

MEMÓRIA DESCRITIVA

DA

PATENTE DE INVENÇÃO

Nº 94 055

NOME: INTERNATIONAL BUSINESS MACHINES CORPORATION, norte-
-americana, (estado: New York), com sede em Armonk,
N.Y. 10504, Estados Unidos da América.

EPÍGRAFE: "MEMÓRIA PRINCIPAL FÍSICA ÚNICA COMPARTILHADA
POR DOIS OU MAIS PROCESSADORES QUE EXECUTAM
SISTEMAS OPERATIVOS RESPECTIVOS"

INVENTORES: Ernest D. Baker, John M. Dinwiddie, Lonnie E.
Grice, John M. Loffredo, Kenneth R. Sanderson
e Gustavo A. Suarez, residentes nos E.U.A.)

Reivindicação do direito de prioridade ao abrigo do artigo
4º da Convenção da União de Paris de 20 de Março de 1883.

Estados Unidos da América - 17 de Maio de 1989, sob o nú-
mero de série 07/353,113.

Descrição referente à patente de invenção de INTERNATIONAL BUSINESS MACHINES CORPORATION, norte-americana, (estado: New York) industrial e comercial, com sede em Armonk, N.Y. 10504, Estados Unidos da América, (inventores: Ernest D. Baker, John M. Dinwiddie, Lonnie E. Grice, John M. Loffredo, Kenneth R. Sanderson e Gustavo A. Suarez, residentes nos E.U.A.), para "MEMÓRIA PRINCIPAL FÍSICA ÚNICA COMPARTILHADA POR DOIS OU MAIS PROCESSADORES QUE EXECUTAM SISTEMAS OPERATIVOS RESPECTIVOS".

DESCRIÇÃO

O presente pedido de patente relaciona-se com outros pedidos de patente do mesmo proprietário depositados no mesmo dia.

Esses outros pedidos de patente são:

<u>Nº. de Série</u>	<u>Título</u>	<u>Inventores</u>
07/353116	Fault Tolerant Data Processor System	E.D. Baker J.M. Dinwiddie L.E. Grice J.M. Joyce J.M. Loffredo K.R. Sanderson G.A. Suarez
07/353114	Uncoupling A Central Processing Unit From Its Associated Hardware for Interaction With Data Handling Apparatus Alien To The Operating System Controlling Said Unit and Hardware	E.D. Baker J.M. Dinwiddie L.E. Grice J.M. Joyce J.M. Loffredo K.R. Sanderson
07/353117	Servicing Interruptor Requests In A Data Processing System Without Using The Services Of An Operating System	J.M. Dinwiddie L.E. Grice J.M. Loffredo K.R. Sanderson
07/353111	Providing Additional System Characteristics To A Data Processing System	E.D. Baker J.M. Dinwiddie L.E. Grice J.M. Joyce J.M. Loffredo K.R. Sanderson
07/353115	Method And Apparatus For The Direct Transfer of Information Between Application Programs Running On Distinct Processors Without Utilizing The Services Of One Or Both Operating Systems	E.D. Baker J.M. Dinwiddie L.E. Grice J.M. Joyce J.M. Loffredo K.R. Sanderson
07/353112	Data Processing System With System Resource Management For An Associated Alien Processor	J.M. Dinwiddie B.J. Freeman L.E. Grice J.M. Loffredo K.R. Sanderson G.A. Suarez



Fundamento da invenção

O aperfeiçoamento a que se refere a presente invenção diz respeito a um processo e aos meios pelos quais duas unidades de processamento central (CPU), cada uma delas operando com o seu sistema operativo respectivo, compartilham uma unidade de memória principal física única, caracterizados por cada um dos sistemas operativos funcionar como se ele controlasse toda a memória do seu sistema configurado e como se não tivesse conhecimento do outro sistema operativo.

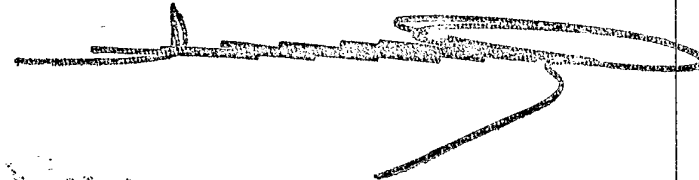
Introdução

São conhecidos muitos sistemas de processamento de dados usando uma memória principal física, uma parte da qual é compartilhada por duas CPU. Porém, tanto quanto é conhecido, estes sistemas têm um sistema operativo usado pelas duas CPU ou têm CPU com sistemas operativos respectivos que não ignoram a existência das duas CPU e dos dois sistemas operativos, por exemplo, através das suas tabelas de configuração. Quando necessário, usa-se a arbitragem das linhas omnibus para permitir o acesso à memória comum.

