

MULTIMICROPROCESSADOR PARA APLICAÇÕES
EM TEMPO REAL (MATER)

POR

JOSÉ SÉRGIO DA ROCHA NETO

TESE DE MESTRADO

ORIENTADORES: Profº JOSÉ HOMERO F. CAVALCANTI
Profº GURDIP SINGH DEEP

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CAMPINA GRANDE (PARAÍBA), MARÇO DE 1982

MULTIMICROPROCESSADOR PARA APLICAÇÕES
EM TEMPO REAL (MATER)

POR

JOSÉ SÉRGIO DA ROCHA NETO

TESE DE MESTRADO

Apresentada à Coordenação Setorial
de Pós-Graduação e Pesquisa da Pró-
Reitoria para Assuntos do Interior
da Universidade Federal da Paraíba,
em cumprimento às exigências para
obtenção do grau de Mestre em Ciên-
cias.

CAMPINA GRANDE, MARÇO DE 1982



R672m	<p>Rocha Neto, José Sérgio da. Multimicroprocessador para aplicações em tempo real (MATER) / José Sérgio da Rocha Neto. - Campina Grande, 1982. 87 f.</p> <p>Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1982. "Orientação : Prof. José Homero F. Cavalcanti, Prof. Gurdip Singh Deep". Referências.</p> <p>1. Processador. 2. Multimicroprocessador. 3. Aplicação em Tempo Real (MATER). 4. Dissertação - Engenharia Elétrica. I. Cavalcanti, José Homero F. Cavalcanti. II. Deep, Gurdip Singh. III. Universidade Federal da Paraíba - Campina Grande (PB). IV. Título</p> <p>CDU 004.3(043)</p>
-------	--

S U M Á R I O

	Página
CAPITULO I	
INTRODUÇÃO	1
1.1 - CONTROLE DE PROCESSOS COM COMPUTADORES	2
1.2 - CONTROLE DE PROCESSOS COM MULTIPROCESSADORES	3
CAPITULO II	
ARQUITETURA DE MULTIPROCESSADORES	
2.1 - MULTIPROCESSADORES	5
2.1.1 - PERFORMANCE E CARACTERÍSTICAS	7
2.1.2 - SISTEMA OPERACIONAL	7
2.1.3 - ORGANIZAÇÃO DO HARDWARE	10
2.2 - COMPARAÇÃO DOS SISTEMAS EXISTENTES	15
CAPITULO III	
MULTIPROCESSADORES E CONTROLE DE PROCESSOS	
3.1 - ARQUITETURA PROPOSTA	18
3.2 - CARACTERÍSTICAS DA ARQUITETURA PROPOSTA ...	20
3.2.1 - MICROPROCESSADOR I (uPi)	26
3.2.2 - MICROPROCESSADOR SINCRONIZADOR (uPS)	26
3.2.3 - MEMÓRIA COMPARTILHADA (M.C.)	27

CAPITULO IV

IMPLEMENTAÇÃO DO MATER

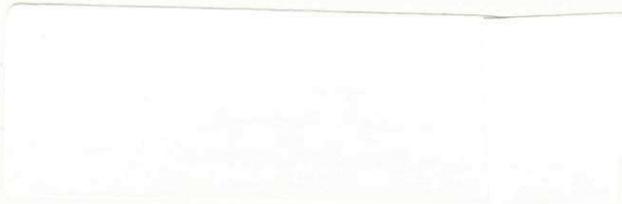
4.1 - CONSIDERAÇÕES DE HARDWARE	28
4.1.1 - MICROPROCESSADOR I (uPi)	29
4.1.2 - MICROPROCESSADOR SINCRONIZADOR (uPS)	37
4.1.3 - MEMÓRIA COMPARTILHADA (M.C.)	40
4.2 - CONSIDERAÇÕES DE SOFTWARE	40
4.2.1 - MICROPROCESSADOR I (uPi)	40
4.2.2 - MICROPROCESSADOR SINCRONIZADOR (uPS)	45
4.2.3 - MEMÓRIA COMPARTILHADA (M.C.)	46
4.2.4 - O GMC-MATER	50

CAPITULO V

CONCLUSÕES	57
APENDICES	60
APENDICE A - OPERAÇÃO DO MATER	61
APENDICE B - EXPANSÃO DO MATER	63
APENDICE C - CIRCUITOS DO MATER	65
APENDICE D - PROGRAMAS DO MATER	71
BIBLIOGRAFIA	85

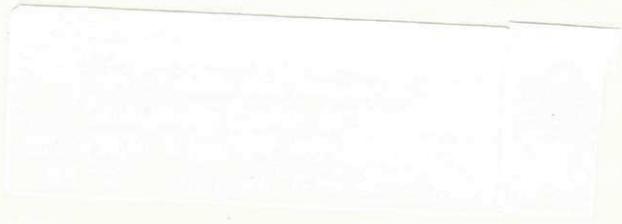
R E S U M O

Um sistema de multiprocessadores possui dois ou mais processadores que compartilham recursos comuns de HARDWARE e SOFTWARE. Neste trabalho se apresenta um estudo de algumas arquiteturas de multiprocessadores e as formas de acesso aos recursos compartilhados. É desenvolvido um multiprocessador para aplicações em tempo real com oito microprocessadores M6800 com seus interfaces de E/S e memória particular compartilhando um banco de memória comum. O acesso a memória comum é controlado por um microcomputador sincronizador. O programa gerenciador da memória compartilhada é desenvolvido para permitir a troca de informações entre os processadores através da memória compartilhada.



A B S T R A C T

A multiprocessor system comprises of two or more processors sharing common hardware and software resources. In this thesis, a study of some multiprocessor architecture and forms of resource sharing is presented. A multimicroprocessor for real time applications, consisting of eight microprocessors M 6800, each with its private memory and I/O devices and sharing a common memory, has been developed. The access to common memory is controlled by a microprocessor based synchroniser. The program developed for management of common memory, permits information exchange between the processors via common memory.



À Adair, Rodrigo e Renata.

Meus agradecimentos

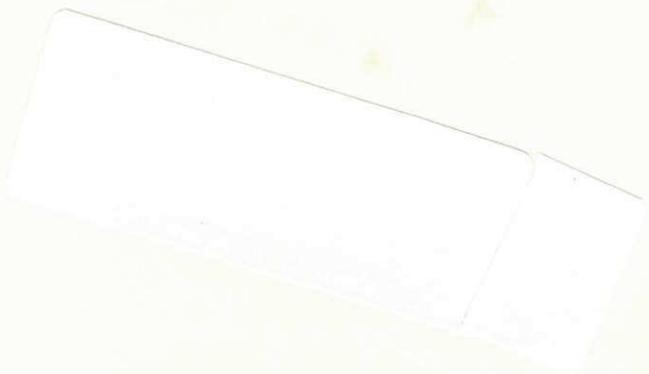
Aos professores Gurdip Singh Deep e José Homero Feitosa Cavalcanti pela valiosa orientação

Ao professor José Ivan Carnaúba Accioly pela sua orientação acadêmica

Aos professores da Faculdade de Engenharia de Joinville (SC) pelos incentivos demonstrados

E a todos que contribuíram na elaboração deste trabalho.





LISTAS DAS FIGURAS

Figura	Página
1.1 - ESTRUTURA DE NETWORK	4
1.2 - ESTRUTURA DE MULTIPROCESSADOR	4
2.1 - SISTEMA VIA COMUM	12
2.2 - SISTEMA DE VIA COMUM MULTIPLAS	13
2.3 - SISTEMA MULTIPORTAS	14
2.4 - SISTEMA CROSSBAR	16
3.1 - DIAGRAMA DE BLOCOS DO MATER	19
3.2 - MICROPROCESSADOR I (uPi)	21
3.3 - DIAGRAMA MAIS DETALHADO DO MATER	22
3.4 - MAPA DE MEMÓRIA DO uPi	24
3.5 - ENDEREÇOS DOS DISPOSITIVOS DE E/S DO uPi	25
4.1 - MICROPROCESSADOR I - ESTRUTURA DE UNIPROCESSADOR ..	30
4.2 - PROTOTIPO DO MATER	31
4.3 - PLACA DO MICROPROCESSADOR I (uPi)	32
4.4 - PLACA DE E/S DO uPi	33
4.5 - BUS DO MATER - VIA MATER	36
4.6 - ARQUITETURA DO SINCRONIZADOR (uPS)	38
4.7 - MAPA DE MEMÓRIA DO uPS	39
4.8 - MAPA DE MEMORIA DA M.C.	41
4.9 - FLUXOGRAMA DO SATER.1	43

4.10 - FLUXOGRAMA DO SATER.1 (SUBROTINAS).....	44
4.11 - FLUXOGRAMA DO GMC-MATER (SUBROTINA ACESSO)	51
4.12 - FLUXOGRAMA DO GMC-MATER (SUBROTINA ORGANIZA)	52
4.13 - FLUXOGRAMA DO GMC-MATER (SUBROTINA TAREFA)	53
4.14 - FLUXOGRAMA DO GMC-MATER (SUBROTINA FIM)	54

LISTA DE ABREVIATURAS OU SIGLAS

M.C.	- MEMÓRIA COMPARTILHADA
MATER	- MULTIMICROPROCESSADOR PARA APLICAÇÕES EM TEMPO REAL
ANSI	- AMERICAN NATIONAL STANDARD VOCABULARY FOR INFORMATION PROCESSING
E/S	- ENTRADA/SAIDA
C.P.U.	- CENTRAL PROCESSING UNIT
uPS	- MICROPROCESSADOR SINCRONIZADOR
uPi	- MICROPROCESSADOR I
M.P.	- MEMÓRIA PARTICULAR
PIA	- PERIPHERAL INTERFACE ADAPTER
PIA-S	- PIA DE SINALIZAÇÃO
GMC-MATER	- PROGRAMA DE GERÊNCIA DA MEMÓRIA COMPARTILHADA
SATER.1	- SISTEMA DE DESENVOLVIMENTO DE PROGRAMAS DE APLICAÇÕES NO MATER
ACIA	- ASSYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER
uPV	- MICROPROCESSADOR VIRTUAL
uPM	- MICROPROCESSADOR MESTRE
RTC	- REAL TIME CLOCK
S.C.	- SINALIZAÇÃO COMUM
S.E.	- SINALIZAÇÃO ESPECÍFICA



CAPITULO I

INTRODUÇÃO

São várias as aplicações da engenharia nas quais necessita-se do controle dos parâmetros de um determinado processo. Este controle realiza-se em função das variáveis do processo que podem envolver grandezas físicas, químicas, etc. A velocidade com que estes processos devem ser controlados varia de um processo à outro. Atualmente, as técnicas empregadas no controle de processos estão evoluindo e envolvendo várias áreas da engenharia.

As técnicas de controle de processos por sistemas convencionais de malha fechada, sistemas analógicos, sistemas digitais (apenas hardware), etc, tem cedido lugar a aplicações de sistemas de computadores com um único processador, ou vários processadores interligados num mesmo sistema.

De modo geral, o controle de processos em tempo real necessita de grandes velocidades de operação, para que o processador além de fazer o controle dos processos, possam fazer a comunicação com operadores, outros sistemas, etc. Uma das técnicas utilizadas consiste em se utilizar processadores interligados num mesmo sistema, compartilhando os recursos (memória, dispositivos de entrada e saída de dados, etc.) deste sistema.

Este capítulo introduz as técnicas mais usuais no controle de processos com sistemas de computadores.

1.1 - CONTROLE DE PROCESSOS COM COMPUTADORES

Os computadores tem sido utilizados em controle de processos de várias maneiras e em diferentes aplicações tais como: o DATA LOGGING, o CONTROLE SUPERVISÓRIO e o CONTROLE DIGITAL DIRETO.

No DATA LOGGING, os dados do processo sobre controle são coletados, analisados e tabulados de forma pré-estabelecida e após isso são colocados a vista do operador a intervalos de tempo regulares, ou então, quando são por este requisitados. O operador utiliza estes resultados para controlar diretamente o processo. O computador desempenha funções tais como: verificação de condições de alarme, conversão de medidas para unidades de engenharia, armazenamento de dados para futuras verificações, etc.

No CONTROLE SUPERVISÓRIO os dados do processo sobre controle depois de obtidos e processados, são utilizados para a execução de cálculos que fornecem os dados apropriados para os controladores que desempenham o controle do processo. O operador pode modificar neste tipo de controle, os parâmetros (dados) de processamento do computador, se um novo valor de referência é desejado.

No CONTROLE DIGITAL DIRETO, o computador obtém os dados do processo sobre controle e utiliza-os para executar programas que simulam as funções dos controladores analógicos. O computador através de circuitos (interfaces) especiais, ajuda o atuador final (válvulas, chaves, etc.) para efetuar o controle dos processos. Os dados de referência para o controle desses processos, podem ser definidos pelo operador ou obtidos pelo próprio computador, desempenhando cálculos estatísticos, comparações, etc. O computador neste caso pode permitir uma ampla seleção de novos algoritmos de controle mais sofisticados, que podem ser utilizados como técnicas de controle mais avançadas. Todavia neste tipo de controle de processos, uma falha no computador significa uma perda no controle de todos os processos a ele associados. Por este motivo o CONTROLE DIGITAL DIRETO é sempre usado com um controle analógico associado, para operar em casos de falhas no compu-

tador.

1.2 - CONTROLE DE PROCESSOS COM MULTIPROCESSADORES

Nos sistemas de controle de processos em tempo real, o computador faz a aquisição dos dados do processo sobre controle, executa os programas de controle e envia os dados finais aos atuadores. Devido a baixa velocidade dos processadores que além de fazer o controle dos processos, permitem a comunicação com os operadores e outros sistemas, utiliza-se vários processadores interligados num mesmo sistema. [1] Um destes sistemas é o NETWORK, (fig. 1.1), neste sistema de ligação entre processadores, cada computador possui seu processador central e sua memória particular, a comunicação entre os processadores é feita através de canais de comunicação em serie e/ou paralelo. [1]

Uma outra solução é o multiprocessador (fig. 1.2). Neste tipo de sistema a principal característica é que a comunicação entre os processadores é feita através dos recursos compartilhados (memória, dispositivos de entrada e saída de dados, etc.) por todos os processadores. É necessário um sincronismo para a utilização destes recursos, afim de evitar que dois ou mais processadores tenham o acesso ao mesmo recurso, ao mesmo tempo. [1]

Este trabalho trata do projeto de um MULTIMICROPROCESSADOR PARA APLICAÇÕES EM TEMPO REAL (MATER). No capítulo 2, é apresentada uma descrição das arquiteturas de multiprocessadores, no capítulo 3, descreve-se as interrelações entre multiprocessadores e controle de processos, no capítulo 4, mostra-se a implementação de HARDWARE e SOFTWARE para o MATER e finalmente no capítulo 5, apresenta-se os comentários dos resultados e conclusões.

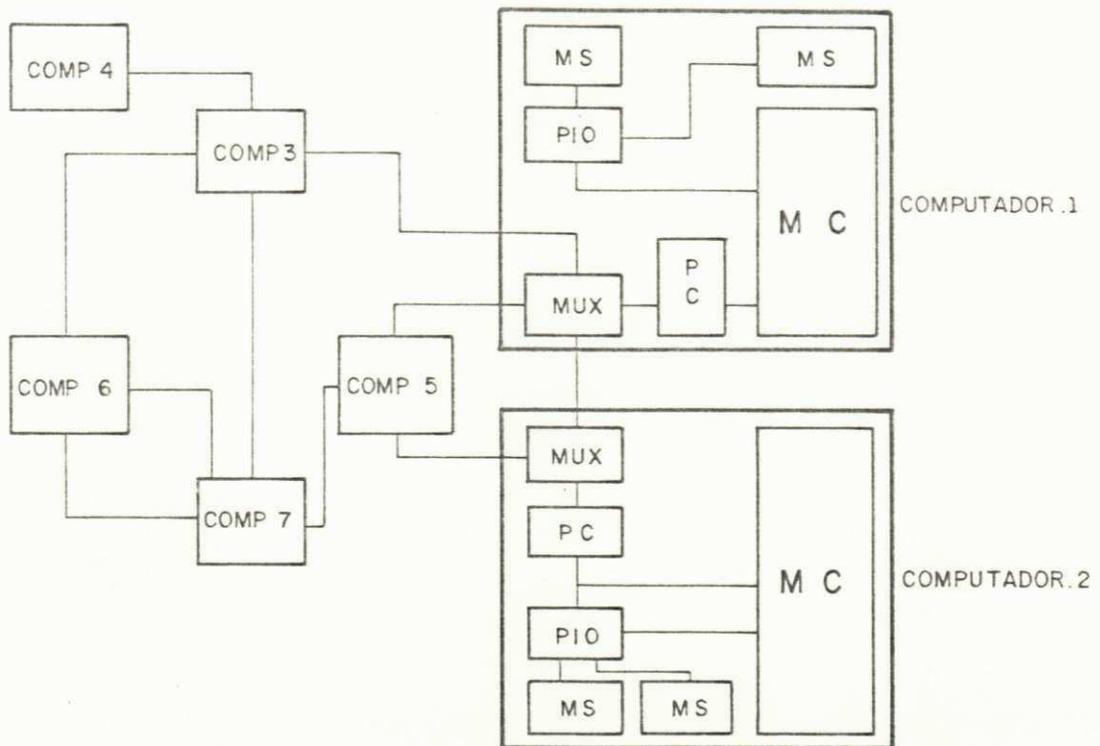
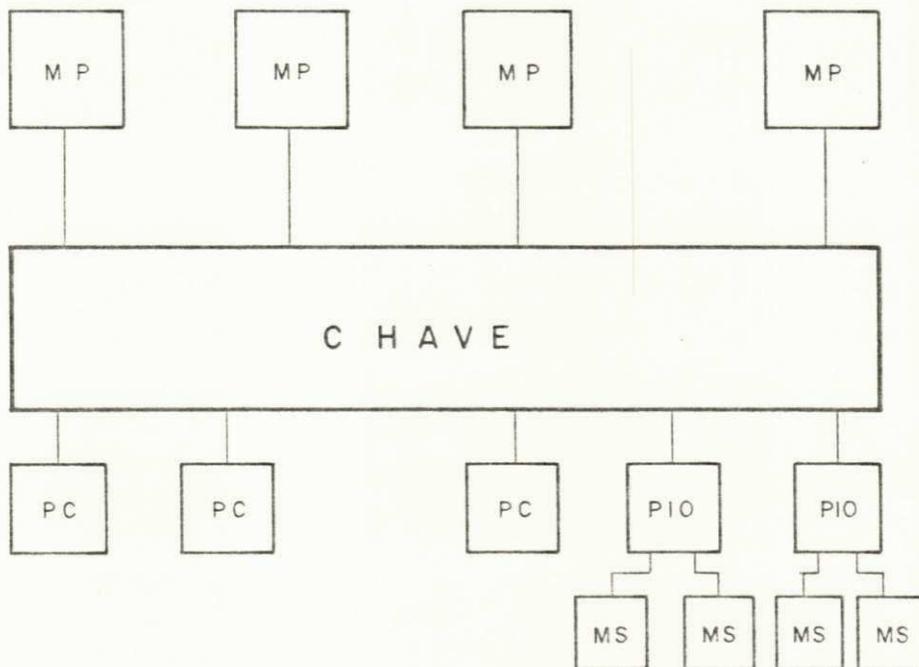


Fig. 1:1 - Estrutura de NETWORK.



LEGENDA: PC= Processador central, PIO= Processador de I/O, MP= Memória primária, MS= Memória secundária.

Fig. 1:2 - Estrutura de multiprocessadores.

CAPITULO II

ARQUITETURAS DE MULTIPROCESSADORES

Em muitos sistemas de multiprocessadores existe uma via (BUS) comum a todos os processadores de modo a permitir o acesso de cada um destes processadores aos recursos compartilhados (memória, dispositivos de entrada e saída de dados, etc.) Um dos requisitos destes sistemas é o controle desta via, de modo a permitir que um único processador tenha acesso a ela de cada vez, assim como evitar que um processador que deseje acesso a esta via, fique muito tempo ocioso, aguardando para que possa utilizá-la.

Existem diversos tipos de arquitetura de multiprocessadores de grande porte, onde soluções para este problema são apresentadas, tais como: o BURROUGHS D 825, o BENDIX G-21 e o IBM 360/65. [3]

Neste capítulo são apresentadas as arquiteturas mais usuais de multiprocessadores, e as técnicas de acesso aos recursos compartilhados, bem como suas principais características, tais como: performance, organização do sistema operacional, organização do hardware.

2.1 - MULTIPROCESSADORES

Enslow [4], apresenta a seguinte definição de um multiprocessador, proposta pela ANSI; "Um multiprocessador é definido

como sendo um computador envolvendo duas ou mais unidades de processamento sobre controle integrado." A definição acima é correta, todavia, não é suficiente. [4] Certamente o requisito para que um multiprocessador tenha controle integrado é extremamente importante, para que um multiprocessador tenha um sistema de operação integrada, entretanto, o controle de participação e interação entre os processadores, os quais são o centro das técnicas de multiprocessadores, não estão incluídas nesta definição. [4]

Um sistema de multiprocessadores pode incluir o compartilhamento direto da memória por todos os processadores, e o compartilhamento dos dispositivos de entrada e saída (E/S), por toda a memória e combinações de processadores. Embora neste ponto possa existir alguma qualificação sobre o compartilhamento de todos os recursos de um tipo particular de dispositivo (processadores, memória, etc.) este conceito é ainda mais amplo. O aspecto, mais importante das interações entre processadores, é o nível no qual estas podem ocorrer, do ponto de vista do controle a interação é possível entre JOB's completos ou entre passos individuais de JOB's. [4]

Enslow [4], Searle e Frebeg [5], definem um multiprocessador como sendo um sistema com as seguintes características:

- 1 - O multiprocessador contém dois ou mais processadores de capacidade comparáveis.
- 2 - Todos os processadores compartilham o acesso a memória compartilhada. (Memória Comum)
- 3 - Todos os processadores compartilham o acesso aos canais de E/S, unidades de controle e demais dispositivos.
- 4 - O sistema completo é controlado por um sistema operacional, que gerencia as interações entre os processadores e seus programas.

Pelo exposto, observa-se que existe muitas similaridades, entre multiplos-computadores e multiprocessadores, desde que ambos são desenvolvidos com o mesmo objetivo; suportar operações simultaneas. [4] Entretanto, existe uma diferença entre estes

dois tipos de sistemas e esta baseada no grau de extensão e compartilhamento dos processadores. Um sistema de múltiplos computadores consiste de vários computadores discretos e separados (apesar de existir comunicação direta entre si), enquanto que um multiprocessador é um computador simples, com múltiplas unidades de processamento. [4],[5]

2.1.1 - PERFORMANCE E CARACTERÍSTICAS

Quatro fatores determinantes da performance de um sistema servem como motivação a continuação do desenvolvimento de sistemas de processamento paralelo, de acordo com Enslow [4] e Nutt [6], são:

- 1 - Throughput
- 2 - Disponibilidade
- 3 - Flexibilidade
- 4 - Confiabilidade

O THROUGHPUT é o tempo médio que um processador necessita para processar um conjunto de instruções que ocorrem com a maior frequência. Este tempo é medido com o auxílio de um programa chamado BENCHMARK. As demais características estão interligadas com a realização física do sistema, assim sendo, a DISPONIBILIDADE de um sistema convencional é apenas dependente da eficácia dos módulos individuais deste.

A FLEXIBILIDADE mede como a configuração pode ser alterada com facilidade, isto é, se um sistema é altamente flexível é também altamente modular.

