS30
12	2	SW1,SW2	SW SPST
13	2	U24,U33	74LS04
14	7	R1,R3,R4,R5,R6,R9,R12	10 K
15	4	U25,U26,U27,U42	68881GA
16	1	R2	4.7 K
17	1	U29	74S30
18	1	U30	74S139
19	1	U31	74S74
20	1	X1	24 MHz
21	2	C1,C2	100 pF
22	2	R7,R11	2.2 K
23	1	U32	74LS00
24	2	R8,R10	220
25	1	C3	10 uF
26	1	D1	DIODE
27	1	U34	74ALS133

Item	Quantity	Reference	Part
28	2	U35,U43	74LS125
✓ 29	1	U36	74LS09
✓ 30	2	U37,U41	74LS03
31	1	U38	74LS02
32	1	U39	74LS32

A:\>

ANEXO 3

LISTA DE ÊRROS ELÉTRICOS

"VET4N1.SCH"

UNCONNECTED REPORT

X=	5.20	Y=	12.00	Output	U1,A22
X=	5.20	Y=	12.10	Output	U1,A23
X=	5.20	Y=	12.20	Output	U1,A24
X=	5.20	Y=	12.30	Output	U1,A25
X=	5.20	Y=	12.40	Output	U1,A26
X=	5.20	Y=	12.50	Output	U1,A27
X=	5.20	Y=	12.60	Output	U1,A28
X=	5.20	Y=	12.70	Output	U1,A29
X=	5.20	Y=	12.80	Output	U1,A30
X=	5.20	Y=	12.90	Output	U1,A31
X=	5.20	Y=	13.10	Output	U1,IPEND
X=	5.20	Y=	14.00	Output	U1,DBEN
X=	5.20	Y=	14.10	Output	U1,ECS
X=	5.20	Y=	14.20	Output	U1,OCS
X=	5.20	Y=	14.30	Output	U1,RMC
X=	5.20	Y=	14.80	I/O	U1,HALT

<<<ERROR>>>	CONFLICT	Found	OUTPUT	connected to HIZ	U35C,O
<<<ERROR>>>	CONFLICT	Found	OUTPUT	connected to OUTPUT	U26,DSACK1
<<<ERROR>>>	CONFLICT	Found	OUTPUT	connected to HIZ	U43A,O
<<<ERROR>>>	CONFLICT	Found	HIZ	connected to OUTPUT	U25,DSACK0
<<<ERROR>>>	CONFLICT	Found	OUTPUT	connected to OUTPUT	U26,DSACK0

A:\>