A CONFIABILIDADE está relacionada com a confiabilidade dos módulos individuais deste sistema, e os aspectos que influenciam as características acima citadas são: dispositivos e circuitos usados, a arquitetura do sistema e a organização do sistema operacional. [4, 6]

2.1.2 - SISTEMA OPERACIONAL

Um sistema operacional é um conjunto de programas residentes

tes em um processador, que atua como uma interface entre o usuário ou entre programas de aplicação do usuário e o hardware do processador. [7]

Os tipos de organizações que podem ser utilizados em projetos de sistemas operacionais para multiprocessadores, são: [5, 6]

- 1 - Mestre-escravo
- 2 - Executivo separado para cada processador
- 3 - Executivo simétrico ou anônimo para todos os processadores

SISTEMA OPERACIONAL MESTRE - ESCRAVO

Os primeiros sistemas de multiprocessadores projetados operavam no modo Mestre-escravo. (Um processador é designado como mestre e os demais como escravo.) Este tipo é o de mais fácil implementação, constituindo-se de uma simples extensão dos sistemas operacionais de uniprocessadores que incluíam características de multiprogramação. Embora o sistema Mestre-escravo seja simples, é usualmente ineficiente em seu controle e utilização dos recursos do sistema. Este tipo de sistema operacional apresenta as seguintes características: [5]

- 1 - Este tipo de sistema requer, comparativamente, simples software e hardware.
- 2 - Este tipo de sistema operacional, é eficiente para aplicações onde as tarefas são bem definidas, ou para sistemas assimétricos, nos quais o escravo tem uma capacidade menor que o mestre.
- 3 - O sistema total é sujeito a grandes fracassos que requerem a intervenção do operador para recomeçar, quando ocorre uma falha no processador designado como mestre.
- 4 - O sistema completo é comparativamente inflexível.
- 5 - A rotina executiva (programa que providência as inte-

rações entre o mestre e os escravos), é sempre executada no mesmo processador, se o escravo precisa de serviços que devem ser fornecidos pelo executivo, este pode requisitar o serviço e espera até que o programa corrente no processador mestre seja interrompido, e o executivo é despachado. O executivo e as rotinas, que este usa não são reentrantes (isto é, são executadas sempre pelo processador mestre.)

SISTEMA OPERACIONAL EXECUTIVO SEPARADO

O Sistema Operacional Executivo Separado para cada processador apresenta as seguintes características: [5]

- 1 - Cada processador atende suas próprias necessidades de software.
- 2 - É necessário que alguns dos códigos supervisores sejam reentrantes ou reaplicáveis, isto é, podem ser executados por mais de um processador.
- 3 - Cada processador ou cada executivo tem seu próprio conjunto de tabelas particulares (dados, tasks, etc.) embora algumas destas possam ser comum ao sistema inteiro, o que cria um problema de acesso as tabelas.
- 4 - Executivos separados não são sensitivos a falhas catastróficas como no sistema operacional mestre-escravo, entretanto, o reinício de um processador individual que apresenta falha é provavelmente difícil.
- 5 - Cada processador tem seu equipamento de E/S, files, etc.

SISTEMA OPERACIONAL EXECUTIVO SIMÉTRICO

Considerar todos os processadores, bem como todos os recursos simetricamente, ou como um bloco anônimo de recursos, é o mais difícil modo de operação, [6] entretanto, o resultado pode justificar o esforço. Este tipo de sistema operacional apresenta as seguintes características: [5]

- 1 - O mestre flutua de um processador para outro, embora vários processadores possam executar rotinas de serviços do supervisor ao mesmo tempo.
- 2 - Este tipo de sistema pode alcançar a melhor distribuição de tarefas e utilização de todos os tipos de recursos do sistema.
- 3 - Os conflitos em requisições de serviços são resolvidos pela prioridade, que pode ser um conjunto sobre controle estático ou dinâmico.
- 4 - Muitos códigos supervisores podem ser reentrantes, desde que vários processadores podem executar a mesma subrotina de serviços ao mesmo tempo.
- 5 - Os conflitos nas tabelas de acesso (a tarefas ou códigos supervisores) e tabelas de consultas podem ocorrer, todavia, este ponto pode ser controlado em cada caminho em que a integridade do sistema é protegida.

2.1.3 - ORGANIZAÇÃO DO HARDWARE

Em sistemas de multiprocessadores, dois fatores são muito importantes para a sua classificação em termos de organização de hardware [8], ou sejam, a topologia e o tipo de interconexões, entre as unidades funcionais. Examinando-se o sistema de chaves para acesso aos recursos compartilhados, pode-se identificar três fundamentais e diferentes sistemas de organização de hardware [3, 4, 5,] e [9], usados em multiprocessadores que são:

- 1 - Sistema de Via comum (common Bus System)
- 2 - Sistema Multiportas (Multiport System)
- 3 - Sistema Crossbar (Crossbar Switch System)

Embora o alcance completo do esquema de interconexões entre as unidades funcionais (processadores, memória, dispositivos de E/S, etc.) seja muito mais amplo do que o apresentado em [3], [4], e [5], existem vários outros sistemas de organizações para operação de processadores em paralelo, algumas destas organi-

zações são: Sistema não Homogeneo[1], Processadores Pipeline [10], Processadores Associativos [11], Pipelined Loop [12] , etc.

SISTEMA DE VIA COMUM

Dos sistemas acima citados, o Via Comum, é o de mais simples interconexão entre as unidades funcionais, (fig. 2.1). Neste sistema todas as unidades funcionais são conectadas em uma única via (BUS). As operações de transferência de dados ou tarefas são controladas pelo estado da via. Uma unidade (processador dispositivo de E/S) ao ordenar o início de uma transferência, primeiro verifica o estado da via. Se esta está disponível e o destinatário pode receber, então inicia-se a transferência. A unidade de recepção organiza as informações e responde ao sinal de controle de envio confirmando a recepção. Este é o conceito básico embora a operação completa não seja totalmente simples.

Este tipo de organização de sistema apresenta uma fácil implementação de hardware. Searle e Frebeg[5], citam dois tipos de sistema de via comum, o sistema de via comum simples (SINGLE TIME-SHARED BUS), como mostrado na fig. 2.1, e o sistema de via comum múltiplas (MULTIPLE TIME-SHARED BUS), como mostrado na figura 2.2. No primeiro tipo uma falha na via (BUS) significa uma falha no sistema, no segundo tem-se a necessidade de dispositivos ativos para controle das vias, o que permite o uso de prioridades e um maior grau de complexidade que no caso anterior.[5]

SISTEMA MULTIPORTAS

O sistema multiportas (MULTIPOINT SYSTEMS OR MULTIPOINT MEMORIES), usa múltiplas vias dedicadas, como mostrado na fig. 2.3. Neste sistema cada elemento passivo tem várias portas, cada uma ligada a uma CPU diferente. Isto aumenta em muito o número de interconexões e além disto cada elemento passivo requer uma forma diferente de aviso de recebimento para cada requisição dos diferentes processadores. Em alguns sistemas com esta configura-

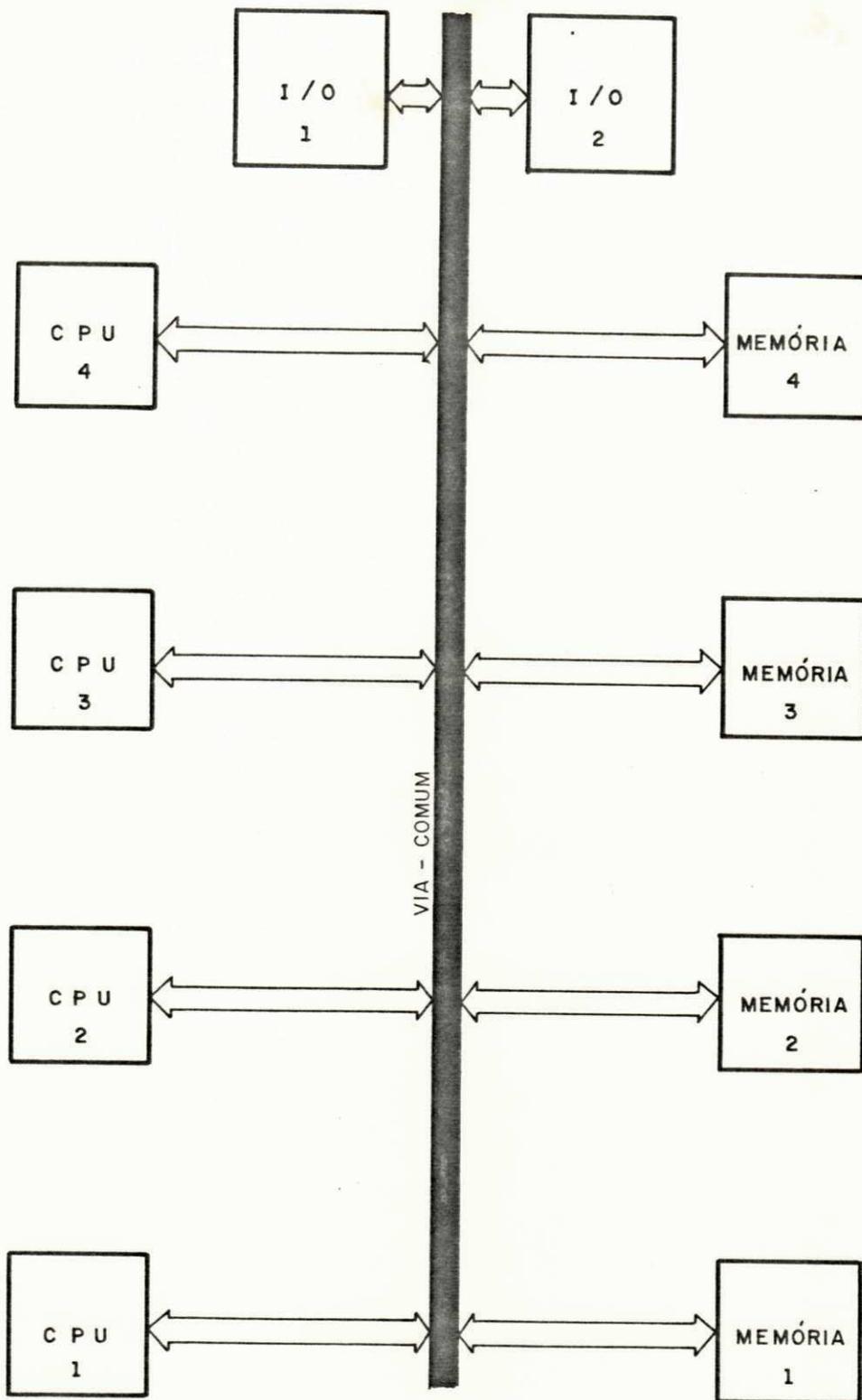


Fig. 2:1 - Sistema de via comum.
(Time - shared or commom-bus.)

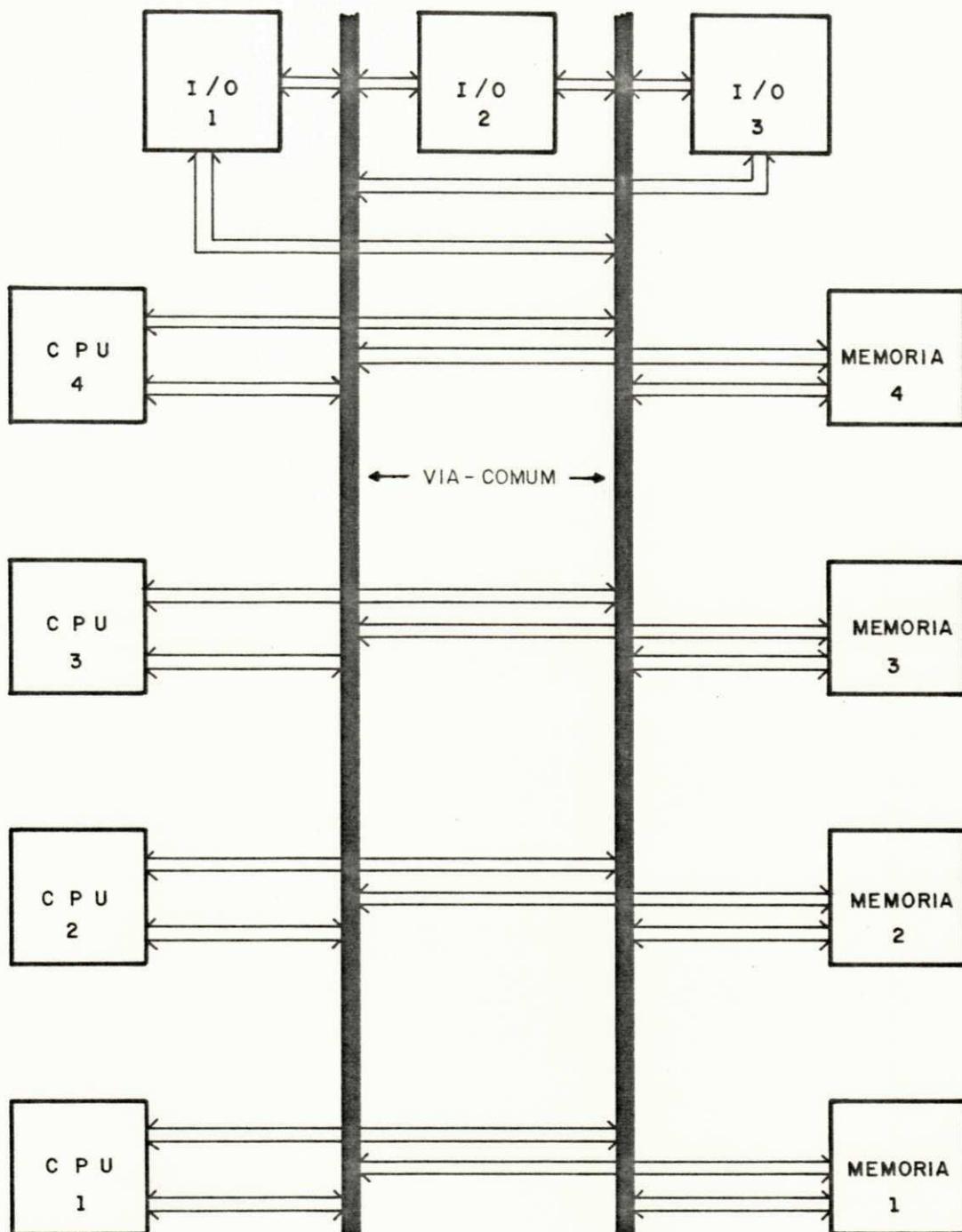


Fig. 2:2 - Sistema via comum multiplas.
(Multiple time shared bus.)

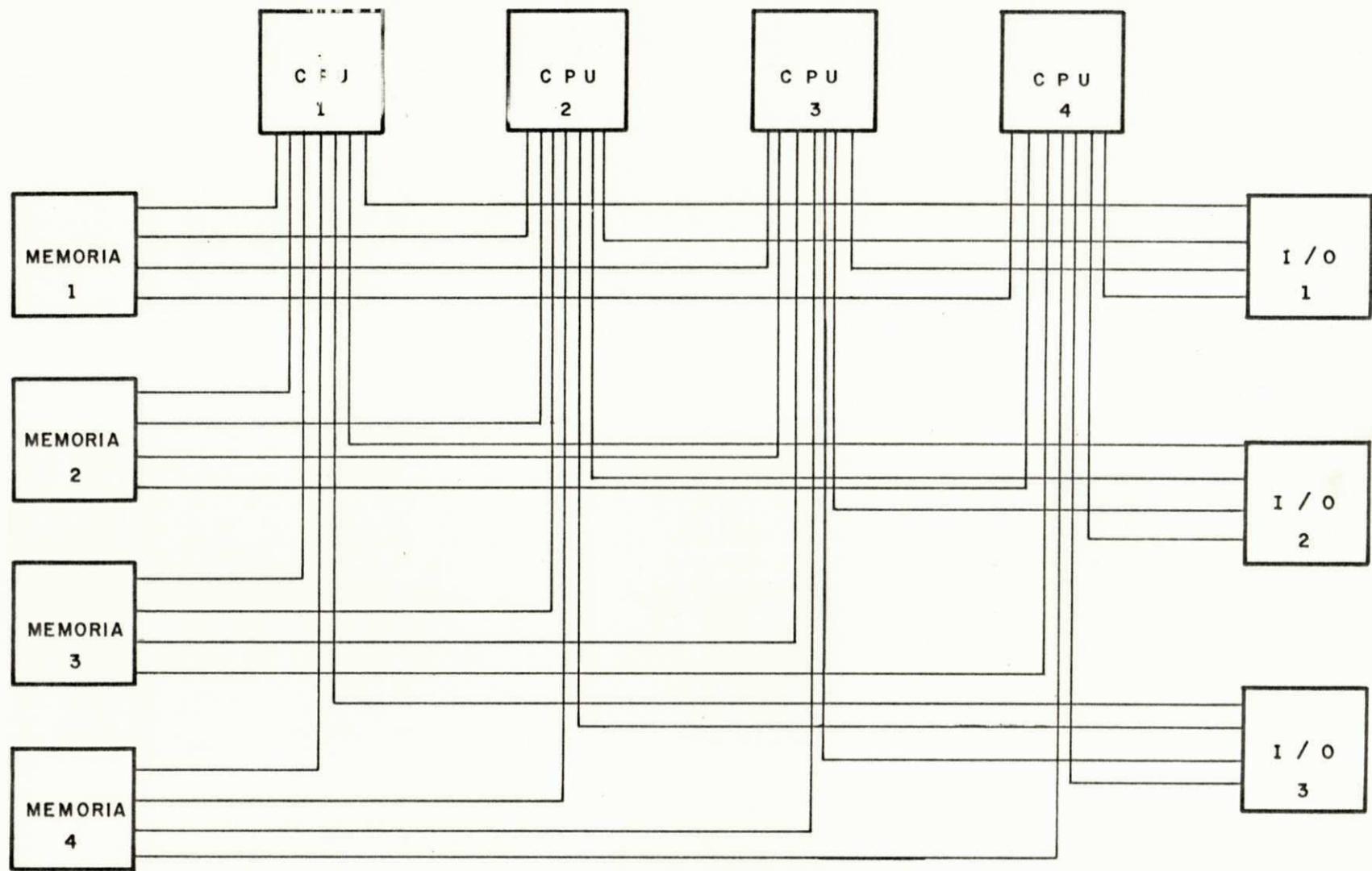


Fig. 2:3 - Sistema multiportas.
 (Multiport memories or multiport systems.)

ção, torna-se necessária uma definição prévia da capacidade máxima do Sistema, o que limita o seu grau de expandibilidade. Sistemas com grandes memórias usam este tipo de organização de hardware, [3],[5] além de uma certa prioridade pré-estabelecida para a interconexão das unidades de CPU com os módulos de memória.[4]

SISTEMA CROSSBAR

O Sistema Crossbar (CROSSBAR SWITCH SYSTEMS), consiste no método de se conectar M elementos de um determinado tipo com N elementos de uma outra classe, todos ligados um a um, como mostrado na fig. 2.4. As conexões múltiplas simultaneas são possíveis desde que estas sejam mutuamente exclusivas. Este método requer uma matriz de MxN chaves, além de um controle associado a estes pontos de interconexões. Tal configuração alcança o maior grau de transferência de informações entre suas unidades funcionais, todavia, a expandibilidade de sistemas com este tipo de interconexão é usualmente difícil. [4]

2.2 - COMPARAÇÃO DOS SISTEMAS EXISTENTES

Entre os fatores que podem ser considerados na comparação dos três tipos de organizações anteriormente descritas para avaliar seu uso em aplicações específicas, os óbvios são: throughput, flexibilidade, segurança, potencial de expansão e capacidade de transferência de informações entre as unidades funcionais.

O sistema Via Comum apresenta o hardware de menor complexibilidade e de mais fácil modificação pelo acréscimo ou remoção de unidades funcionais, todavia a falha da via significa a falha do sistema. Sua expansão é limitada pois pode-se degradar facilmente a performance total do sistema, com a adição de unidades funcionais e a eficiência efetiva do sistema é a mais baixa. Este tipo de organização é usualmente apropriada para pequenos sistemas.

O Sistema Multiportas apresenta um potencial para a taxa

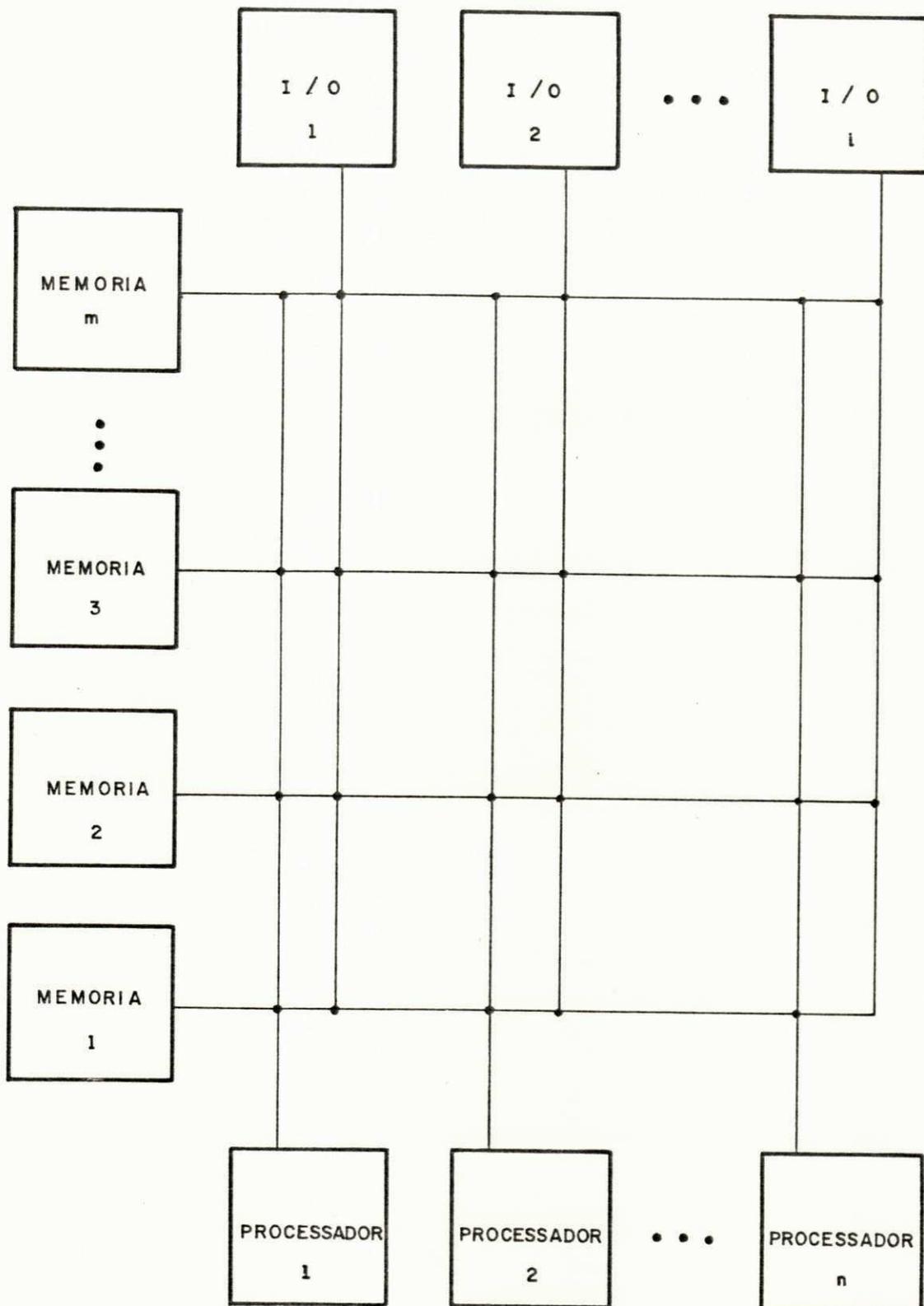


Fig. 2:4 - Sistema crossbar.
(Crossbar switch systems.)

de transferência de informações alto em relação ao sistema via comum e inferior ao sistema crossbar. Este tipo de organização requer uma unidade de memória maior do que os demais sistemas, desde que o controle das chaves (pontos de interconexões) está incluído nas unidades de memória. Em relação aos outros sistemas as unidades funcionais permite um relativo baixo custo para realização dos sistema, a partir de um único processador, enquanto que a dimensão e opção das configurações são limitadas pelo número e tipos de memórias disponíveis, e para as ligações das unidades funcionais, este tipo de sistema requer um grande número de cabos e conectores em relação aos demais sistemas.

O Sistema Crossbar, tem o mais complexo sistema de interconexão entre as unidades funcionais, apresenta a mais alta taxa de transferência de informações entre os processadores e a expansão do sistema é limitada apenas pelo tamanho físico da matriz de chaveamento. A eficiência do sistema é alta e a confiabilidade das chaves (pontos de interconexões) pode ser melhorada pela segmentação destas.

CAPITULO III

MULTIPROCESSADORES E CONTROLE DE PROCESSOS

Baseado nas características de aplicações em tempo real, tais como: confiabilidade, alta velocidade de resposta, comunicação em tempo real com operadores, memória comum, etc. E considerando-se um sistema de multiprocessadores como definido, por Englow [4] , Searle e Frebeg [5] , descrito anteriormente no capítulo 2, e objetivando as características de controle de processo por computadores, propomos um sistema composto de oito microprocessadores; estes processadores compartilham um banco de memória comum (memória compartilhada), através do qual com o auxílio de um microprocessador sincronizador (uPS) [13] , é realizada comunicação entre os microprocessadores do Sistema; este sistema é designado como MATER (Multimicroprocessador para aplicações em tempo real), para a transferência de informações (dados, tarefas, etc.)

Neste capítulo, propõe-se uma arquitetura para MATER, em termos de Software e Hardware.

3.1 - ARQUITETURA PROPOSTA

A fig. 3.1, mostra em diagrama de blocos, a arquitetura proposta para o MATER. Todos os microprocessadores do MATER tem acesso a M.G. através de um bloco de chaves tri-state bidirecio-

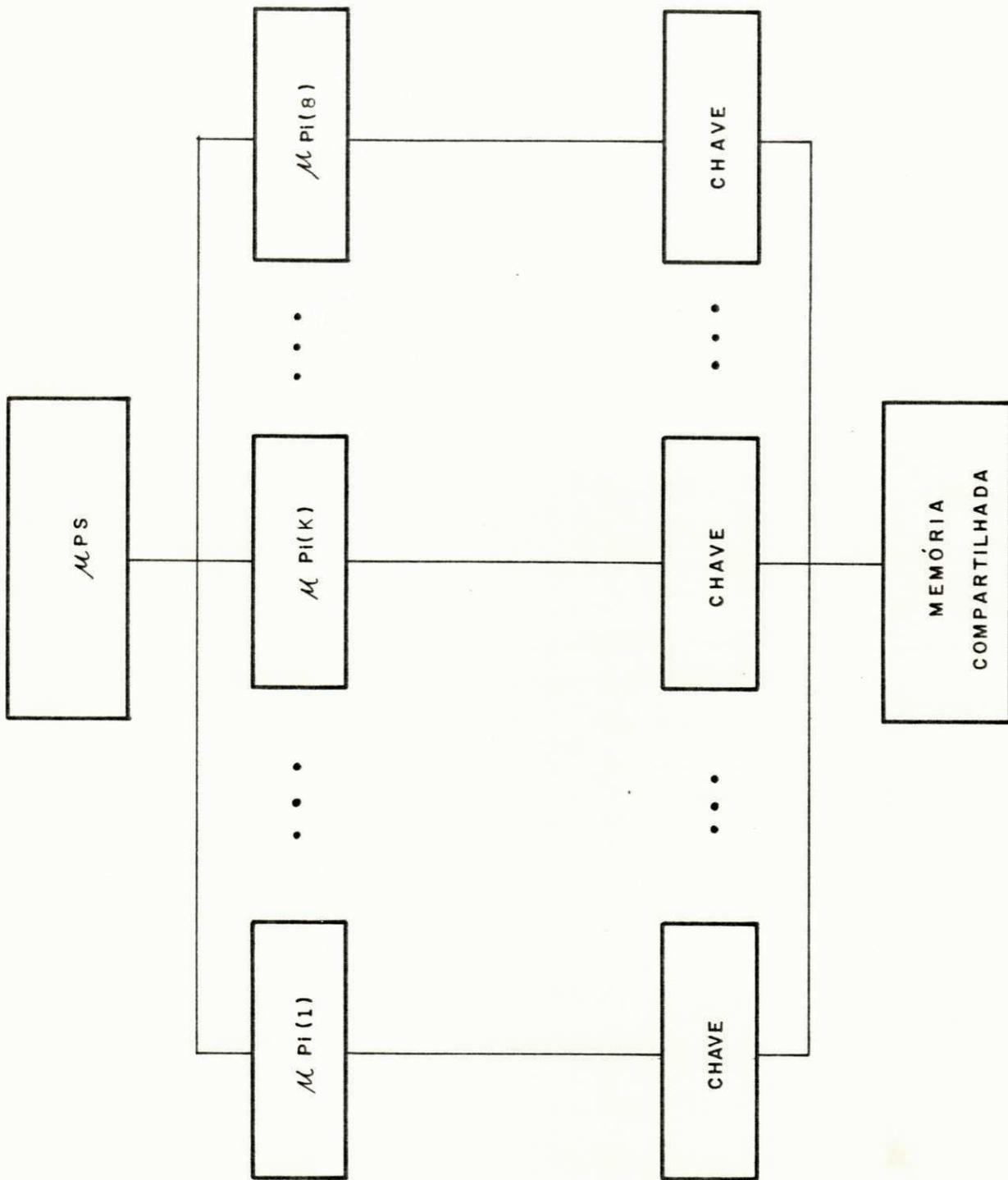


Fig. 3:1 - Diagrama de blocos do MATER.

nais, controladas com o auxílio de um dos processadores do MATER (uPi) pelo uPS. O uPS é o único microprocessador do MATER que não tem acesso a M.C. apenas controla (garante que um único uPi acessa a M.C. de cada vez) e analisa este acesso. Cada microprocessador (uPi) do MATER, possui sua memória particular (M.P.) composta de um bloco de RAM e outro de EPROM, suas interfaces de E/S, controle de interrupções, clock, circuitos de reset, codificadores necessários para o endereçamento destes dispositivos (RAM, EPROM, E/S) e um bloco de chaves tri-states bidirecionais que fornece o acesso a M.C., como mostrado na fig. 3.2.

Durante o funcionamento do sistema todos os uPi são assíncronos entre si. Quando um uPi necessita do acesso a M.C. este sinaliza o uPS [13] , através de uma PIA de sinalização (PIA-S), solicitando o acesso. Após ser concedida a permissão, a M.C. passa a ser tratada como se fosse uma parte da memória particular (M.P.) deste uPi, podendo este realizar tarefas de leitura e/ou escrita na M.C. Este processo é realizado sob a supervisão do software residente na M.C., designado de Programa de Gerência da Memória Compartilhada (GMC-MATER), gravado em EPROM na própria M.C., o GMC-MATER será descrito com detalhes os itens 3.2.3, e 4.2.3.

3.2 - CARACTERÍSTICAS DA ARQUITETURA PROPOSTA

Todos os uPi do sistema proposto tem a mesma estrutura fundamental em termos de hardware e software, as pequenas divergências ocorrem em dois pontos principais: Nas interfaces para E/S (PIA ou ACIA) usada em cada uPi, dependendo do dispositivo de E/S associado ao uPi (TTY, terminal de vídeo/teclado, outro processador, etc.), e no software necessário para controle de transferência de dados através destas unidades de E/S.

Na fig. 3.3 cada uPi do MATER é apresentado com maiores detalhes. A M.P. de cada uPi é composta de 4K-bytes de EPROM, usada para armazenar o software básico do MATER de acordo com as necessidades de cada uPi e 4K-bytes de RAM que podem ser usados pe-

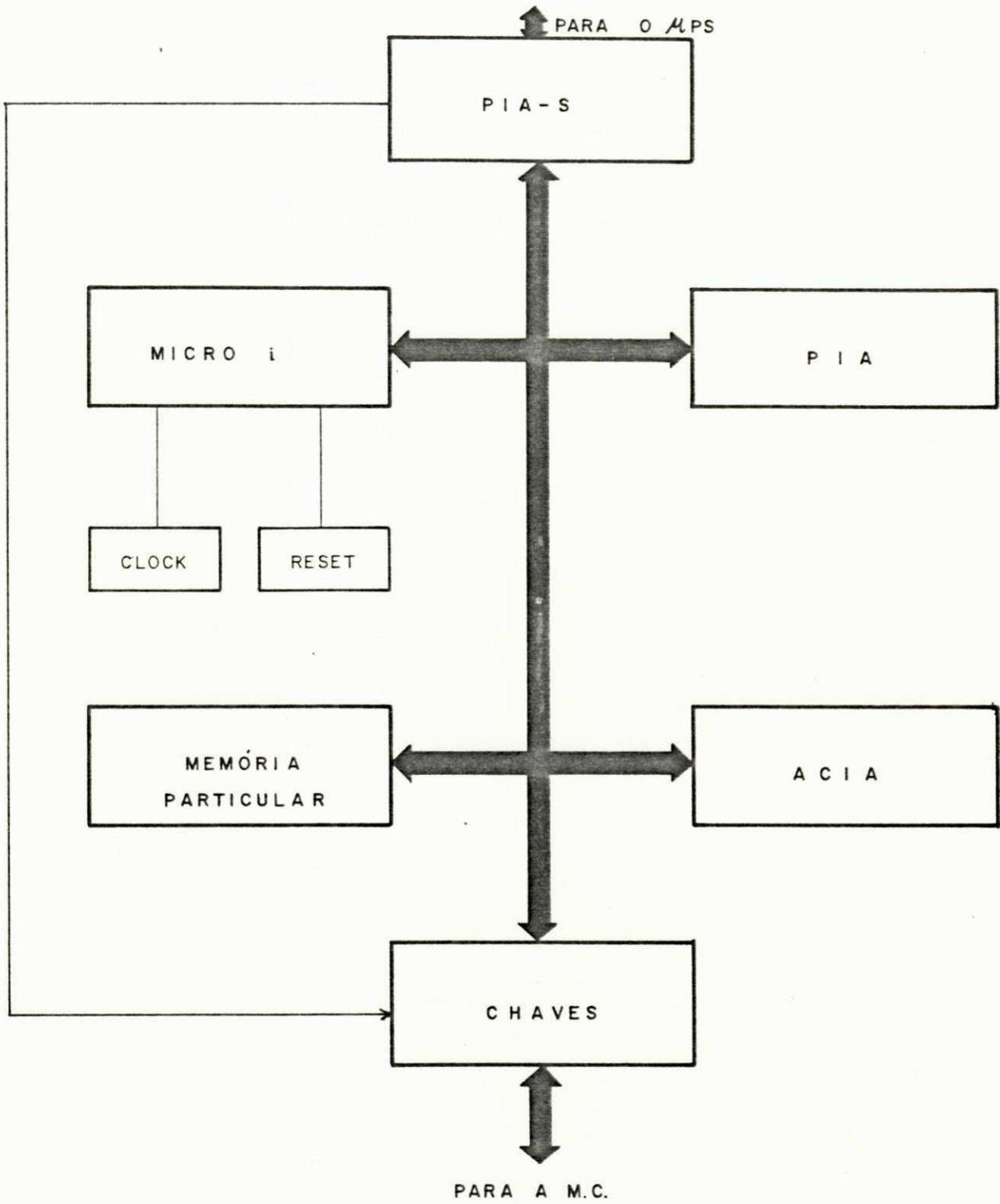


Fig. 3:2 - Microprocessador I (MP1).

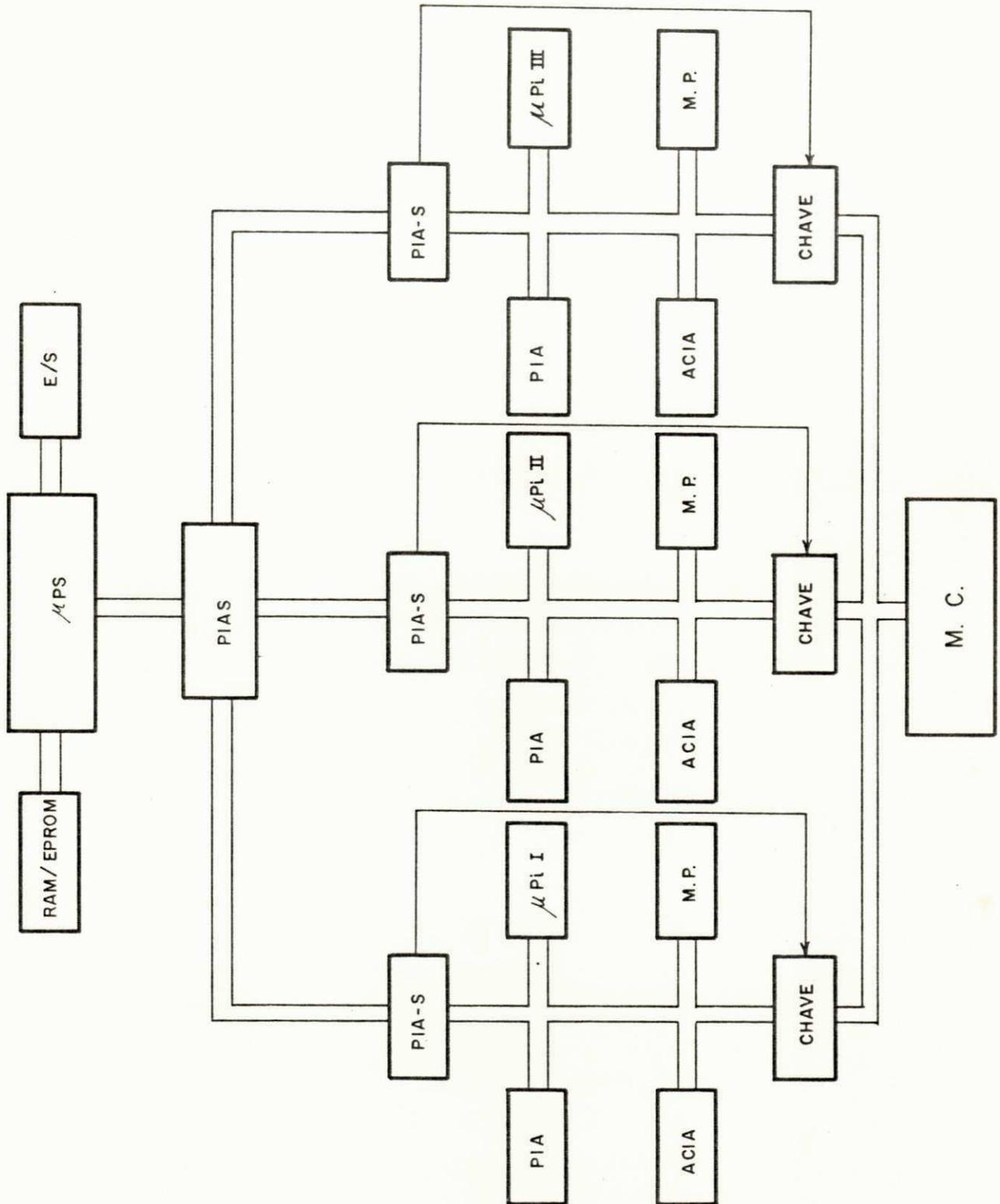


Fig. 3:3 - Diagrama mais detalhado do MATER. (PROTOTIPO)

lo usuário para propósito geral, salvo uma pequena área de 256 bytes, que é usada como RAM (rascunho) do SATER.1 (Sistema de Desenvolvimento de Programas de aplicações no MATER) que será descrito com detalhes no item 4.2. A fig. 3.4, apresenta o mapa de memória de cada uPi do MATER. Observe-se que dos 64K-bytes endereçáveis por cada uPi, apenas 26K-bytes são utilizados, os 38K-bytes restantes podem ser usados em futuras expansões de memória ou dispositivos de E/S. (ver apêndice B-Expansão do MATER). A fig. 3.5, apresenta com detalhes o mapa de memória da área reservada para as unidades de E/S, um máximo de 12 bytes são utilizados, os 2036 bytes restantes podem ser utilizados em futuras expansões. Afim de permitir esta expansão todos os dispositivos (memória, E/S) de cada uPi do MATER, tem o seu endereço codificado com todas as 16 linhas de endereço, facilitando deste modo, a adição de novos blocos em endereços pré-determinados. Isto justifica o uso de um bom número de codificadores (multiplexadores, portas lógicas), e buffers que garantem o fan-out necessário a todos os dispositivos do uPi. O MATER foi projetado para operar com oito uPi por questões de ordem prática/financeira o presente protótipo é composto apenas de três uPi, além do uPS. [13]

A memória compartilhada do MATER compõe-se de um bloco de RAM (14K-bytes) e outro de EPROM (2K-bytes). A parte de EPROM é usada para armazenar o GMC-MATER, que controla a parte de RAM da M.C., guardando e organizando as informações (dados e tarefas) trocadas pelo uPi. A M.C. tem as seguintes características:

- 1 - A M.C. deve ser usada para um dos uPi realizar transferência de dados e tarefas para outro uPi.
- 2 - Dois tipos de mensagens existem na M.C. — um para transferência de informações entre os uPi e outro para transferência de dados e tarefas.
- 3 - O programa de controle da M.C. (GMA-MATER) está gravado em EPROM na própria M.C.
- 4 - A integridade dos dados armazenados na M.C. é garantida pelo GMC-MATER.

ENDEREÇO	DESCRIÇÃO
0000 a 0FFF	04 K - bytes de RAM. Memória particular.
1000 a 3FFF	12 K - bytes. Futuras expansões.
4000 a 7FFF	16 K - bytes. (2K EPROM, 14K RAM) Memória compartilhada.
8000 a BFFF	16K - bytes. Futuras expansões.
C000 a C7FF	02 K - bytes. Dispositivos de E/S. (Ver fig. 3:5)
C800 a EFFF	10 K - bytes. Futuras expansões.
F000 a FFFF	04K - bytes de EPROM. Memória particular.
ENDEREÇO	DESCRIÇÃO

Fig. 3:4 - Mapa de memória do μ Pi.

ENDEREÇO	DESCRIÇÃO		
	μ PI	μ PII	μ PIII
C000 C003	PIA - S	PIA - S	PIA-S
C004 C007	D	PIA-1	PIA-1
C008 C009	D	ACIA-1U	D
C00A C00B	D	ACIA-TV	D
C00C C00D	ACIA - TTY	D	D
C00E C00F	ACIA - PDP	D	D
C010 C013	D	D	PIA - 2
C014 C7FF	D	D	D

LEGENDA: "D"= Disponível para futuras expansões.

Fig. 3:5 - ENDEREÇOS DOS DISPOSITIVOS DE
 E/S DOS μ PI.

- 5 - A transferência de dados e o limite de espaço da M.C. são controlados pelos apontamentos do GMC-MATER.
- 6 - Os dados armazenados na M.C. são organizados em páginas pelo GMC-MATER.

3.2.1 - MICROPROCESSADOR I (uPi)

Cada uPi do MATER se encarrega de receber dados dos processos sobre controle ou de um outro uPi, através da M.C., processa-los e então estes resultados podem ser enviados ao processo sobre controle ou são armazenados para uso posterior. Necessita-se de um sistema operacional para o controle de processos, gerência de arquivos e comunicação entre os uPi e controle das unidades de E/S e gerência dos processos externos. Portanto, o sistema operacional do MATER além do SATER.1, constará de programas de controle dos processos e controle das interfaces de E/S, gerência de arquivos, etc.

3.2.2 - MICROPROCESSADOR SINCRONIZADOR (uPS)

Uma análise mais detalhada do uPS, em termos de projeto e implementação de hardware e software é feita por Silva [13] com todos os detalhes deste microprocessador, reportamo-nos aqui apenas ao essencial para o funcionamento do MATER, ou seja, o uPS apenas verifica se algum dos uPi solicitou acesso a M.C., e se este acesso pode ou não ser permitido, sem que haja ao mesmo tempo mais de um uPi trabalhando na M.C.

Quando o acesso é solicitado ao uPS, este verifica se há algum uPi trabalhando na M. C., esta verificação é realizada pelo sistema operacional do uPS, através de um programa chamado KERNEL [14], que sincroniza as diferentes tarefas do uPS, se não existe uPi na M.C. o acesso é permitido caso contrário o uPi espera um novo ciclo do uPS para então ter o acesso a M.C., ou seja, espera a liberação da M.C. pelo uPi que a está utilizando.

Durante a operação do MATER o uPS tem condições de forne-

cer ao usuário uma análise estatística do tempo de acesso de cada uPi a M.C., além de verificar quais dos uPi está ou não em operação. [13]

3.2.3 - MEMÓRIA COMPARTILHADA (M.C.)

Após o uPi ter obtido a permissão para acessar a M.C. este executa um programa que fecha as chaves de acesso a M.C., e a partir deste ponto a M.C. faz parte do uPi. (O uPS garante a exclusão dos demais uPi). [13] O passo seguinte é ir a uma determinada locação da M.C., e passar o controle do sistema operacional para o GMC-MATER. Passando este a controlar e realizar todas as operações de leitura e escrita, transferência de dados e controle da M.C., garantindo todas as características descritas no item 3.2.1.

O fluxograma bem como a listagem do GMC-MATER está incluído no apêndice D. Para o controle voltar para o sistema operacional é necessário que todas as operações tenham sido realizadas ou então que o uPi tenha sido interrompido neste caso uma subrotina do SATER.1 é executada pelo uPi sendo as chaves de acesso a M.C. abertas (M.C. não faz mais parte do uPi) e o uPS é sinalizado indicando a liberação da M.C., podendo outro uPi ter acesso a M.C. se assim desejar.

CAPITULO IV

IMPLEMENTAÇÃO DO MATER

A arquitetura proposta para o MATER no capítulo anterior, foi implementada com as seguintes características:

- 1 - Cada processador (uPi) do MATER possui sua memória particular e dispositivos de E/S.
- 2 - Cada processador possui seus circuitos de clock, reset e chaves de acesso a M.C.
- 3 - Durante o funcionamento do MATER todos os uPi são assíncronos entre si.
- 4 - Todos os processadores do MATER compartilham o acesso a um banco de memória comum (Memória Compartilhada).
- 5 - A comunicação entre os processadores é realizada através da M.C. sob controle do programa de gerência da M.C.
- 6 - O GMC-MATER garante a integridade das informações armazenadas na M.C.

Neste capítulo apresenta-se a arquitetura proposta com considerações de hardware e software e o programa de gerência da M.C. (GMC-MATER) é implementado.

4.1 - CONSIDERAÇÕES DO HARDWARE

No capítulo anterior discutimos a arquitetura do MATER, em

seguida analisa-se separadamente a estrutura de hardware dos três blocos básicos do MATER; Microprocessador I (uPi), Microprocessador Sincronizador (uPS) e Memória Compartilhada (M.C.)

4.1.1 - MICROPROCESSADOR I (uPi)

A figura 3.3, apresenta um esquema mais detalhado do MATER. Todo o sistema compõe-se de oito microprocessadores (uPi), e um microprocessador sincronizador (uPS) [13]. Todavia o prototipo compõe-se de apenas três microprocessadores (uPi) e o microprocessador sincronizador (uPS). Todos os uPi do MATER tem basicamente, a mesma estrutura em termos de hardware e software (itens 3.1 e 3.2) e podem individualmente serem considerados como um pequeno sistema de uniprocessador, como mostra a figura 4.1, neste caso não são necessárias as chaves de acesso à M.C. e a PIA-S.

Na montagem do prototipo usamos a técnica WIRE-WRAP, constituindo-se todo o prototipo (inclusive o uPS) de oito placas de montagem como mostrado na fig. 4.2. A fim de permitir uma maior modularidade e flexibilidade, cada uPi esta montado em duas placas. Uma placa (placa do microprocessador) é composta de microprocessador, memória particular (M.P.), buffers, codificadores de endereços, circuitos de clock e reset (fig.4.3) e chaves de acesso a M.C. A segunda placa, designada placa de E/S (I/O), (fig. 4.4), compõe-se das unidades de E/S (PIA e/ou ACIA), PIA de sinalização (PIA-S), codificadores de endereço e demais circuitos necessários a E/S do uPi, dependendo da unidade de saída (terminal de vídeo/teclado, TTY, outro computador, etc.) ligada a estes dispositivos, resultando deste modo seis placas para os três uPi. O uPS e a M.C. compõe duas placas do MATER, perfazendo um total de oito placas de montagem.

Os mapas de memória dos uPi do MATER, é mostrado na fig. 3.4. A memória particular (M.P.) compõe-se de dois blocos de 4 K-bytes cada. Um bloco de RAM 2114 [15], endereçavel de 0000 a 0FFF usada para proposito geral, estando as locações de memória de 0F00 a 0FFF (256 bytes), reservada para operações do SATER.1 (rascunho).

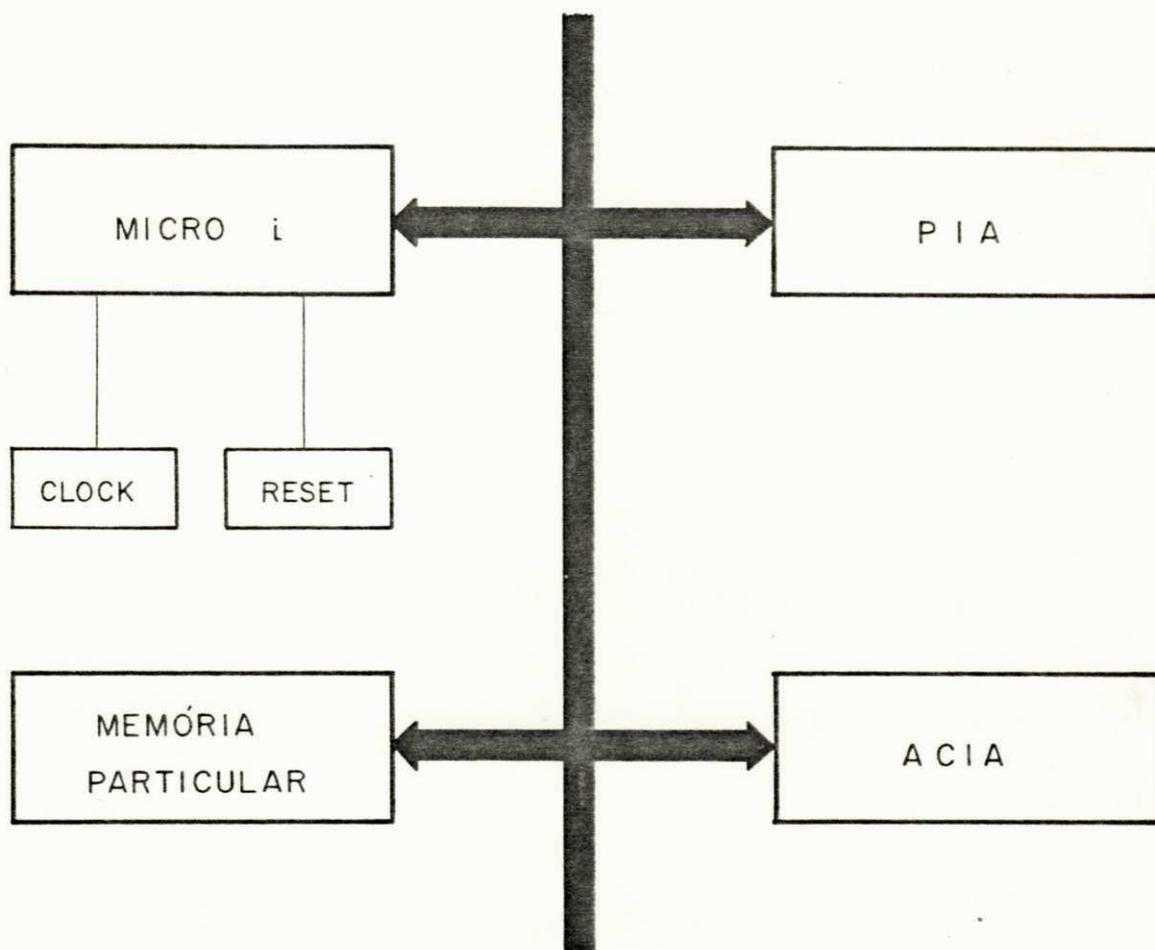


Fig. 4.1 - Microprocessador I (MPi) estrutura de processador.

UNIVERSIDADE FEDERAL DA PARAÍBA
 Pró-Reitoria para Assuntos do Interior
 Coa de Educação Superior de Pós-Graduação
 Rua Américo Veloso, 832 - Tel (083) 321-7222-R 355
 681.100 - *Campina Grande - Paraíba*

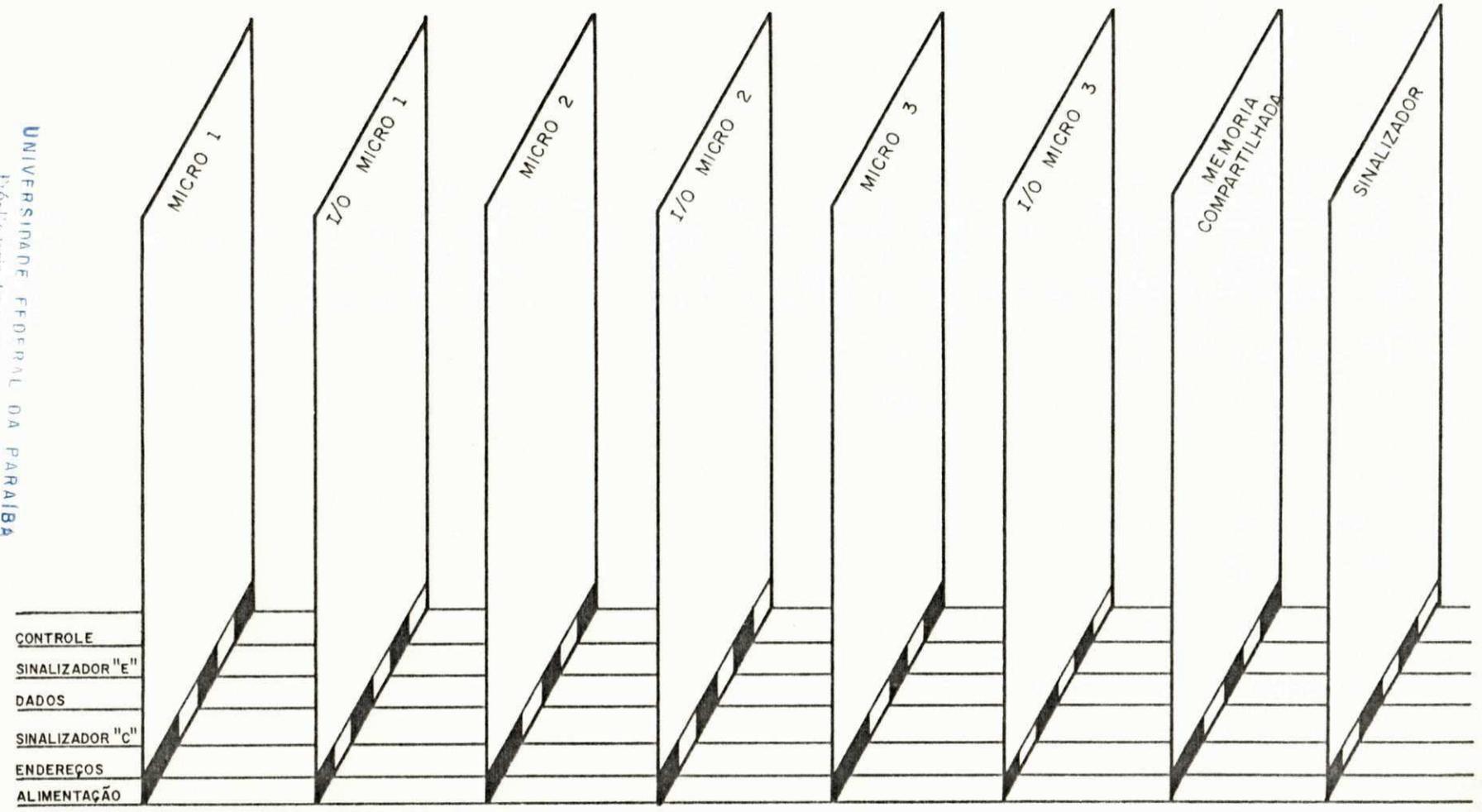


Fig. 4:2 - Prototipo do MATER.
 (Placas de montagem)

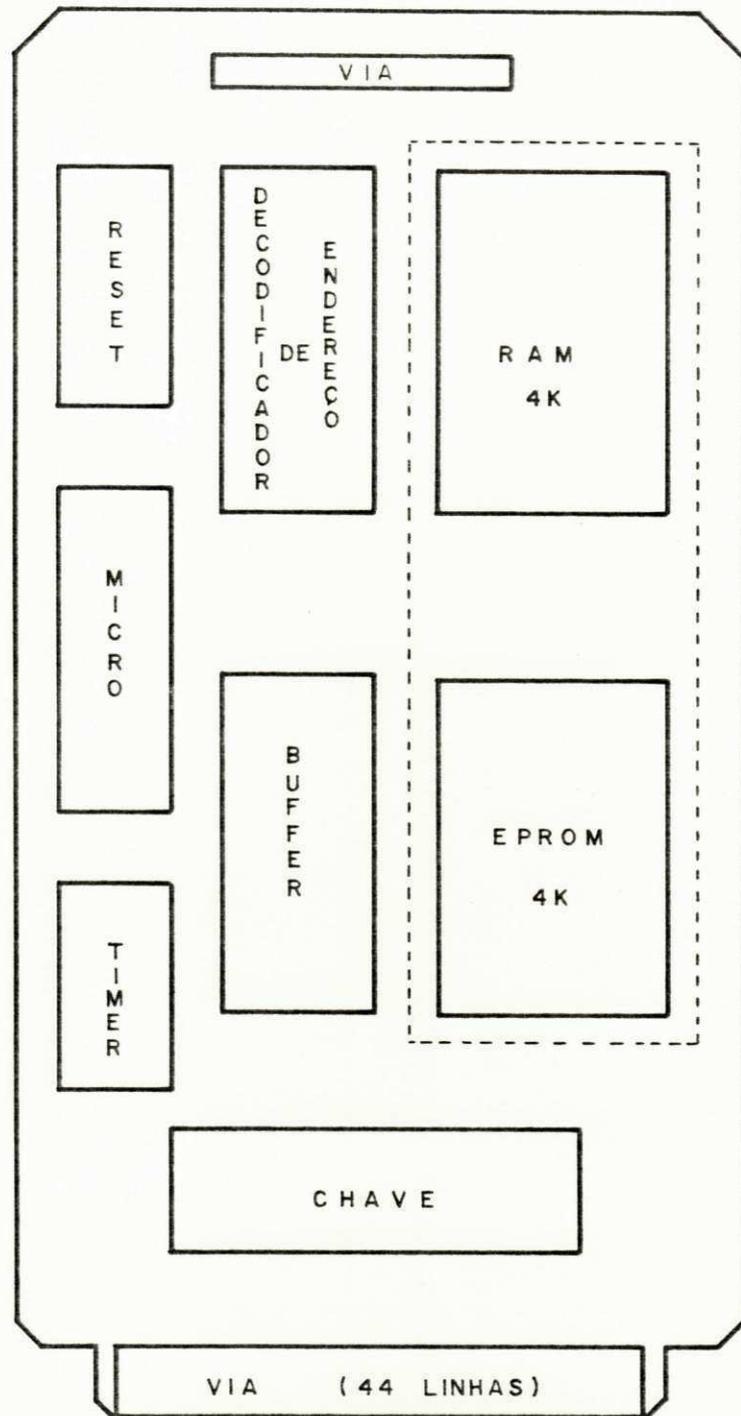


Fig. 4.3 - Placa do microprocessador I (MPi)

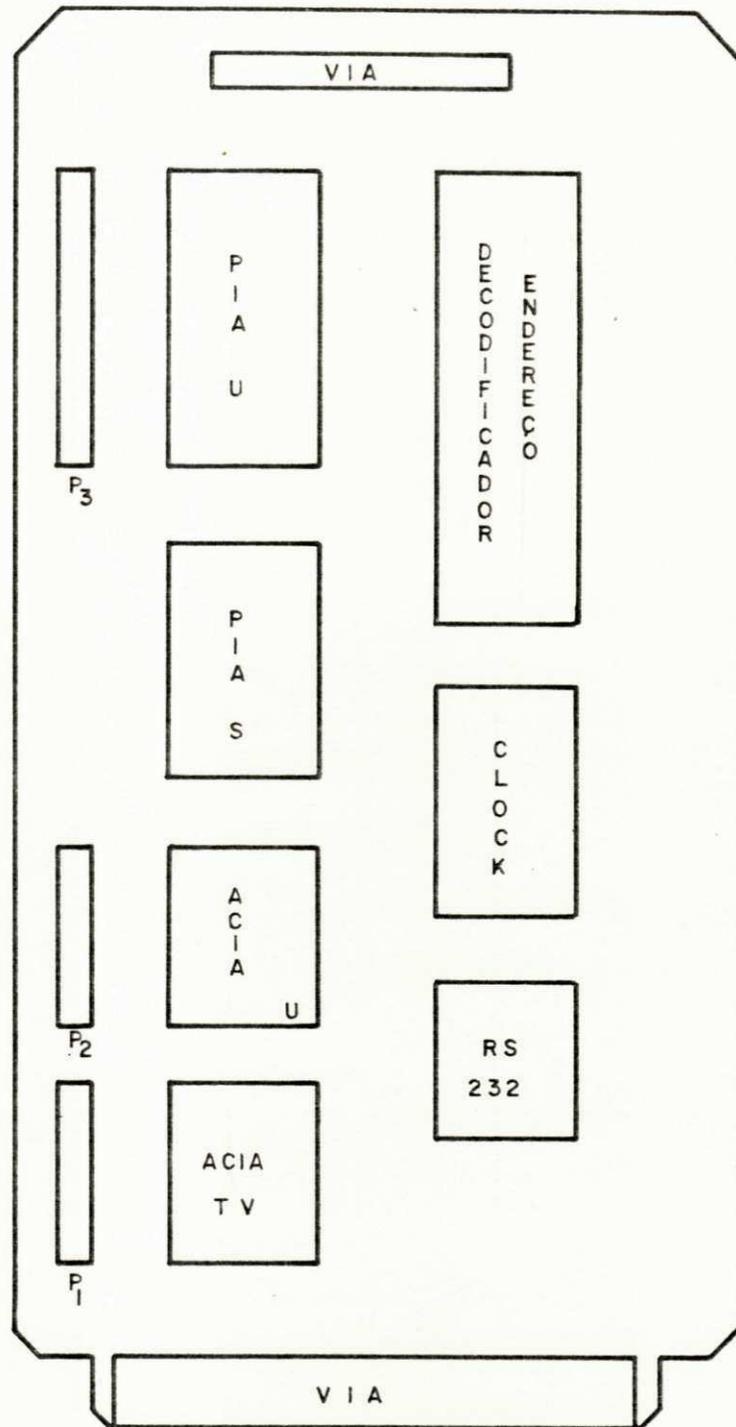


Fig. 4:4 - Placa de E/S do μPi

O outro bloco de 4K-bytes é de EPROM 2708 [16] endereçável de F000 a FFFF, é usado para armazenar o SATER.1 e as tarefas (ou programas) específicos do sistema operacional de cada uPi.

As locações de 4000 a 7FFF, compõe a M.C., descrita com detalhes no item 4.1.3. Para cada uPi, dispomos de 2K-bytes endereçáveis de C000 a C7FF, para as unidades de E/S, os detalhes desta parte da memória são apresentados na fig. 3.5. Nos três uPi, nos endereços de C000 a C003 esta localizada uma PIA 6820 [17], designada como PIA de sinalização (PIA-S), que faz a comunicação do uPi com o uPS. No uPi (I) dispomos de duas ACIAs 6850 [17], uma para ligação do uPi com a TTY e a outra para ligação com o PDP 11/34, nos endereços C00C - C00D e C00E - C00F, respectivamente. No uPi (II), dispomos de uma PIA para acesso a processos externos sob controle, (PIA-1C), endereçável de C004 a C007 e duas ACIAs, sendo uma para acesso a processos externos sob controle e outra para o terminal de vídeo/teclado, cujos endereços são respectivamente C008 - C009 e C00A - C00B. Finalmente no uPi (III), dispomos de duas PIAs para acesso a processos externos sob controle, PIA-1C e PIA-2C, endereçáveis de C004 a C007 e C010 a C013, respectivamente.

Nos três uPi do MATER, o circuito de clock é implementado com dois monoestáveis 74123 [18] de modo a gerar dois sinais de forma de onda quadrada, com frequência de 1 MHz defasados de 180 graus, que constituem os clocks ϕ_1 e ϕ_2 necessários ao funcionamento do microprocessador. O circuito de RESET é um latch implementado com duas portas NAND 7400 [18] de modo a eliminar o ruído da chave reset. Os clocks de transmissão e recepção de dados das ACIAs são gerados por um circuito oscilador que usa três portas inversoras CMOS 4049 [19], permitindo uma taxa de transmissão/recepção de 1200 bauds. (Com uma TTY ou terminal de vídeo/teclado associado a ACIA).

A codificação dos endereços dos dispositivos de cada uPi é realizada com o auxílio de gates lógicos e multiplexadores, no apêndice C, estão incluído os diagramas de ligações do MATER.

Em cada uPi do MATER os endereços de 1000 a 3FFF (12K-by

tes), 8000 a BFFF (16K-bytes) e C800 a EFFF (10K-bytes), perfazendo um total de 38K-bytes são destinados a futuras expansões do sistema, cada uPi ou M.C. (Ver apendice B - Expansão do MATER)

Afim de garantir uma boa margem de segurança, evitar sobrecarga no processador, e facilitar futuras expansões, empregamos em cada uPi, buffers para garantir o fan-out necessário e suficiente. Nas linhas de endereços utilizamos o buffer tri-state unidirecional 74365 [18] e nas linhas de dados usamos o buffer tri-state bidirecional 8216 [20].

As chaves de acesso a M.C. controladas pelo uPi com a permissão do uPS [13] através da PIA-S foram realizadas com buffer tri-state bidirecional 8216 [20].

As oito placas de montagem do MATER, são ligadas por duas vias (BUS) designados por via de E/S e via-MATER, como mostrado na fig. 4.5. A via de E/S é apenas uma extensão das ligações internas do hardware de cada uPi com a sua placa de E/S, compõe-se das linhas de endereço, dados e controle necessárias a ligação das duas placas.

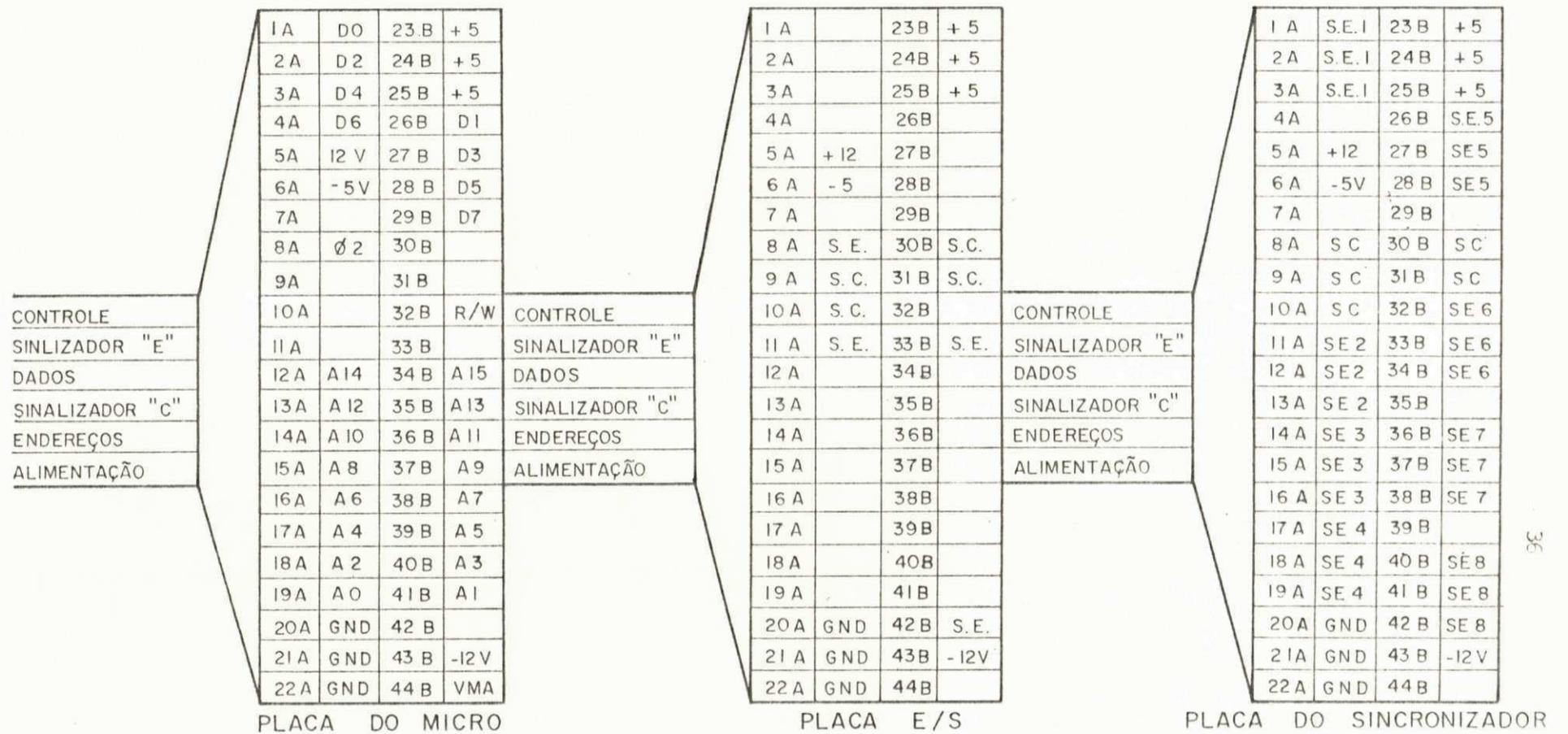
Na via-MATER, temos as ligações de cada uPi com o uPS e a M.C., realizada da forma mostrada na fig. 4.5, satisfazendo dois pontos principais:

1 - As ligações dos uPi com a M.C.

Estas ligações constituem-se das linhas de endereço, dados e controle necessárias para tal fim, estas linhas partem de cada placa do uPi e alcançam a M.C. com a disposição mostrada na fig. 4.5, sendo todas estas ligações comuns as placas dos uPi (I, II, III) e a placa da M.C.

2 - As ligações dos uPi com o uPS

A comunicação de cada uPi com o uPS é realizada através da PIA-S, localizada na placa de E/S de cada uPi, esta comunicação é realizada através das linhas de sinalização comum (S.C.) aos três uPi e ao uPS, e as linhas de sinalização específica (S.E.) de cada uPi para o uPS.



LEGENDA:

S.C. - SINALIZAÇÃO COMUM
 S.E. - SINALIZAÇÃO ESPECÍFICA

Fig. - 4.5 - BUS DO MATER.(VIA MATER)

Nesta via, apenas as linhas de alimentação são comuns a todas as placas do MATER, e apesar do prototipo compor-se de apenas três uPi o hardware de via-MATER, suporta sem modificações até oito uPi.

4.1.2 - MICROPROCESSADOR SINCRONIZADOR (uPS)

A fig. 4.6 mostra em diagrama de blocos a arquitetura utilizada para o microprocessador sincronizador (uPS). Compõe-se de um microprocessador 6800 [17], uma ACIA 6850 [17] para comunicação com o usuário através de um terminal de vídeo/teclado, 2K-bytes de memória EPROM 2708 [16] para armazenamento do sistema de desenvolvimento e do sistema operacional, 1K-byte de RAM 2114 [15] disponível para desenvolvimento de novos algoritmos de controle, 256 bytes de RAM 2112 [15] para armazenamento temporário dos dados usados pelo sistema de desenvolvimento e para armazenamento de dados estatísticos, coletados pelo uPS. Duas PIAs 6820 [17] para comunicação com os uPi e um relógio de tempo real (Real Time Clock) implementado com o LM 555 [21] que gera interrupções de NMI a cada 200 milisegundos. Na fig. 4.7 é mostrado o mapa de memória do uPS [13].

O circuito de clock é implementado com dois monoestáveis, 74123 [18] de modo a gerar dois sinais de forma de onda quadrada com frequência de 1MHz defasados de 180 graus, que constituem os clocks ϕ_1 e ϕ_2 necessários ao funcionamento do microprocessador. O circuito de reset é um latch implementado com duas portas NAND 7400 [18] de modo a eliminar o ruído da chave reset. Os clocks de transmissão e de recepção da ACIA é gerado por um circuito que usa portas inversoras CMOS 4049 [19]. No apêndice C, apresentamos o diagrama de ligações de uPS [13].

As ligações físicas dos uPi com o uPS é analisada com detalhes nos itens 3.2 e 3.3 de [13], páginas 23 a 27, é o mesmo processo descrito anteriormente no item 4.1.1, o diagrama destas ligações é apresentado no apêndice C.

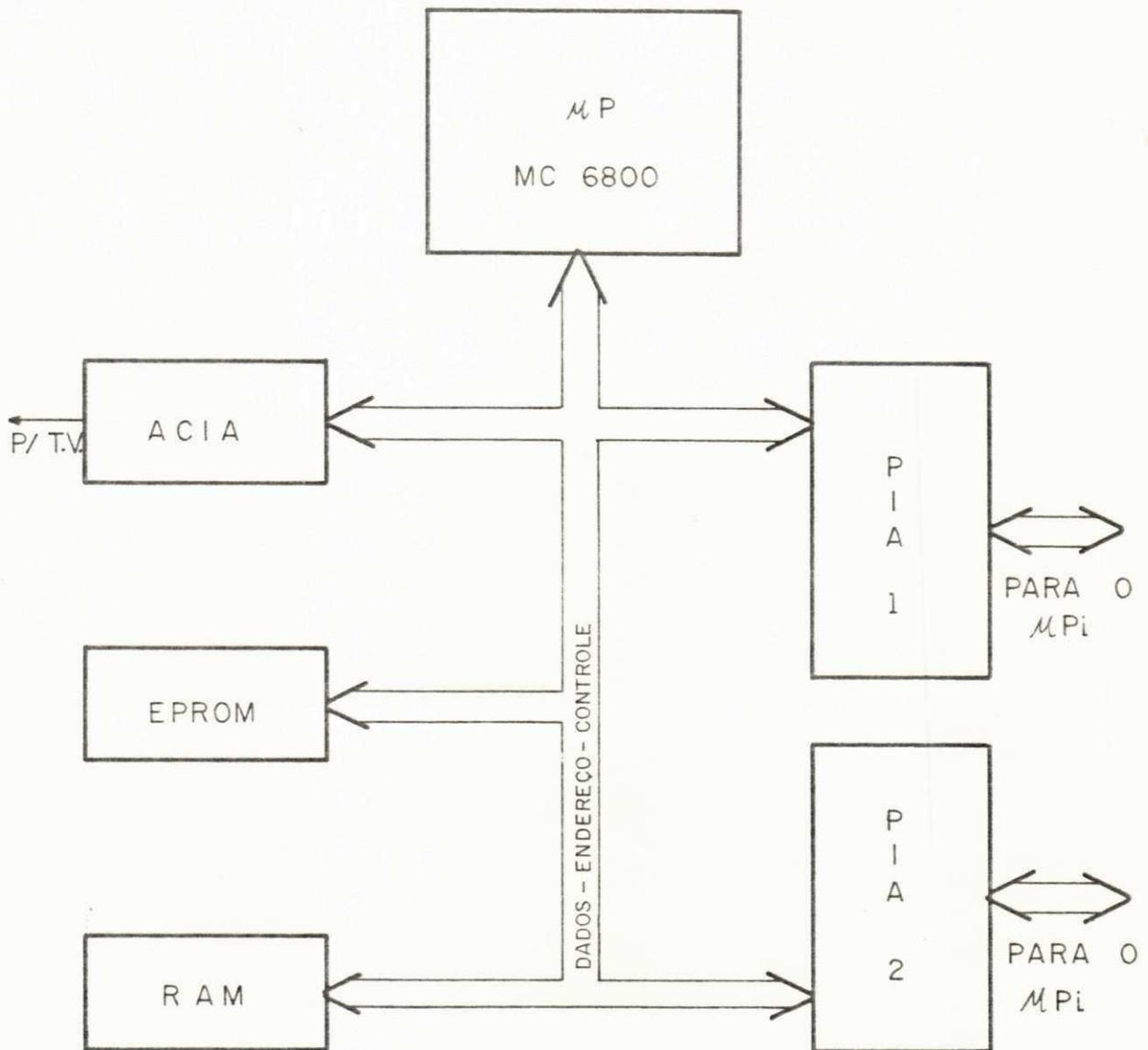


Fig. 4.6 - Arquitetura do sincronizador (μPs)

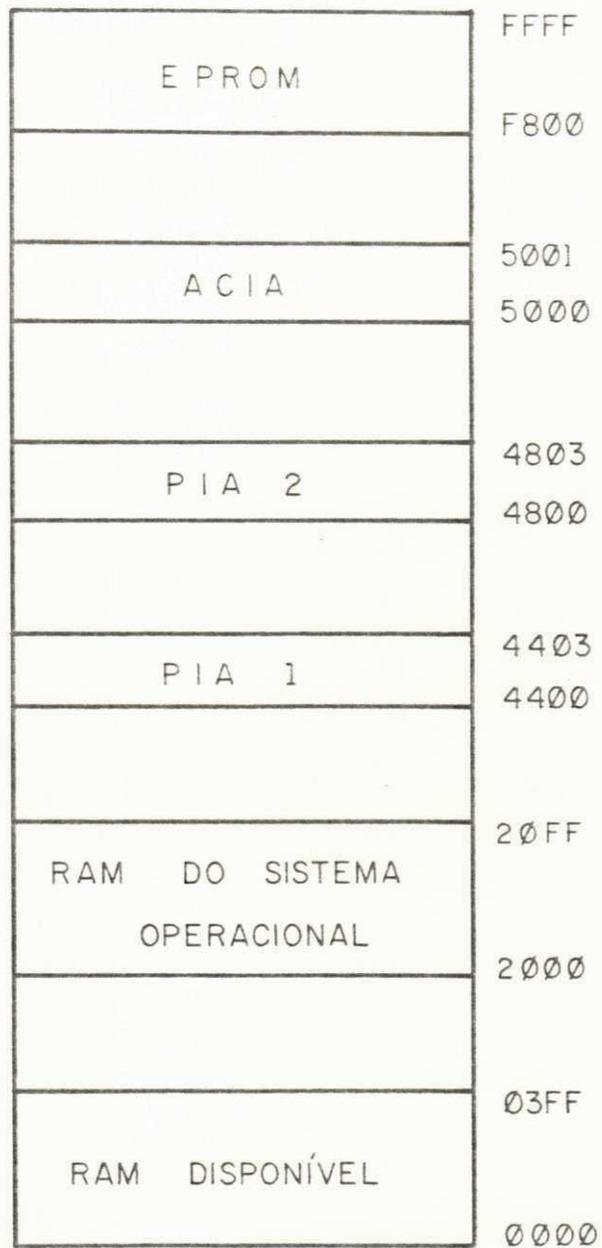


Fig. 4.7 - Mapa de memória do μ Ps.

4.1.3 - MEMÓRIA COMPARTILHADA (M.C.)

A memória compartilhada esta localizada no mesmo endereço para todos os uPi do MATER, isto é, de 4000 a 7FFF (fig. 3.4). A fig. 4.8, apresenta o mapa de memória da M.C. Compõe-se esta memória de 16K-bytes, sendo 21K-bytes de EPROM 2708 [16] endereçáveis de 4000 a 47FF, no qual esta gravado o programa de gerencia da M.C. (GMC-MATER), e de 14K-bytes de RAM 2114 [15] endereçáveis de 4800 a 7FFF, usado para a comunicação entre os uPi.

A codificação destes endereços é realizada pelo multiplexador 74154 [18], localizado na própria placa da M.C. A via de acesso tem a disposição mostrada na fig. 4.5 e no apendice C, apresentamos o diagrama de ligações da M.C.

4.2 - CONSIDERAÇÕES DE SOFTWARE

O sistema operacional do MATER, além dos programas de controle de processos, programas de gerência de memória auxiliar, e tc; compor-se-a de mais dois programas gravados em EPROM. Estes programas são: o SATER.1 (Sistema de Desenvolvimento de Programas de Aplicações em Tempo Real) e o GMC-MATER (Programa de Gerência de Memória Compartilhada do MATER).

O software do uPS é apresentado por Silva [13], com detalhes de implementação, trata-se na realidade de uma versão do SOFTR [22].

4.2.1 - MICROPROCESSADOR I (uPi)

O SATER.1 é uma versão do SOFTR [22], usado como apoio no desenvolvimento de software de aplicações em tempo real em microprocessadores, ocupa 1K-byte de memória EPROM e 256 bytes de memória RAM, endereçáveis de FC00 a FFFF e 0F00 a 0FFF, respectivamente, em cada uPi do MATER.

Apesar de ocupar as mesmas locações de memória em cada uPi do MATER, o SATER.1 apresenta pequenas diferenças de um uPi

ENDEREÇO	DESCRIÇÃO
4000 o 47FF	02 - K Bytes EPROM. (GMC - MATER).
4800 o 497F	RAM do GMC-MATER.
4980 o 49BF	Apontadores dos HEADER.
49C0 o 49FF	Apontadores de mensagens curtas.
4A00 o 4BFF	Páginas dos HEADER.
4C00 o 4DFF	Páginas das mensagens curtas.
4E00 o 4FFF	Apontadores das mensagens longas.
5000 o 7FFF	Páginas das mensagens longas.

Fig. 4:8 - Mapa de memória da MC.

para outro uPi, dependendo das unidades de E/S de cada uPi. O SATER.1 foi desenvolvido com os seguintes objetivos:

- 1 - Facilitar o desenvolvimento do software final do MATER.
- 2 - Facilitar o manuseamento dos vetores de interrupção de cada uPi.
- 3 - Permitir o uso da TTY ou terminal de vídeo/teclado pelo usuário.
- 4 - Permitir o uso das subrotinas do SATER.1 pelos programas de aplicação do usuário.

As figs. 4.9 e 4.10, mostram os comandos do SATER.1 em forma de fluxograma. Quando o uPi é resetado, o controle deste é transferido para o SATER.1. Inicialmente uma parte deste programa é executada e as chaves de acesso a M.C. são abertas (M.C. não faz parte do uPi), e a seguinte mensagem é impressa (TTY ou terminal de vídeo/teclado), por exemplo no uPi (II):

```
MATER-SATER.1
MICRO DOIS OK.
*
```

Após a impressão do asterisco o SATER.1 fica esperando que o usuário tecle um dos comandos permitidos e passará a executar a subrotina chamada. (fig.4.10) Durante o funcionamento do uPi o SATER.1 poderá responder aos três tipos de interrupção do M 6800, tendo o usuário controle sobre a localização destes vetores.

Os comandos do SATER.1 são usados para a comunicação interativa entre o usuário e o MATER, estes comandos são descritos resumidamente a seguir.

- 1 - Comando M (Subrotina TROCA), com este comando pode-se examinar/modificar o conteúdo de uma determinada localização de memória. A troca de conteúdo é permitida apenas nas localizações de RAM.
- 2 - Comando G (Subrotina DESVIO), é utilizado para desviar para o programa do usuário, passando o micropro-

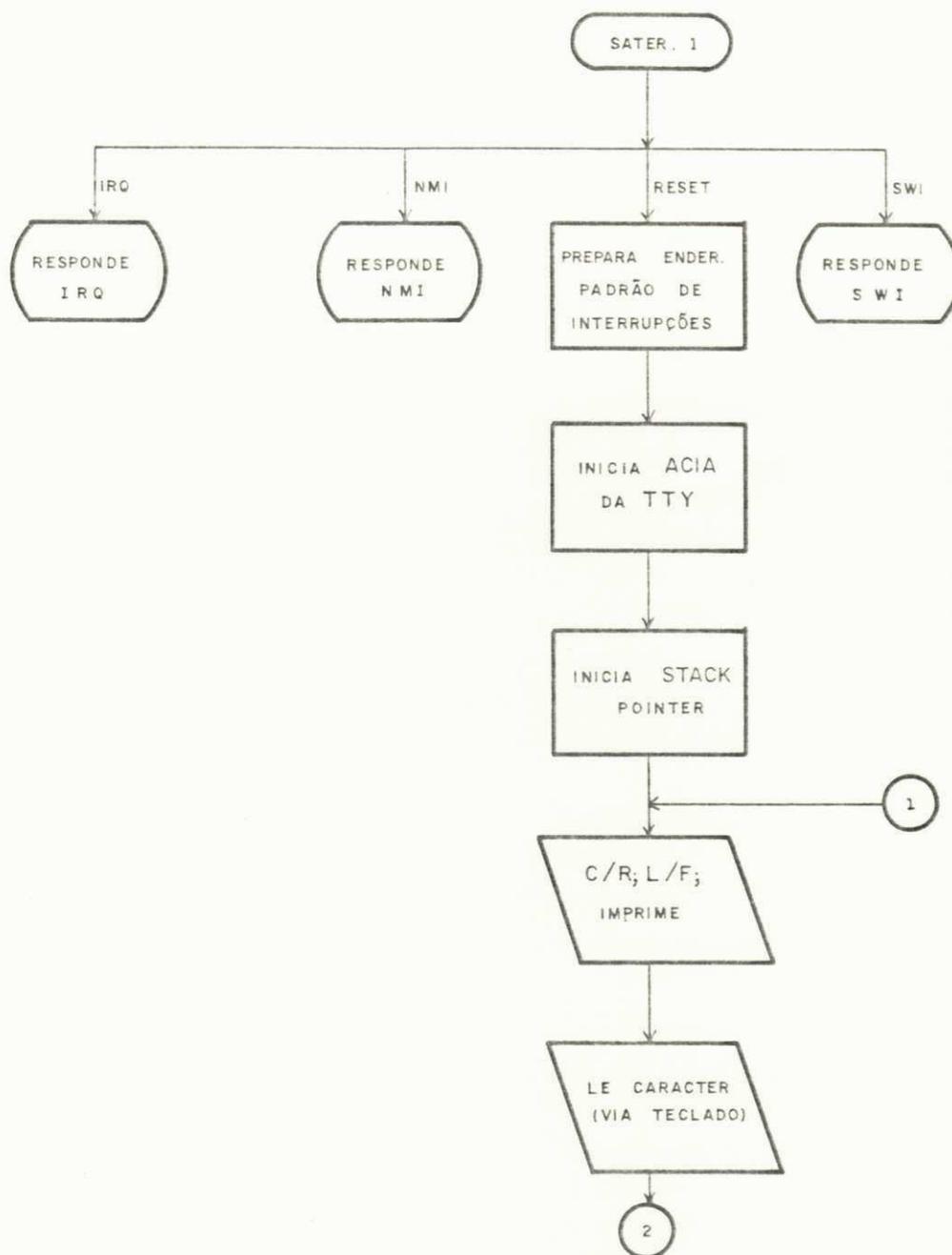


Fig. 4:9 - Fluxograma do sater. 1.
(Inicialização)

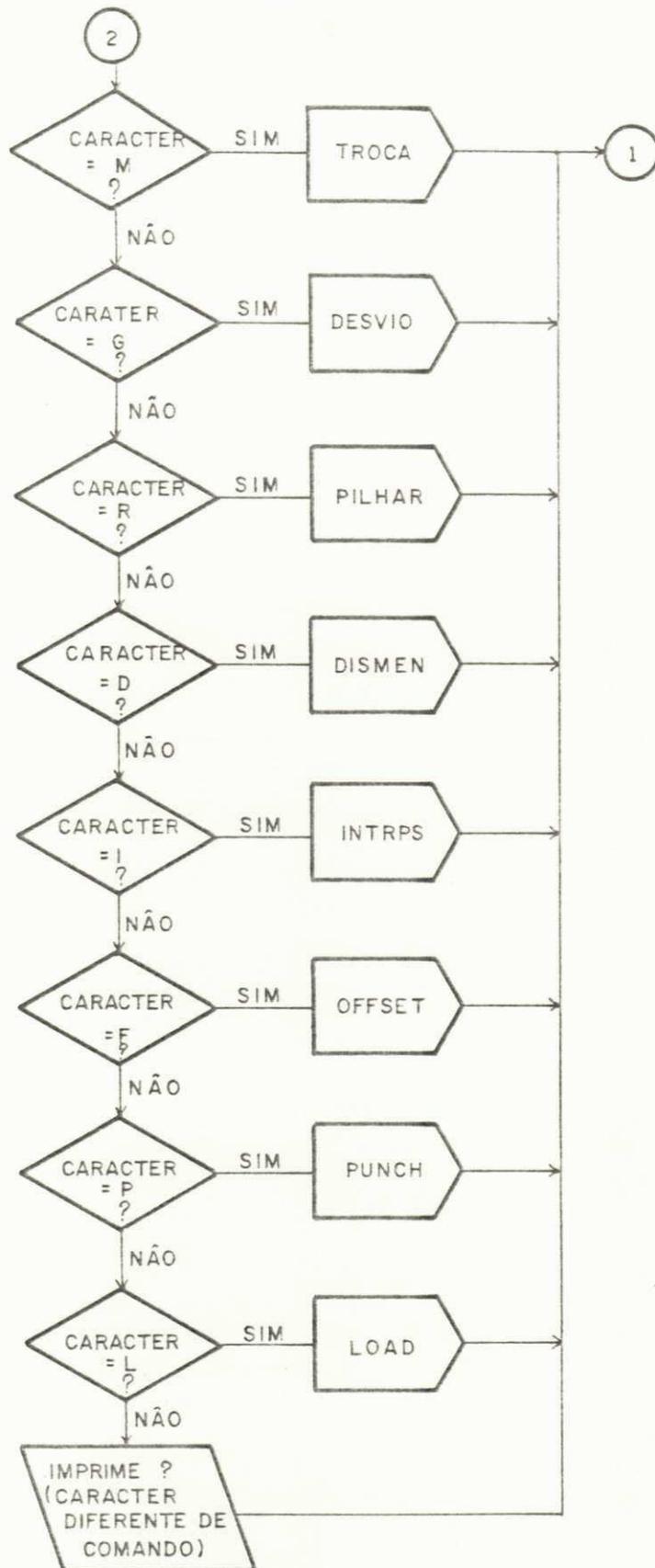


Fig. 4.10 - Fluxograma do Sater.1 (Subrotinas).

- cessador a executar este programa, uma instrução de SWI faz com que o comando volte para o SATER.1.
- 3 - Comando R (Subrotina PILHAR), este comando mostra o conteúdo dos registradores da CPU, na seguinte ordem: Registrador de Código de Condição (C), Acumulador A (A), Acumulador B (B), Registrador de Indexação (X), Contador de Programa (P), e Apontador de Pilha (S).
 - 4 - Comando D (Subrotina DISMEN), permite a impressão (display) do conteúdo de locações indicadas pelo usuário. Este conteúdo é impresso em blocos de 16 posições consecutivas (32 algarismos hexadecimais) seguidos dos seus caracteres equivalentes.
 - 5 - Comando I (Subrotina INTRPS), este comando permite a verificação e/ou modificação das posições dos vetores de interrupções dos uPi do MATER.
 - 6 - Comando F (Subrotina OFFSET), permite o cálculo de deslocamentos para desvios.
 - 7 - Comando P (Subrotina PUNCH), permite que o conteúdo das locações de memória solicitadas pelo usuário sejam impressos em fita de papel, através de uma impressora TTY.
 - 8 - Comando L (Subrotina LOAD), permite carregar a memória particular (RAM) com dados perfurados em fita de papel através de uma leitora de fita.

4.2.2 - MICROPROCESSADOR SINCRONIZADOR (uPS)

O sistema operacional do uPS [13] baseia-se em um programa chamado KERNEL [14] que sincroniza e aciona as diferentes tarefas do uPS, conforme o tipo de interrupção e da sua procedência. O KERNEL aceita os seguintes tipos de interrupções:

- a. IRQ que pode ser acionada por entrada de caracteres via teclado, saída de caracteres via terminal de vídeo/teclado e interrupções de hardware externas, devido ao processo sobre controle [13] .

b. NMI que é acionada por um relógio de tempo real (Real Time Clock) a cada 200 milisegundos [13].

Os tipos de tarefas aceitas pelo sistema operacional do uPS, os tipos de mensagens solicitadas pelo operador e as solicitadas pelo algoritmo de controle são apresentadas e analisadas por SILVA [13]. O KERNEL do uPS foi desenvolvido com o auxílio do SATER.1 [13] que é uma versão do SOFTR [22] específica para o uPS.

4.2.3 - MEMÓRIA COMPARTILHADA (M:C.)

Uma das características do MATER é a transmissão de informações entre os uPi . Baseado nestas características, desenvolvemos um banco de memória para ser compartilhada por todos os uPi.

O M6800 não possui instruções para transferir os conteúdos de blocos de memória de uma posição para outra (instrução tipo MOVE). Para esta relocação de dados, necessita-se do desenvolvimento de uma subrotina que requer diversos de máquinas para relocar um único byte (aproximadamente 40 us por byte relocado). Baseado nesta restrição a transferência de dados do uPi para a M. C. e vice-versa, terá um tempo elevado, conseqüentemente a comunicação entre os uPi será bastante lenta (gastará um tempo elevado).

Devido a característica do MATER em que a frequência de comunicação entre os uPi foi projetada para ser bastante baixa (isto é, o sistema operacional sendo do tipo mestre/escravo e que cada uPi possui sua memória particular e seus dispositivos de E/S, a transmissão de dados deverá ser feita, principalmente, entre o uPi mestre e os demais), o tempo de comunicação entre os uPi não afetará a característica para aplicações em tempo real dos uPi.

No projeto da M.C. três parâmetros principais considerados

- 1 - Integridade dos dados armazenados na M.C.
- 2 - Velocidade de transmissão entre a M.C. e a memória par

ticular de cada uPi.

- 3 · Espaço de memória gasto na gerência da M.C. (GMC-MATER).

Como descrito anteriormente, projetamos a M.C. como um bloco de 16K-bytes, 14K-bytes de memória RAM e 2K-bytes de memória EPROM para armazenar o GMC-MATER. Com esta configuração, o GMC-MATER será compartilhado por todos os uPi do MATER, isto resolve o parâmetro três.

Os dados serão armazenados na M.C. usando-se páginas encadeadas, o que garante o segundo parâmetro, o problema final a se resolver foi a integridade dos dados armazenados na M.C. parâmetro um, esta integridade é garantida como descrito a seguir.

Durante o acesso a M.C. por um uPi, além das considerações anteriores, existe a organização dos dados, tabelas, comunicação com outros uPi, preparação de estatísticas do uso da M.C., criação dos apontadores de mensagens e páginas de dados, etc. esta organização é realizada pelo uPi designado como mestre, em caso contrário nenhum outro uPi terá acesso a M.C.

Considerando-se que quando o uPi estiver executando o programa armazenado na EPROM da M.C., estará sob o controle da M.C., podemos supor que exista um outro microprocessador na M.C. sob este ponto de vista o uPi poderá considerar que está se comunicando com um microprocessador na M.C. Este microprocessador da M.C. existente sob o ponto de vista de programação, chamou-se microprocessador VIRTUAL (uPV).

A partir desta concepção de um uPV na M.C., tornou-se possível garantir a integridade dos dados armazenados na M.C. A comunicação aparentemente entre dois microprocessadores, tornou-se relativamente simples o desenvolvimento do GMC-MATER, e resolve o parâmetro um.

Para a comunicação entre os uPi, desenvolveu-se um protocolo especial designado por HEADER. O HEADER consiste em uma mensagem inicial de 16 bytes, nos quais estão incluídos todas as informações necessárias e suficientes para a comunicação en-

tre os uPi e o uPV.

Os 16 bytes do HEADER, tem a seguinte estrutura; organizada pelo sistema operacional do MATER.

byte 0 - Indica o uPi transmissor da mensagem.

Este byte indicará o número do uPi transmissor, isto é, o uPi no qual se origina a comunicação.

byte 1 - Indicará o uPi receptor da mensagem.

Este byte indicará o número do uPi receptor, isto é, o uPi para o qual se destina a mensagem a ser transmitida pelo uPi transmissor.

byte 2 - Indica se o uPi transmite ou recebe a mensagem.

Quando o conteúdo do byte for igual a 54, isto significará que o uPi (byte 0) transmite para outro uPi (byte 1). Quando o conteúdo do byte 2 for igual a 52, isto indicará que o uPi (byte 0) recebe mensagem enviada por outro uPi (byte 1).

bytes 3 e 4 - Indicam o endereço da volta.

Nestes bytes, o sistema operacional de cada uPi, guardará, o endereço da locação de memória, para o qual o programa corrente será desviado, após o término da comunicação entre o uPi e o uPV.

byte 5 - Indica se tem mensagem curta.

Chama-se de mensagem curta, a uma mensagem utilizada para a troca de informações entre dois uPi, esta mensagem é composta, de 16 bytes e pode ou não estar presente quando um uPi transmite ou recebe um HEADER. Quando o conteúdo deste byte é 43, isto indicará a transmissão ou recepção de uma mensagem curta, caso contrário não existirá a mensagem curta. Este tipo de mensagem será de grande utilidade, quando um dos uPi necessitar sinalizar um outro uPi pois o tempo gasto para a transmissão e/ou recepção desta mensagem será relativamente pequeno quando comparado ao tempo de transmissão e/ou recepção de mensagem longa, descrita

no byte 9.

byte 6 - Indica o número de HEADER a serem transmitidos ou recebidos pelo uPi.

Quando iniciada a comunicação entre os uPi, poderá existir situações nas quais mais de uma mensagem para uPi distintos, deverá ser transmitida ou recebida pelo uPi (transmissor ou receptor). Tendo-se deste modo a necessidade de um parâmetro que indique o mínimo de HEADER a serem transmitidos ou recebidos durante uma comunicação.

bytes 7 e 8 - Indica o endereço da mensagem curta.

Após identificar a existência de uma mensagem curta, o uPi terá necessidade de saber o endereço desta mensagem para poder transmiti-la ou recebe-la.

byte 9 - Indica o número de páginas de mensagem longa.

Chama-se de mensagem longa, a uma mensagem que utiliza a troca de informações entre dois uPi, quando a quantidade de informações é relativamente grande, quando comparada com o número de bytes de uma mensagem curta. Esta mensagem será utilizada pelos uPi para a troca de dados e tarefas, é composta de páginas encadeadas de 30 bytes cada, sendo os dois últimos bytes para apontarem o endereço da página seguinte. Quando o conteúdo deste byte é igual a $\emptyset\emptyset$, isto indicará que não existe mensagem longa para ser transmitida ou recebida. Caso contrário indicará o número de páginas que compõem a mensagem longa.

bytes A e B - Indicam o endereço da primeira página da mensagem longa.

Após indicar a existência de uma mensagem longa, o uPi terá necessidade de conhecer o endereço desta mensagem para poder transmiti-la ou recebe-la.

bytes C,D,E, e F - Sem utilização no estágio atual do HEADER.

Os bytes C,D,E, e F estão reservados para futuras utili-

zações, dependendo das necessidades ou não de ampliações do HEADER.

Antes de iniciar a comunicação, o sistema operacional do uPi, comporá o HEADER como descrito anteriormente. Todavia, se o uPi necessita fazer uma verificação na M.C., para determinar se tem alguma mensagem para si, apenas os bytes 1 e 2 do HEADER serão necessários, pois o GMC-MATER tem condições de localizar todas as mensagens para o uPi, com base nestes dois bytes.

4.2.4 - PROGRAMA DE GERÊNCIA DA MEMÓRIA COMPARTILHADA - GMC-MATER

O GMC-MATER foi desenvolvido com o auxílio do SATER.1 no próprio uPi do MATER, ocupa 1K-byte de ERROM e uma área de 17 F bytes de RAM como rascunho. As figuras 4.11 e 4.14, mostram em forma de fluxograma as quatro principais subrotinas do GMC-MATER e no apêndice D, tem-se o GMC-MATER na linguagem assembly do microprocessador M6800.

SUBROTINA ACESSO

Quando um dos uPi necessita do acesso a M.C., este executa uma subrotina do GMC-MATER, designada de subrotina acesso, a figura 4.11, apresenta o fluxograma desta subrotina, que apesar de fazer parte do GMC-MATER, esta gravada em EPROM na M.P. de cada uPi, isto evitará que os uPi utilizem a M.C., sem que esta esteja organizada. Três pontos fundamentais constituem esta subrotina, são eles:

1. A solicitação do acesso a M.C. ao uPs.
2. Teste de organização da M.C.
3. Solicitação do uPV, para organizar a M.C. quando esta não está organizada.

A solicitação do acesso a M.C. já foi descrito no item 3.2.2; o teste de organização da M.C., consiste na verificação, das tabelas, apontadores e contadores dos HEADER, mensagem CURTA ou LONGA, descritas na subrotina ORGANIZA.

SUBROTINA ORGANIZA

Esta subrotina será executada, toda vez que o acesso do



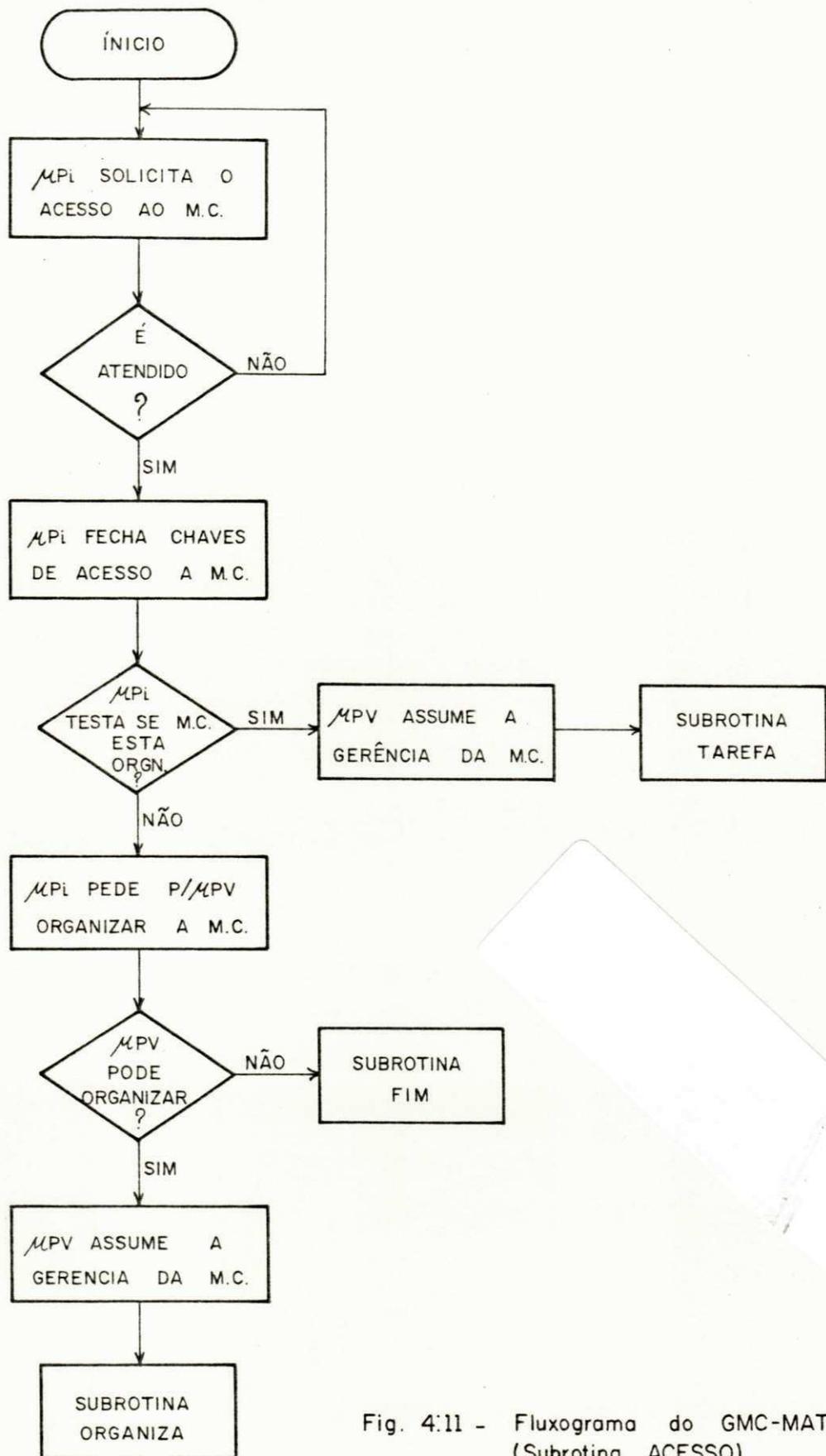


Fig. 4:11 - Fluxograma do GMC-MATER. (Subrotina ACESSO).

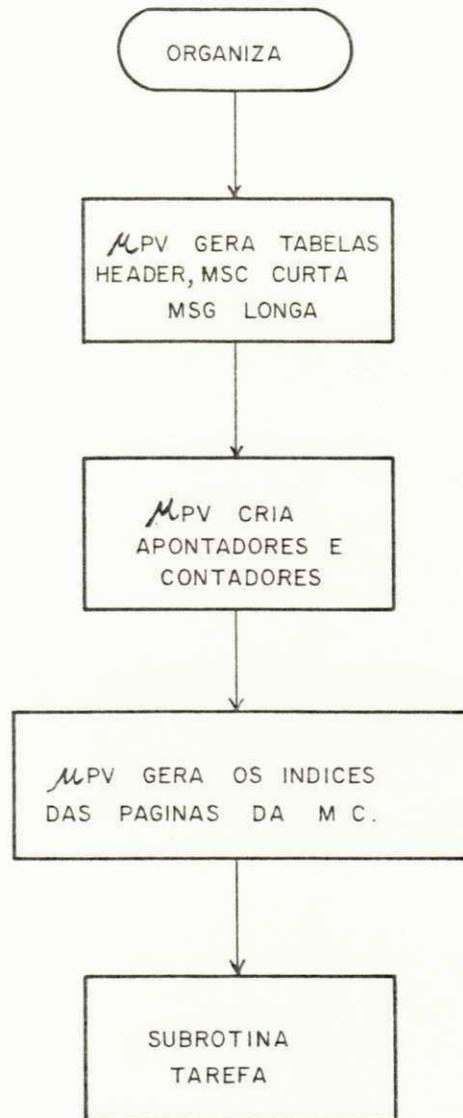


Fig. 4:12 - Fluxograma do GMC-MATER.
(Subrotina ORGANIZA)

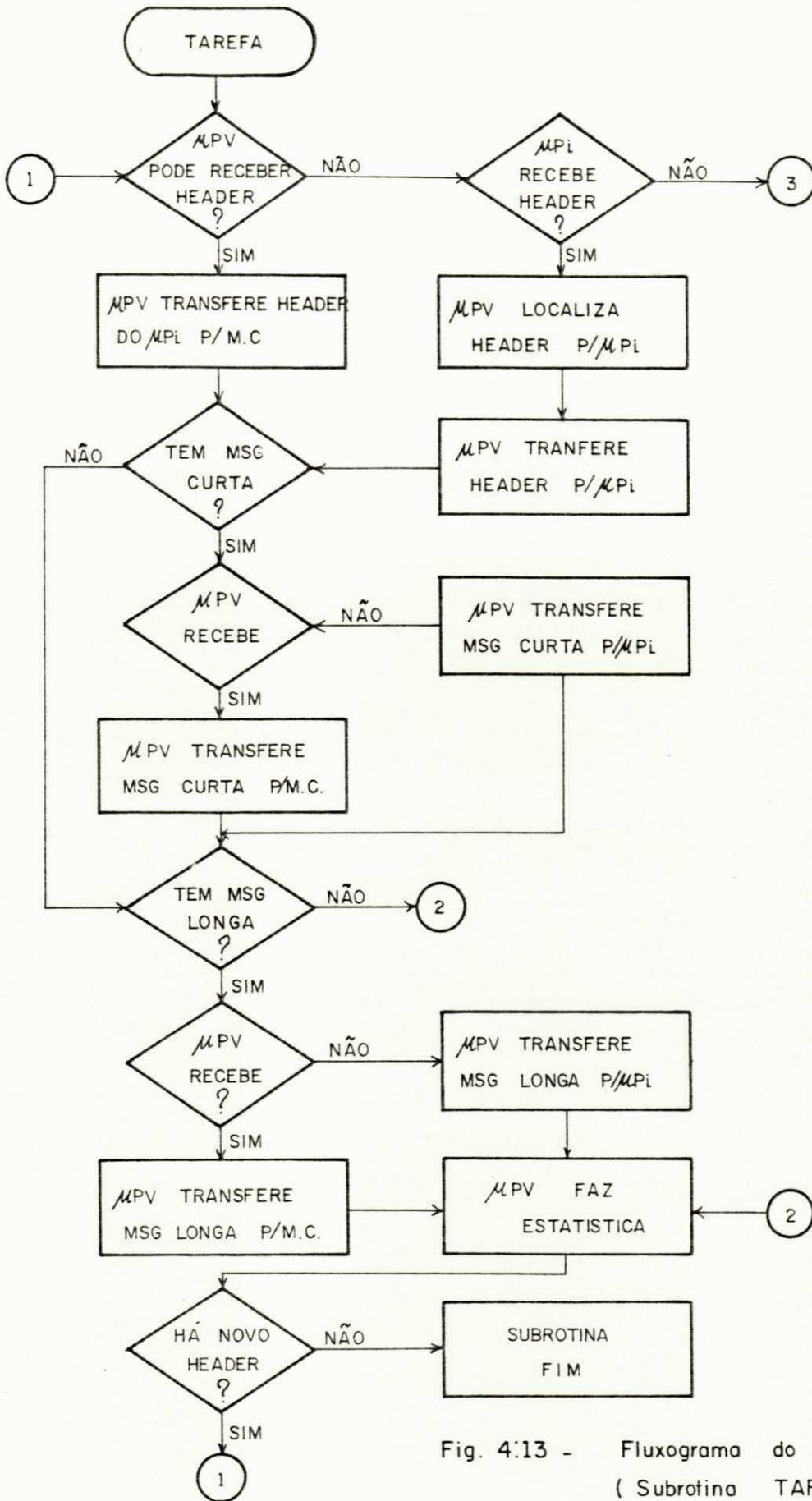


Fig. 4:13 - Fluxograma do GMC-MATER. (Subrotina TAREFA).

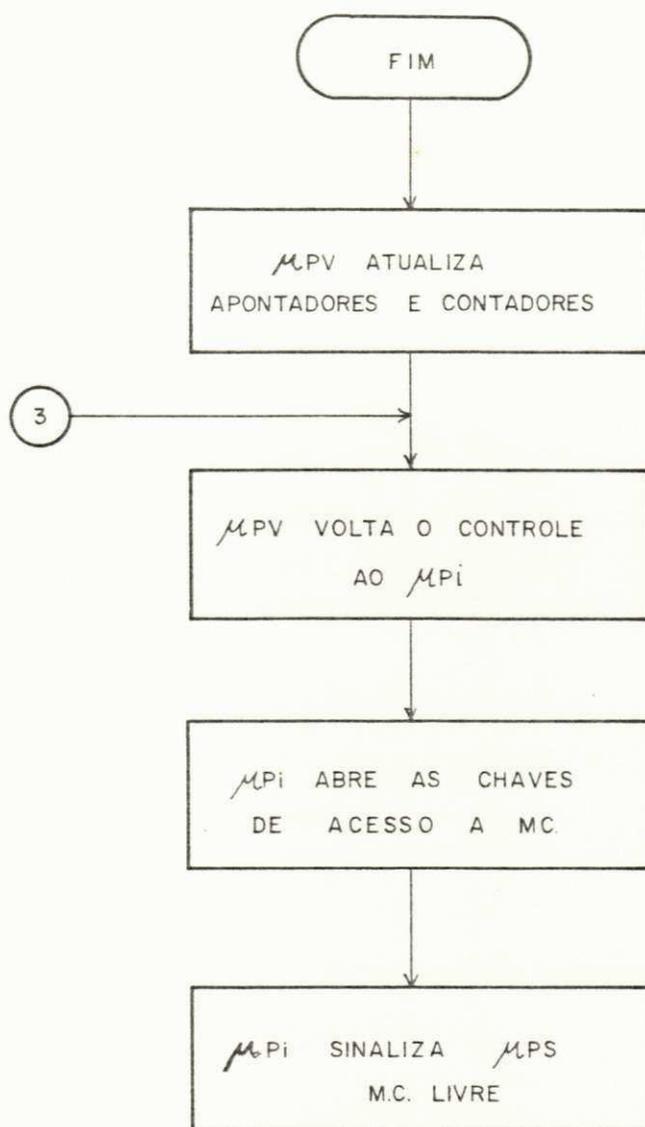


Fig. 4.14 - Fluxograma do GMC-MATER.
(Subrotina FIM).

uPi a M.C. for permitido, considerando-se dois requisitos fundamentais:

1. A memória compartilhada não está organizada.
2. O uPV identifica o uPi como sendo o uPM.

Chama-se de uPM (Microprocessador Mestre) ao uPi do MASTER, que entre outras tarefas, tem condições de organizar a M.C. apenas o uPM organiza a M.C.

A subrotina ORGANIZA consiste nas seguintes etapas:

- a) Geração das tabelas: do HEADER; da mensagem CURTA e da mensagem LONGA.
- b) Geração dos apontadores e contadores: do HEADER, da mensagem CURTA e mensagem LONGA.
- c) Geração dos índices das páginas da M.C.

Durante a execução desta subrotina, são gerados as tabelas, apontadores e contadores do HEADER, um total de $(1F)_{16}$ HEADER, mensagem CURTA, um total de $(1F)_{16}$ mensagem LONGA, um total de $(1F)_{16}$ mensagem longa e páginas da M.C., um total de $(FF)_{16}$ páginas. A figura 4.8 mostra as áreas reservadas para todos estes elementos.

SUBROTINA TAREFA

Após serem satisfeitos os requisitos das subrotinas ACesso e ORGANIZA, três pontos são considerados para a execução da subrotina TAREFA.

1. a identificação do tipo de mensagem
2. o limite de espaço disponível na M.C.
3. a execução desta tarefa.

A identificação do tipo de mensagem consiste basicamente em se determinar se o uPi transmite mensagem para o uPV, o que depende do limite de espaço disponível na M.C., ou se uPi recebe mensagem do uPV, esta etapa é independente da existência de espaço disponível na M.C.. Definidos estes pontos a subrotina TAREFA é executada e a troca de informações entre os uPi e o uPV é realizada, neste caso realiza-se para cada uPi uma estatística do

número, tipo e tamanho (nº de páginas) recebidas ou transmitidas durante a execução desta. Se o limite de espaço não é suficiente para que o uPV receba as mensagens do uPi, a subrotina FIM passará a ser executada.

SUBROTINA FIM

Esta subrotina é constituída de dois pontos principais, a atualização dos apontadores e contadores, toda vez que a subrotina TAREFA executa a troca de informação e a volta de controle para o uPi, isto é realizado após a execução da subrotina TAREFA, ou quando o limite de espaço na M.C. não é suficiente para a troca de informações entre o uPi e o uPV.

Constitui-se deste modo a subrotina FIM as duas etapas bem definidas:

- 1) a atualização dos apontadores e contadores,
- 2) a volta do controle para o uPi e conseqüentemente liberação da M.C.

A primeira etapa, está gravada na EPROM da M.C., enquanto a segunda etapa está gravada na EPROM da M.P. o que reduz em aproximadamente 30 microsegundos o tempo de acesso de cada uPi a M.C.

A versão atual do GMC-MATER, ainda não será a definitiva após a conclusão do sistema operacional do MATER, esta será adequada para permitir uma melhor utilização da memória compartilhada.

CAPITULO V

CONCLUSÕES

Inicialmente propomos o desenvolvimento do MATER, com as características apresentadas no capítulo três. Devido a alguns problemas durante o decorrer do trabalho (serão especificados mais adiante), o desenvolvimento do MATER foi o seguinte:

- 1 - Construimos: a caixa para conter os uPi, M.C. e uPS; a via-MATER, para permitir a comunicação entre os microprocessadores e a memória compartilhada; e a fonte de alimentação.
- 2 - Desenvolvemos: o sistema operacional dos uPS por SILVA [13]; uma versão do SOFTR [22] (o SATER.1) para os uPi e o GMC-MATER para gerência da M.C.

Na arquitetura proposta para o MATER, no capítulo três, propõe-se um sistema com oito microprocessadores. Este número está limitado apenas pelo fato do uPS ter sido projetado com duas PIAs para sinalização dos uPi. O número de microprocessadores a ser utilizado no MATER, só poderá ser determinado após a conclusão do sistema operacional, realizando-se um levantamento estatístico do tempo do acesso dos uPi a M.C. Este número dependerá, também, da aplicação específica do MATER.

No prototipo do MATER, propõe-se um sistema de três uPi, compartilhando a M.C. entretanto, apenas dois microprocessadores

(uPi e uPS) e a M.C. foram implementados, devido a falta de material no laboratório para a implementação dos demais uPi. Todavia, no apêndice C, apresentamos os diagramas de ligações do sistema com três uPi.

Durante a montagem do protótipo observamos a limitação da via-MATER, devido ao tipo de placa de montagem empregada. Sugerimos que se utilize uma placa de montagem com maior número de pinos de saída, o BUS S-100 por exemplo, e de maiores dimensões físicas, o que permitirá uma menor densidade de circuitos por placa de montagem, facilitando deste modo a realização do hardware.

O protótipo contém uma única fonte de alimentação para todas as placas de montagem. Isto criou uma pequena inconveniência. Toda vez que se necessita modificar o hardware em uma das placas, todo o sistema tem que ser desligado. Sugere-se que cada uPi tenha seu próprio regulador de tensão independente, isto facilitará a remoção das placas de montagem, sem a necessidade de se desativar todo o sistema e conseqüentemente parada do controle dos processos associados ao sistema.

Como apresentado no item 4.2.3, devido a baixa velocidade de relocação de dados (com o M6800), o sistema operacional do MATER, deve ser do tipo mestre/escravo, limitando deste modo a frequência de transferência de informações entre os uPi. Sugere-se o uso de um microprocessador que tenha uma instrução do tipo MOVE, facilitando deste modo a troca de informações entre os uPi com uma conseqüente redução no tempo de utilização da M.C. pelos uPi.

O sistema operacional dos uPi está sendo desenvolvido para permitir o controle de processos em tempo real. Sente-se a necessidade de uma linguagem de alto nível, para desenvolvimento de software, específico para microprocessadores, o que facilitará a implementação de programas de controle de processos e do próprio sistema operacional do MATER.

O uso de uma linguagem específica para aplicações em tempo real, facilitará em muito a implementação de programas para

controle de processos externos. Finalmente, sugere-se que seja observado, após a conclusão do sistema operacional, o uso de um dispositivo de E/S compartilhado, e que seja observada a performance do sistema, com os uPi projetados com uma CPU mais poderosa que a M 6800 .

Apesar do fato do prototipo estar implementado com um uPi o uPS e a M.C., a comunicação entre os processadores foi simulada, com o uPi sendo designado em determinado instante como sendo o uPi (I), (II) e (III). O tempo gasto para relocação de 1-K byte de dados da M.C. para a M.P. e vice-versa, é da ordem de aproximadamente 41 milisegundos.

Vale salientar que os testes e conclusivos sobre o funcionamento do GMC-MATER, só poderão ser realizados com um número mínimo de dois ou três uPi. No caso de utilização de dois microprocessadores pode-se verificar a possibilidade do uso de DMA para acesso a M.C.

APENDICES

Estes apêndices descrevem em detalhes as informações necessárias e suficientes para o usuário operar e expandir futuramente o MATER. Constatam ainda os diagramas de ligações dos uPi e M.C. do MATER e do uPS. O GMC-MATER é apresentado na linguagem do M 6800.

APENDICE A

OPERAÇÃO DO MATER

A operação do MATER, como a de qualquer sistema de microprocessador, requer conhecimentos básicos da estrutura interna, conjunto de instruções, modos de endereçamento da CPU, afim de permitir uma maior interação entre o usuário e o sistema, além de conhecimentos do software "residente" do sistema, neste caso o SATER.1.

Os conhecimentos sobre a CPU M6800, podem ser adquiridos em [18] e [20], por exemplo, e os conhecimentos específicos sobre o MATER, podem ser adquiridos após estudos dos capítulos 3 e 4 deste trabalho.

Todavia, o principio de funcionamento de cada uPi do MATER, é relativamente simples. Após ligada a chave de alimentação, do sistema, - todo sistema deve ser energizado e/ou desligado ao mesmo tempo, pois alimentação por partes, pode causar danos ao SATER.1 e GMC-MATER, gravados em EPROM na M.P. e M.C., respectivamente - o usuário deve resetar cada uPi do MATER, e estes respondem com a seguinte mensagem:

MATER - SATER.1

MICRO (I) OK

*

onde (I=UM,DOIS), impressa no terminal associado a cada uPi (ter

minal de vídeo/teclado, TTY, etc), o microprocessador TRES, não responde com esta mensagem, devido ao fato de estar ligado apenas a processos sobre controle. Entretanto, seu funcionamento pode ser verificado pelo uPS [13].

A não impressão desta mensagem implica em problemas com a fonte de alimentação (mal contato ou baixa amperagem). Após a impressão do asterisco, o uPi do MATER esta sob o comando do SATER.1, devendo o usuário proceder de acordo com o descrito no item 4.2, afim de efetuar os trabalhos desejados.

O acesso a M.C. deve ser feito sempre sob o comando do SATER.1, encarregando-se este de sinalizar o uPS [13] e passar o comando da M.C. ao uPV, sempre que se tiver necessidade de acessar este banco de memória.

Dos 4K-bytes da memória RAM da M.P. de cada uPi, as locações de endereço de 0F00 a 0FFF, são usadas como RAM do SATER.1, a utilização desta área de memória deve ser realizada apenas quando se desejar modificar a posição dos vetores de interrupção de cada processador, isto pode ser realizado com o auxílio dos comandos do próprio SATER.1.

APENDICE B

EXPANSÃO DO MATER

Como descrito no item 4.1.1, dispõe-se em cada uPi das seguintes locações de memória: 1000 a 3FFF, 8000 a BFFF, C800 a EFFF, perfazendo um total de 38K-bytes, os quais podem ser destinados a expansão de cada uPi em particular, ou como expansão da M.C. A expansão de um determinado uPi, não implica na redução da área disponível em outro uPi, salvo se esta expansão é realizada na M.C.

Suponha-se que precisamos expandir a M.C. de um dos uPi de 4K-bytes para 8K-bytes. Podemos escolher, por exemplo, os endereços de 1000 a 1FFF, isto não implica que não possamos utilizar estas mesmas locações de memória para expansão dos demais uPi, isto é possível devido ao fato de cada uPi ser um sistema particular.

No caso de expansão da M.C., esta área expandida deve ser considerada como uma expansão particular de cada uPi, pois quando um dos uPi acessa a M.C. esta passa a ser tratada como parte deste, devemos portanto evitar o endereçamento de outro dispositivo nestas locações.

De posse destas informações, a expansão de um uPi ou da M.C. do MATER, resume-se nas seguintes etapas:

- 1 - Escolher dentre as locações disponíveis as que se de

seja utilizar. (Se a expansão for na M.C. estas locações não podem ser mais utilizadas para expansão dos uPi.)

- 2 - Codificar estes endereços com o auxílio de portas lógicas ou multiplexadores.
- 3 - Realizar o hardware necessário a esta expansão.
- 4 - Adiciona-se a nova placa ao sistema

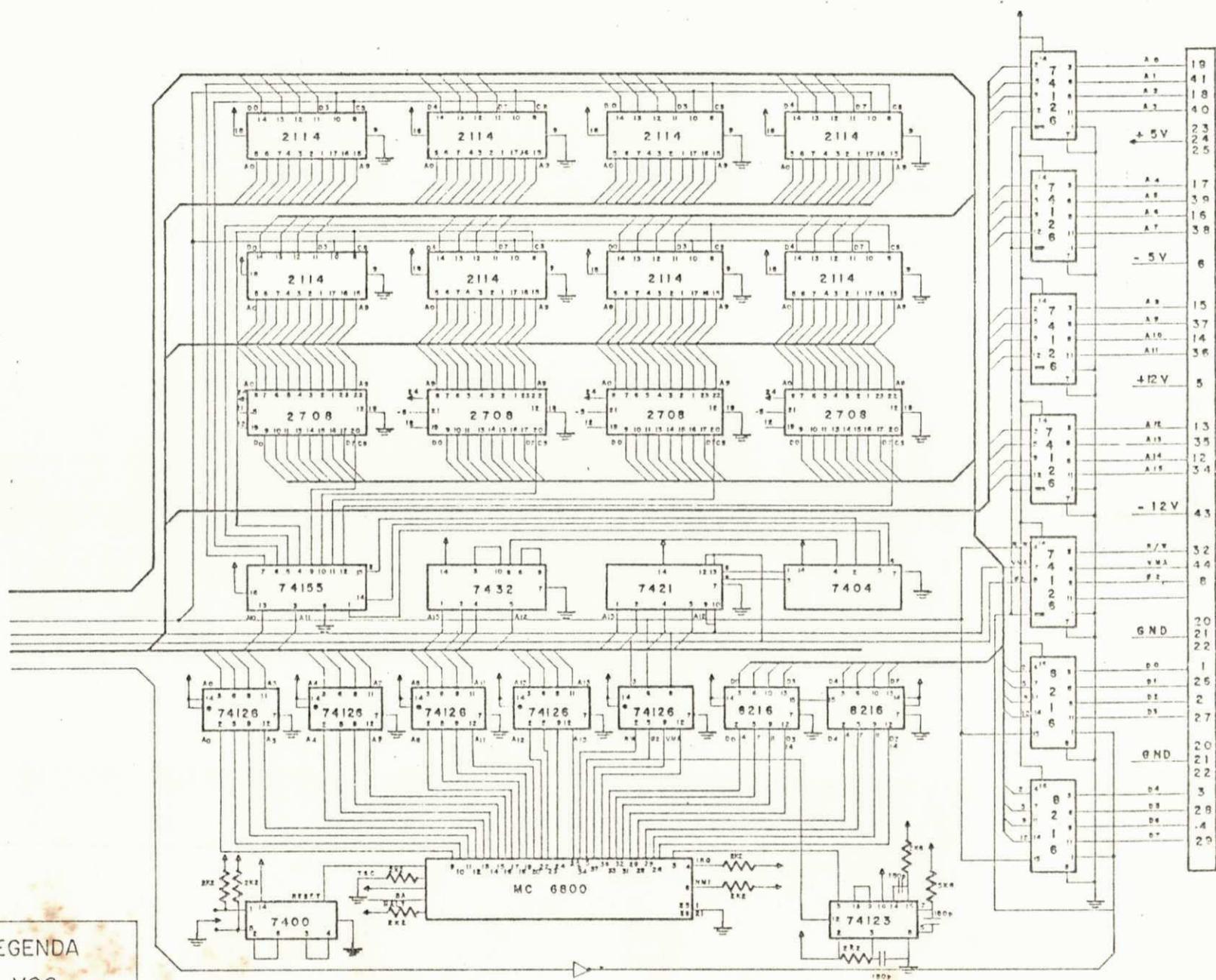
Estas mesmas etapas devem ser realizadas se desejamos adicionar uma ou mais unidades de E/S (PIA e/ou ACIA) a um uPi, sendo também necessário o software requerido por estes dispositivos. No caso de expansão de unidades de E/S, aconselha-se o uso das locações de memória assinaladas com D, na figura 3.4.

Na expansão da M.C. é também necessário que o GMC-MATER, seja modificado, afim de que possa gerenciar as novas locações de memória, caso contrário o GMC-MATER, considerará apenas a parte inicial da M.C., ignorando a expansão realizada.

APENDICE C

CIRCUITOS DO MATER





LEGENDA

- + 5V VCC
- 1.4.10.13 MICRO 1.3

DIAGRAMA DE LIGAÇÕES DOS MICROS 1-3

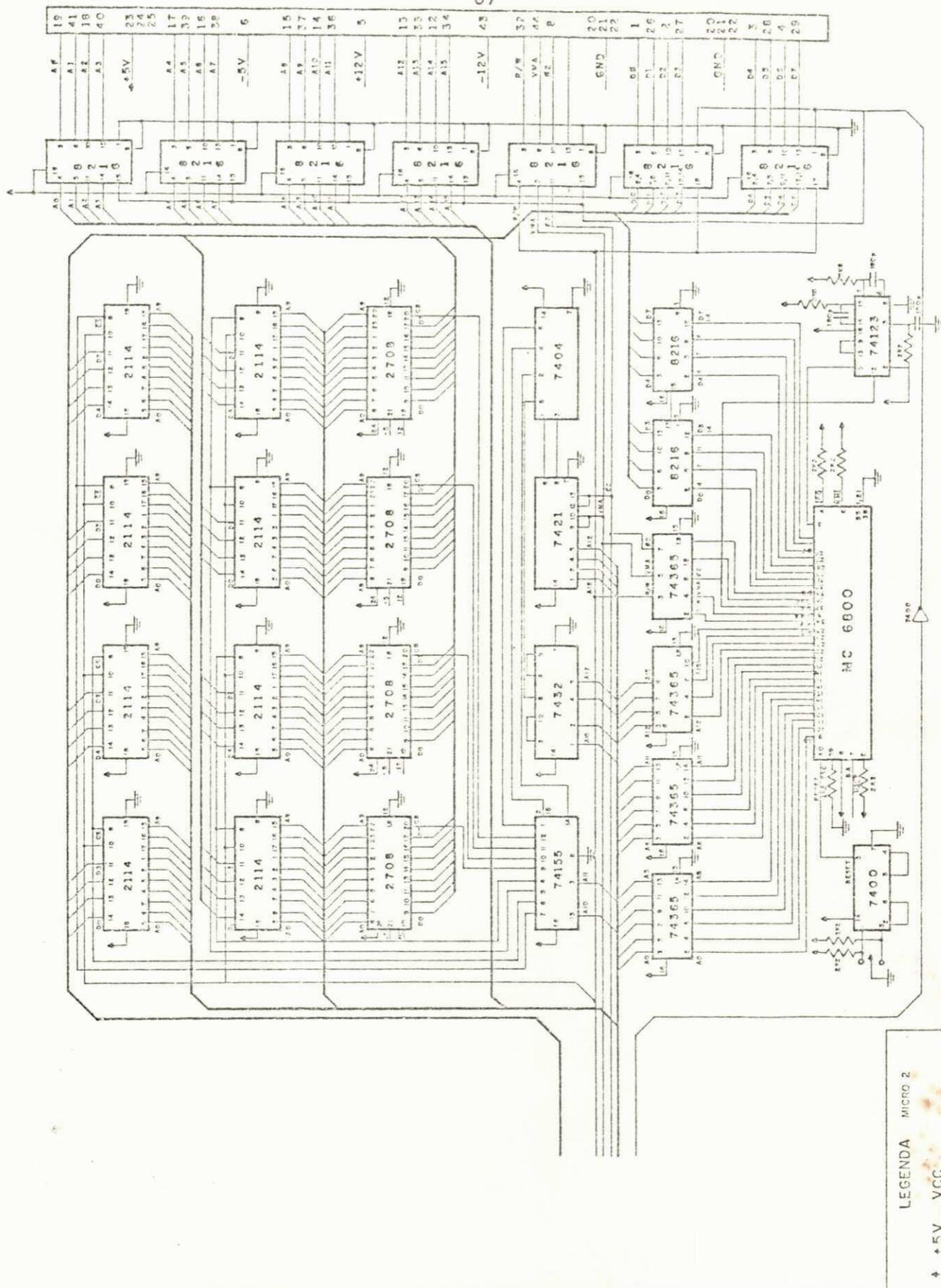


DIAGRAMA DE LIGAÇÕES DO MICRO 2

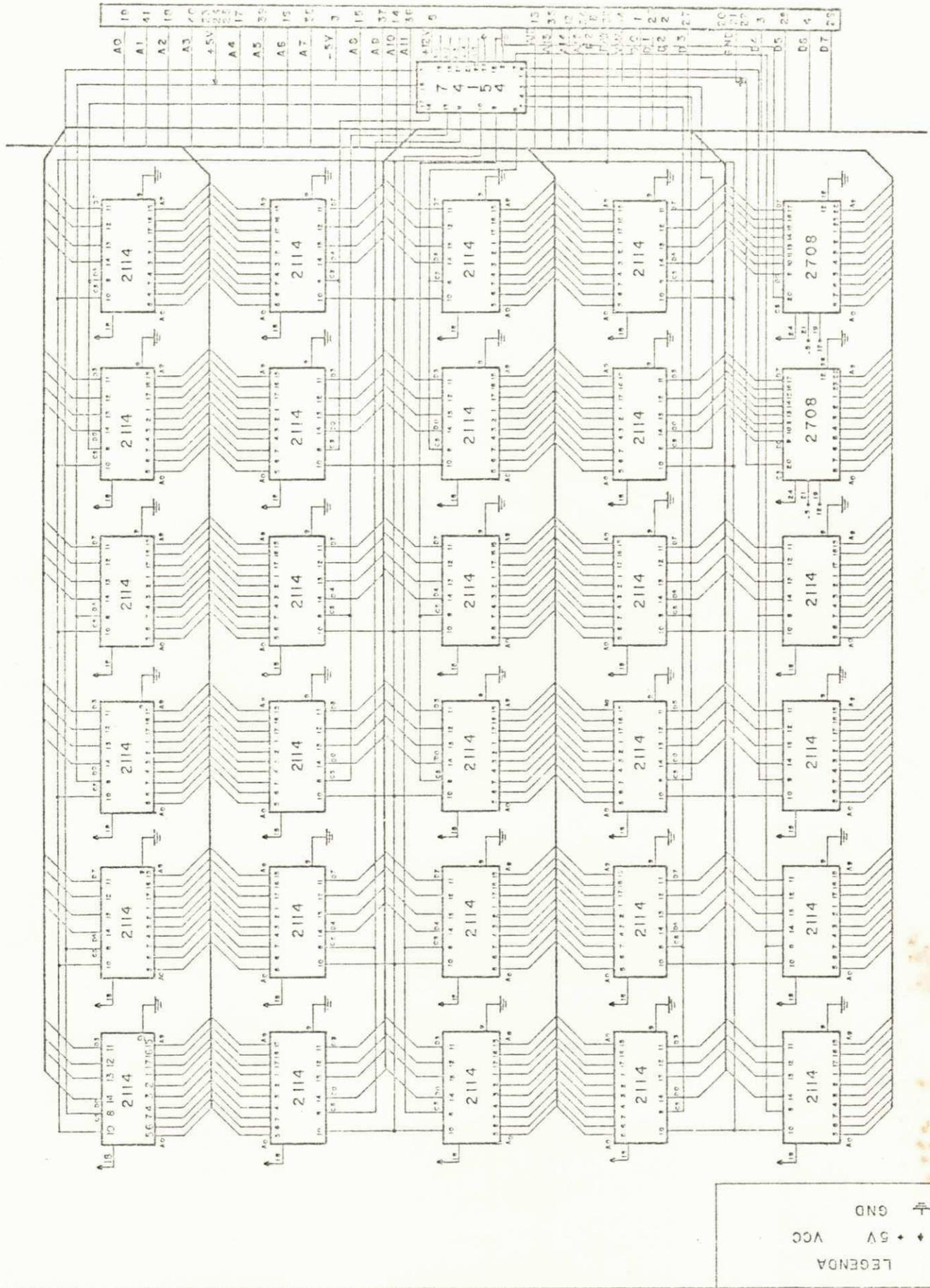
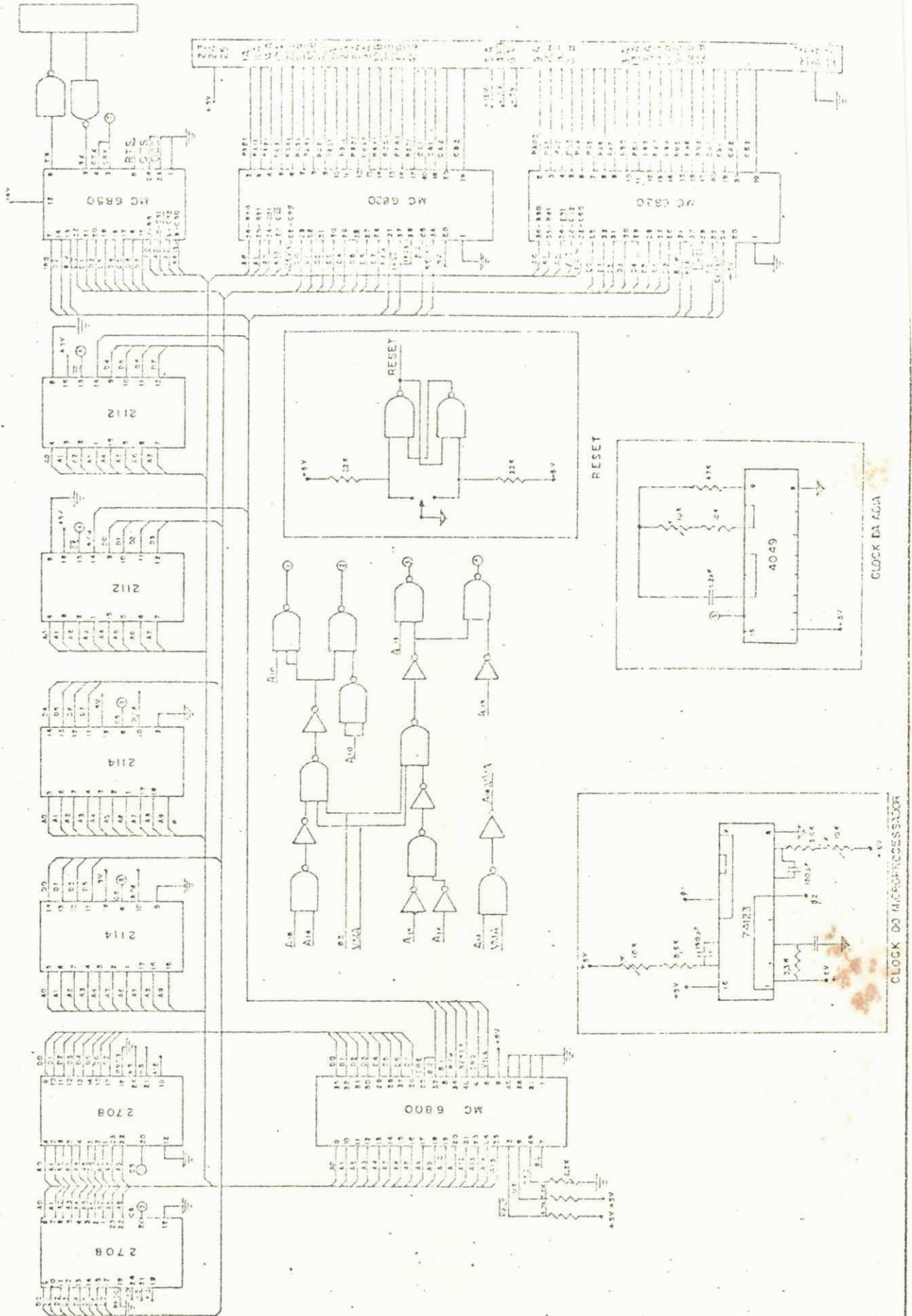


DIAGRAMA DE LIGAÇÕES DA M.C.

LEGENDA
 + + 5V VCC
 ⊥ GND

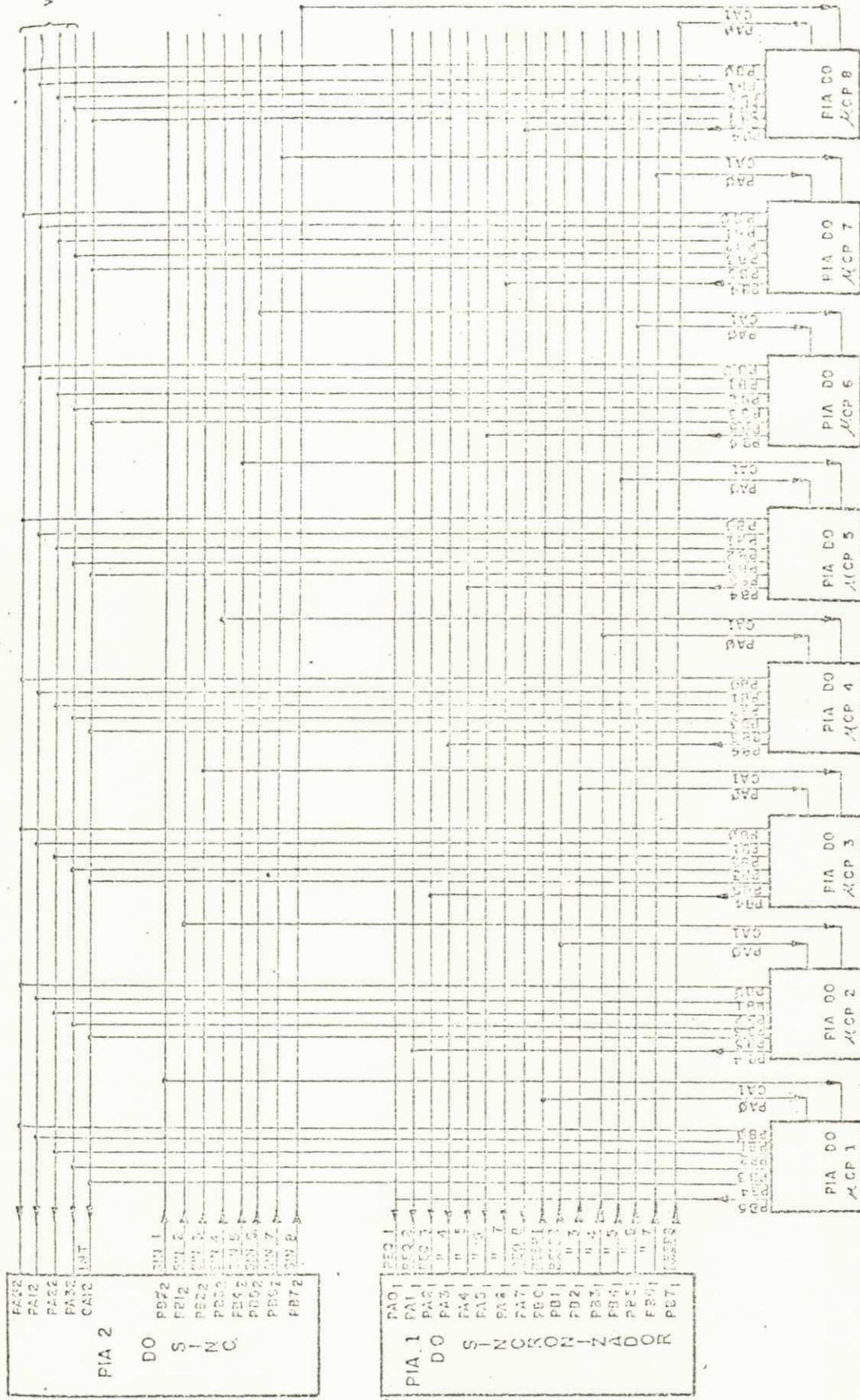
Esquema de ligações do ACS



CLOCK DO MICROPROCESSOR

CLOCK DA AGA

VIA DE CODIGO



Ligação do ucs com os oito ucp's

APENDICE D

PROGRAMAS DO MATER

ADRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
F400	86	80		LDAA	# 80	uPi fecha as chaves
F402	87	C002		STAA	C002	de acesso à M.C.
F405	86	04		LDAA	# 04	
F407	87	C003		STAA	C003	
F40A	86	80		LDAA	# 80	
F40C	87	C002		STAA	C002	
F40F	86	AA		LDAA	# AA	uPi testa se M.C.es
F411	B1	4800		CMPA	4800	ta organizada.
F414	27	0A		BEQ	TAREFA	
F416	86	55		LDAA	# 55	uPV identifica uPi.
F418	B1	480C		CMPA	480C	
F41B	27	06		BEQ	ORGANIZA	
F41D	7E	FF9C		JMP	SATER.1	M.C. não esta organi zada
F420	7E	4075	TAREFA	JMP	SUB-TAR	M.C. esta organizada uPV assume a gerên- cia da M.C.
F423	7E	4025	ORGANIZA	JMP	SUB-ORG	M.C.não esta organi- zada;uPV organiza a M.C.
F426	86	80		LDAA	# 80	Substância FIM- uPi
F428	B7	C002		STAA	C002	abre as chaves de a-
F42B	86	00		LDAA	# 00	cesso a M.C. e o con
F42D	B7	C003		STAA	C003	trole volta para o
F430	86	80		LDAA	# 80	SATER.1
F432	B7	C002		STAA	C002	
F435	7E	FF9C		JMP	SATER.1	
F438			ESTOURO			M.C. cheia,controla volta para o sistema operacional

ADRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
4000	7F	480D	TROCAMSG	CLR	FINAL	Esta subrotina troca
4003	7D	480D	LOOP	TST	FINAL	mensagens (HEADER,
4006	26	1C		BNE	FIM	mensagem curta e men
4008	FE	480E		LDX	ENDI1	sagem longa) entre
400B	A6	00		LDA	0,X	os uPi e o uPV e vi-
400D	08			INX		ce versa.
400E	FF	480E		STX	ENDI1	
4011	BC	4812		CPX	ENDIFIM	
4014	26	03		BNE	NU	
4016	7C	480D		INC	FINAL	
4019	FE	4810	NU	LDX	ENDI2	
401C	A7	00		STAA	0,X	
401E	08			INX		
401F	FF	4810		STX	ENDI2	
4022	20	DF		BRA	LOOP	
4024	39		FIM	RTS		
4025	CE	4E00	SUBORG	LDX	4E00	Subrotina organiza ,
4028	86	00		LDA	# 00	uPV cria tabelas de
402A	C6	50		LDAB		apontadores dos HEA-
402C	E7	00	Volta 1	STAB	0,X	DER, mensagens longas
402E	A7	01		STAA	1,X	e mensagens curtas.
4030	08			INX		
4031	08			INX		
4032	8B	30		ADDA	# 30	
4034	C9	00		ADCB	# 00	
4036	8C	5000		CPX	# 5000	
4039	2D	F1		BLT	Volta 1	
403B	CE	4980		LDX	# 4980	
403E	86	00		LDA	# 00	
4040	C6	4A		LDAB	# 4A	

ADRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
4042	E7	00	Volta 2	STAB	0,X	
4044	A7	01		STAA	1,X	
4046	08			INX		
4047	08			INX		
4048	8B	10		ADDA	1,0	
404A	C9	00		ADCB	00	
404C	8C	4A00		CPX	4A00	
404F	2D	F1		BLT	Volta 2	
4051	86	1F		LDAA	1F	uPV cria contadores
4053	B7	4801		STAA	CONTHD	de HEADER, mensagem
4056	B7	4802		STAA	CONIMC	curta e mensagem
4059	86	FF		LDAA	CONIML	longa.
405B	B7	4803		STAA	4803	
405E	CE	4E00		LDX	# 4E00	uPV cria apontadores
4061	FF	4808		STX	APONTHD	dos HEADER, mensagem
4064	CE	4980		LDX	# 4980	curta e mensagem lon
4067	FF	4804		STX	APONIMC	ga
406A	CE	49C0		LDX	# 49C0	
406D	FF	4806		STX	APONIML	
4070	86	AA		LDAA	# AA	uPV indica que a M.
4072	B7	4800		STAA	# 4800	C. esta organizada.
4075	CE	0C00		LDX	# ENDHEAD	Subrotina TAREFA ini
4078	FF	4822		STX	# CONIEH	ciabilização
407B	CE	0900		LDX	# DESTHEAD	ENDHEAD=Endereço do
407E	FF	4824		STX	TEMPH 2	HEADER
4081	7F	4818		CLR	CONTA 3	
4084	7F	4819		CLR	CONTA 4	
4087	7F	4817		CLR	CONTE	
408A	7F	4814		CLR	AUX 0	
408D	CE	4A01		LDX	# 4A01	

ADDRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
4090	FF	482A		STX	AUX 4H	
4093	CE	0E00		LDX	# 0E00	
4096	FF	482E		STX	AUX 17H	
4099	08			INX		
409A	08			INX		
409B	FF	483E		STX	AUX 16H	
409E	CE	0D00		LDX	# 0D00	
40A1	FF	482C		STX	AUX 5 H	
40A4	86	00		LDAA	# 00	
40A6	CE	4801		LDX	# 4801	
40A9	E6	00		LDAB	0,X	uPV testa se pode re
40AB	11			CBA		ceber HEADER
40AC	27	36		BEQ	PULO 1	
40AE	FE	483E		LDX	AUX 16H	
40B1	A6	00		LDAA	0,X	
40B3	C6	52		LDAB	# 52	
40B5	11			CBA		
40B6	26	39		BNE	PULO 2	
40B8	FE	482A		LDX	AUX 4H	
40BB	A6	00	Volta 3	LDAA	0,X	
40BD	F6	0000		LDAB	nº uPi	
40C0	11			CBA		
40C1	27	31		BEQ	TEM MSG	uPi recebe mensagem.
40C3	FE	482A		LDX	AUX 4 H	
40C6	B6	482A		LDAA	AUX 4 H	
40C9	F6	482B		LDAB	AUX 4 L	
40CC	CB	10		ADDB	# 1 0	
40CE	89	00		ADCA	# 00	
40D0	B7	482A		STAA	AUX 4 H	
40D3	F7	482B		STAB	AUX 4 L	

ADRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
40D6	FE	482A		LDX	AUX 4H	
40D9	8C	4C01		CPX	# 4C01	
40DC	27	03		BEQ	SUB FIM	
40DE	7E	40BB		JMP	Volta 3	uPV procura novo HEADER para uPi
40E1	7E	F426	SUB FIM	JMP	FIM	Não tem mensagens para o uPi.
40E4	FE	483E	PULO 1	LDX	AUX 16H	
40E7	A6	00		LDAA	0,X	
40E9	C6	52		LDAB	# 52	
40EB	11			CBA		
40EC	27	CD		BEQ	Volta 3	
40EE	7E	F438		JMP	Estouro	M.C. esta cheia, não há espaço disponível
40F1	7E	4246	PULO 2	JMP	uPV RH	uPV recebe HEADER do uPi.
40F4	09		TEM MSG	DEX		UPi recebe HEADER
40F5	FF	480E		STX	480E	
40F8	FF	4815		STX	AUX 1H	
40FB	B6	4816		LDAA	AUX 1L	
40FE	8B	0F		ADDA	# 0F	
4100	B7	4816		STAA	AUX 1L	
4103	FE	4815		LDX	AUX 1H	
4106	FF	4812		STX	4812	
4109	FE	482C		LDX	AUX 5H	
410C	FF	4810		STX	4810	
410F	FE	4804		LDX	4804	uPV atualiza contado res e apontadores.
4112	8C	4980		CPX	# 4980	
4115	27	15		BEQ	Pulo 3	

ADDRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
4117	Ø9			DEX		
4118	Ø9			DEX		
4119	FF	48Ø4		STX	48Ø4	
411C	B6	4815		LDAA	AUX1H	
411F	F6	4816		LDAB	AUX1L	
4122	A7	ØØ		STAA	Ø,X	
4124	E7	Ø1		STAB	1,X	
4126	FE	4815		LDX	AUX1H	
4129	FF	4828		STX	AUX3H	
412C	B6	48Ø1	PULO 3	LDAA	CONTHD	
412F	4C			INCA		
413Ø	B7	48Ø1		STAA	CONTHD	
4133	BD	4ØØØ		JER	TROCANSG	uPV transmite HEA- DER para uPi
4136	FE	482C	TESTE ML	LDX	AUX5H	uPV testa se tem
4139	A6	Ø5		LDAA	5,X	mensagem curta.
413B	81	43		CMPA	# 43	
413D	27	35		BEQ	PULO4	
413F	FE	482C	MSG L	LDX	AUX 5H	uPV testa se tem
4142	A6	Ø9		LDAA	9,X	mensagem longa
4144	81	ØØ		CMPA	ØØ	
4146	26	6E		BNE	PULO5	
4148	FE	482C		LDX	AUX 5H	uPV testa se tem
414B	A6	Ø6		LDAA	6,X	novo HEADER p/uPi
414D	81	Ø1		CMBA	# Ø,1	
414F	27	ØØ		BEQ	BFIM	
4151	7C	4817		INC	CONTE	uPV testa nº de
4154	F6	4817		LDAB	CONTE	HEADER

ADDRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
4157	C1	Ø3		CMPB	≠ Ø3	
4159	2E	16		BGT	BFIM	
415B	B6	482D		LDAA	AUX 5L	uPV atualiza as
415E	8B	1Ø		ADDA	≠ 1Ø	variaveis auxilia
416Ø	B7	482D		STAA	AUX 5L	res e procura no-
4163	B6	482B		LDAA	382B	vo HEADER para o
4166	8B	1Ø		ADDA	≠ 1Ø	uPi.
4168	B7	4828		STAA	482B	
416B	FE	482A		LDX	482A	
416E	7E	4ØBB		JMP	Volta3	
4171	7E	F426	BFIM	JMP	FIM	uPV volta o con-
						trole ao uPi. Fin
						das tarefas.
4174	FE	482C	PULO 4	LDX	AUX 5H	uPi recebe mensa-
4177	A6	Ø7		LDAA	7,X	gem curta.
4179	E6	Ø8		LDAB	8,X	
417B	B7	48ØA		STAA	TEMP H	
417E	F7	48ØB		STAB	TEMP L	
4181	FE	48ØA		LDX	TEMP H	
4184	FF	48ØE		STX	48ØE	
4187	FE	4822		LDS	CONTE H	
418A	FF	481Ø		STX	481Ø	
418D	B6	48ØA		LDAA	TEMP H	
419Ø	F6	48ØB		LDAB	TEMP L	
4193	CB	1F		ADDB	≠ 1 F	
4195	89	ØØ		ADCA	ØØ	
4197	B7	48ØA		STAA	TEMP H	
4199	F7	48ØB		STAB	TEMP L	
419D	FE	48ØA		LDX	TEMP H	

ADRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
41A0	FF	4812		STX	4812	
41A3	ED	4000		JSR	TROCA MSG	
41A6	B6	4822		LDAA	CONTE H	uPV salva endere-
41A9	F6	4823		LDAB	CONTE L	ço da mensagem
41AC	CB	10		ADDB	10	curta.
41AE	89	00		ADEA	00	
41B0	B7	4822		STAA	CONTE H	
41B3	F7	4823		STAB	CONTE L	
41B6	7E	413F		JMP	MSG L	
41B9	FE	4824	PULO 5	LDX	TEMP H 2	uPi recebe mensa-
41BC	FF	4810		STX	4810	gem longa.
41BF	FE	482C		LDX	AUX 5 H	
41C2	A6	0A		LDAA	A,X	
41C4	E6	0B		LDAB	B,X	
41C6	B7	4832		STAA	AUX 8 H	
41C9	F7	4833		STAB	AUX 8 L	
41CC	FE	482C		LDX	AUX 5 H	
41CF	A6	09		LDAA	9,X	
41D1	B7	4814		STAA	AUX 0	
41D4	FE	4832		LDX	AUX 8 H	
41D7	FF	480E	NOVA ML	STX	480E	
41DA	FF	4830		STX	CONTA H	
41DD	B6	4831		LDAA	CONTA L	
41E0	F6	4830		LDAB	CONTA H	
41E3	8B	2F		ADDA	2F	
41E5	C9	00		ADCB	00	
41E7	B7	4831		STAA	CONTA L	
41EA	F7	4830		STAB	CONTA H	
41ED	FE	4830		LDX	CONTA H	

ADRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
41F0	FF	4812		STX	4812	
41F3	BD	4000		JSR	TROCA MSG	
41F6	B6	4814		LDAA	AUX 0	uPV atualiza os a- pontadores e conta dores de mensagem longa.
41F9	F6	4803		LDAB	4803	
41FC	4A			DECA		
41FD	SC			INCB		
41FE	F7	4803		STAB	4803	
4201	B7	4814		STAB	AUX 0	
4204	FE	4832		LDX	AUX 8 H	
4207	FF	4834		STX	TROCP H	
420A	FE	4832		LDX	AUX 8 H	
420D	09			DEX		
420E	09			DEX		
420F	B6	4834		LDAA	TROCP H	
4212	A7	00		STAA	0,X	
4214	F6	4835		LDAB	TROCP L	
4217	E7	01		STAB	1,X	
4219	B6	4814		LDAA	AUX 0	
421C	81	00		CMIA	00	
421E	26	03		BNE	NOVA PD	
4220	7E	40BB		JMP	Volta 3	
4223	B6	4810	NOVA PD	LDAA	4810	uPi recebe nova pa- gina de dados da mensagem longa.
4226	F6	4811		LDAB	4811	
4229	CB	30		ADDB	30	
422B	89	00		ADCB	00	
422D	B7	4810		STAA	4810	
4230	F7	4811		STAB	4811	
4233	FE	4812		LDX	4812	
4236	A6	00		LDAA	0,X	

ADDRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
4238	F6	Ø1		LDAB	1,X	
423A	B7	4836		STAA	AUX 10H	
423D	F7	4837		STAB	AUX 10L	
424Ø	FE	4836		LDX	AUX 10H	
4243	7E	42FF		JMP	NOVA ML	
4246	FE	482E	uPV RH	LDX	AUX 17X	uPV testa esforço
4249	A6	Ø5		LDAA	5,X	da M.C.
424B	81	43		CMPA	43	
424D	26	5C		BNE	TESTE ML	
424F	F6	48Ø2		LDAB	48Ø2	
4252	C1	1F		CMPB	1F	
4254	27	Ø8		BEQ	FORA	
4256	A6	Ø9		LDAA	9,X	
4258	F6	48Ø3		LDAB	48Ø3	
425B	11			CBA		
425C	2E	Ø3		BGT	uPV PR	
425E	7E	F438	FORA	JMP	ESTOURO	
4261	FF	48ØE	uPV PR	STX	48ØE	uPV recebe Header
4264	FE	48Ø4		LDX	48Ø4	do uPi
4267	FF	481Ø		STX	481Ø	
426A	FE	482E		LDX	AUX 17H	
426D	E6	Ø1		LDAB	1,X	
426F	CB	1F		ADDB	1F	
4271	A6	ØØ		LDAA	Ø,X	
4273	B7	4812		STAA	4812	
4276	F7	4813		STAB	4813	
4279	B6	48Ø1		LDAA	CONTHD	
427C	4C			INCA		
427D	B7	48Ø1		STAA	CONTHD	

ADDRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
4280	B6	4804		LDAA	4804	
4283	F6	4805		LDAB	4805	
4286	CB	10		ADDB	10	
4288	89	00		ADCA	00	
428A	B7	4804		STAA	4804	
428D	F7	4805		STAB	4805	
4290	BD	4000		JSR	TROCAMSG	
4293	FE	482E		LDX	AUX 17H	uPV testa se rece-
4296	A6	05		LDAA	5,X	be mensagem curta.
4298	81	43		CMPA	43	
429A	27	1F		BEQ	RECEBE MC	
429C	A6	09	Volta 7	LDAA	9,X	uPV testa se rece-
429E	81	00		CMPA	00	be mensagem longa.
42A0	26	19		BNE	RECEBEM L	
42A2	7C	4818		INC	CONTA 3	uPV testa se rece-
42A5	B6	4818		LDAA	CONTA 3	be novo HEADER do
42A8	E6	06		LDAB	6,X	uPi.
42AA	11			CBA		
42AB	2D	0B		BLT	FIM 2	
42AD	F6	482F		LDAB	AUX 17L	
42B0	CB	10		ADDB	10	
42B2	F7	482F		STAB	AUX 17L	
42B5	7E	40BB		JMP	uPV RH	
42B8	7E	F426	FIM 2	JMP	FIM	
42BB	B6	4802	RECEBE MC	LDAA	4802	uPV recebe mensagen
42BE	4A			DECA		curta do uPi
42BF	B7	4802		STAA	4802	

ADDRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
42C2	A6	Ø7		LDAA	7,X	
42C4	E6	Ø8		LDAB	8,X	
42C6	B7	48ØE		STAA	48ØE	
42C9	F7	48ØF		STAB	48ØF	
42CC	FE	48Ø6		LDX	48Ø6	
42CF	FF	481Ø		STX	481Ø	
42D2	Ø8			INX		
42D3	Ø8			INX		
42D4	FF	48Ø6		STX	48Ø6	
42D7	B6	481Ø		LDAA	481Ø	
42DA	F6	4811		LDAB	4811	
42DD	CB	2F		ADDB	2F	
42DF	89	ØØ		ADCB	ØØ	
42E1	B7	4812		STAA	4812	
42E4	F7	4813		STAB	4813	
42E7	BD	4ØØØ		JSR	TROCA MSG	
42EA	&E	429C		JMP	VOLTA 7	
42ED	FE	482E	RECEBEM L	LDX	AUX 17	uPV recebe mensagen
42FØ	A6	Ø9		LDAA	9,X	longa do uPi
42F2	B7	4819		STAA	CONTA 4	
42F5	A6	ØA		LDAA	A,X	
42F7	E6	ØB		LDAB	B,X	
42F9	B7	48ØE		STAA	48ØE	
42FC	F7	48ØF		STAB	48ØF	
42FF	FE	48Ø8	NOVA ML	LDX	48Ø8	
43Ø2	FF	481Ø		STX	481Ø	
43Ø5	B6	48ØE		LDAA	48ØE	
43Ø8	F6	48ØF		LDAB	48ØF	
43ØB	CB	2D		ACCB	2D	
43ØD	89	ØØ		ADCA	ØØ	

ADDRESS	OP.	CODE	LABEL	MNEM	OPERANDO	COMENTÁRIOS
430F	B7	4812		STAA	4812	
4312	F7	4813		STAB	4813	
4315	FE	4812		LDX	4812	
4318	FF	4826		STX	AUX20H	
431B	FE	4808		LDX	4808	uPV atualiza os
431E	08			INX		contadores e apon
431F	08			INX		tadores da mensa-
4320	FF	4808		STX	4808	gem longa.
4323	B6	4803		LDAA	4803	
4326	4A			DECA		
4327	B7	4803		STAA	4803	
432A	BD	4000		JSR	TROCA MSG	
432D	B6	4808		LDAA	4808	uPV recupera apon
4330	F6	4809		LDAB	4809	tadores e verifi-
4333	FE	4812		LDX	4812	ca se tem outra
4336	A7	00		STAA	0,X	mensagem longa.
4338	E7	01		STAB	1,X	
433A	B6	4819		LDAA	CONTA 4	
433D	4A			DECA		
433E	81	00		CMPA	00	
4340	27			BEQ	PULO 8	
4341	FE	4826		LDX	AUX 20H	
4344	08			INX		
4345	FF	480E		STX	480E	
4348	7E	42FF		JMP	NOVA ML	
434B	7E	F426	PULO 8	JMP	FIM	uPV já recebeu to- das as informações do uPi.

UNIVERSIDADE FEDERAL DA PARAIBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprigio Veloso 882 - Tel (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba

B I B L I O G R A F I A

- 1 - SATAYNARAYNAM; M - Comercial Multiprocessing Systems
Computer - May 1980 - pág 75/116
- 2 - ROSICA; G. A. - The Digital Computer Interface
Electronics Engineer's Handbook - 1ª Edição -
MacGraw-Hill Book Company - 1975 - pág 22/30
- 3 - FULLER; S. H. "et alii" - Multi-microprocessor: An
Overview and Working Example -
Proceedings of IEEE - Vol 66 - nº 02 - February 1978 -
pág 216/228
- 4 - ENSLOW Jr; P. H. - Multiprocessor Organization -
A Survey
Computing Systems - Vol 09 - nº 01 - March 1977 -
pág 103/129
- 5 - SEARLE; B. C. And FREBEG; D.E. - Tutorial: Micropro-
cessor Application in Multiple Processor Systems -
Computer - October 1975 - pág 22/30
- 6 - NUTT; G.J. - A Parallel Processor Operating Systems
Comparison -
Transactions on Software Engineering - Vol SE-3 -
nº 06 - November 1977
- 7 - ANDERSON; D.A. - Operating Systems
Computer - June 1981 - pág 69/82
- 8 - PEATMAN; J. B. - Microcomputer - Based Design
McGraw-Hill Book Company - 1977

- 9 - SRODAWA; R. J. - Positive Experiences with a Multi-processing Systems -
Computer Surveys - vol 10 - March 1978 - pág 73/82
- 10 - RAMANOORTHY; C. V. - Pipeline Architecture -
Computer Surveys - vol 09 - nº 01 - March - 1977
- 11 - YAN; S. S. and FUNG; H. S. - Associative Processor Architecture - A Survey Computer Surveys - vol 09 - nº 01 - March 1977
- 12 - ASHANGY; R. "et alii" - Shift Register Interconnection System - U.S. Patent 3,735,362 MAY, 1973
- 13 - SILVA; I. S. S. - Sincronizador para Multiprocessador
Tese de Mestrado DEE/CCT/UFPb - Setembro de 1981
- 14 - CAVALCANTI; J. H. F. e DEEP; G.S. - Um Kernel para controle de Processos Industriais em tempo real usando Microcomputadores - Anais do VII Seminário Integrado de SOFTWARE e HARDWARE - Campinas - SP 1980
- 15 - Memory Databook
National Semiconductor - 1977 - pág 22/25
- 16 - Signetics Bipolar & MOS Memory - Data Manual
Signetics - 1977
- 17 - M 6800 - Systems Reference and Data Sheets -
Motorola Semiconductor Products INC - 1975
- 18 - The TTL Databook for Design Engineers -
Texas Instruments - 2a Edição - 1976
- 19 - CMOS Databook
National Semiconductor - 1978
- 20 - MCS - 48 Microcomputer User's Manual
INTEL - 1977 - pág 7/83 à 7/87

- 21 - Linear Databook
National Semiconductor - 1978
- 22 - CAVALCANTI; J. H. F. "et Alii" - SOFTR - Sistema
de Desenvolvimento de Software de Microprocessado-
res para aplicações em Tempo Real -
Panel 81 - EXPODATA - Buenos Aires - Argentina.