Sumário da invenção

Proporciona-se um processo e meios aperfeiçoados para capturar uma secção ou zona da memória principal a partir de um primeiro sistema de processamento de dados, incluindo um primeiro elemento de processamento, a memória principal e aparelhos de entrada/saída (I/O) operados sob controlo de um primeiro sistema operativo, para utilização por um segundo elemento de processamento que possui meios que acoplam o segundo elemento de processamento à memória principal e operando sob o controlo de um segundo sistema operativo, de uma maneira indiscernível para os dois sistemas operativos.

Numa forma de realização preferida, um



gestor da memória no primeiro sistema operativo cria uma lista de entradas correspondentes a blocos de memória não usados, para distribuir memória aos processos. Um programa de aplicação, que é passado no modo de supervisão no primeiro elemento de processamento, retira da lista um grupo de entradas correspondente a uma área contígua de memória com uma dimensão pré-determinada. Os dados de endereço correspondentes à referida área contígua da memória são transferidos para os referidos meios de acoplamento para permitir o acesso à área contígua pelo referido segundo elemento de processamento.

Durante a execução normal das instruções, é dado acesso ao segundo elemento de processamento à referida área de memória contígua e dá-se acesso do primeiro elemento de processamento à área restante da memória. Porém, a um programa de aplicação especial que é passado no primeiro elemento de processamento (mas não ao primeiro sistema operativo) é dado acesso à referida área de memória contígua.

Breve descrição dos desenhos

Nos desenhos anexos, as figuras representam:

A fig. 1, esquematicamente, os sistemas normalizados de computadores de interligação utilizando uma linha de telecomunicações;

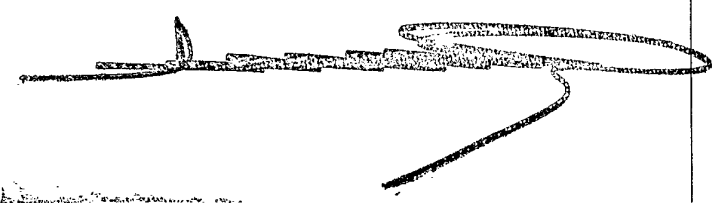
A fig. 2, esquematicamente a interligação de processadores S/88 num ambiente que tolera avarias;

A fig. 3, esquematicamente, a interligação de processadores S/370 com processadores S/88 na forma de realização preferida;

A fig. 4, esquematicamente, um sistema S/370 acoplado a um sistema S/38 de acordo com a forma de realização preferida;

A fig. 5, esquematicamente, o desacoplamento de um processador S/88 para proporcionar a permuta de dados entre o S/370 e o S/88 da forma de realização preferida;

As fig. 6A, 6B e 6C, esquematicamente,



o módulo da técnica anterior IBM System/88, vários módulos interligados por interligações de dados de alta velocidade (HSDI) e vários módulos interligados através de uma rede, num ambiente que tolera avarias, com uma imagem de sistema único;

A fig. 7, esquematicamente, uma forma do módulo aperfeiçoado segundo a presente invenção que proporciona processadores S/370 que executam programas de aplicação S/370 sob o controlo de um sistema operativo S/370, que se tornam tolerantes para avarias em virtude da maneira como os processadores estão ligados uns aos outros e a processadores S/88, às I/O e à memória principal;

A fig. 8, esquematicamente, com mais pormenor, a interligação de unidades S/370 e S/88 emparelhadas uma com as outras para formar uma unidade de processador e a sua ligação a uma unidade de processador idêntica cooperante para o funcionamento com tolerância de avarias;

As fig. 9A e 9B, uma forma de acondicionamento de unidades S/370 e S/88 emparelhadas em dois cartões para inserção no painel traseiro de um invólucro de um sistema de processamento;

A fig. 10, teoricamente, a memória principal S/88 e secções dessa memória dedicadas a unidades de processador S/370 sem conhecimento por parte do sistema operativo S/88;

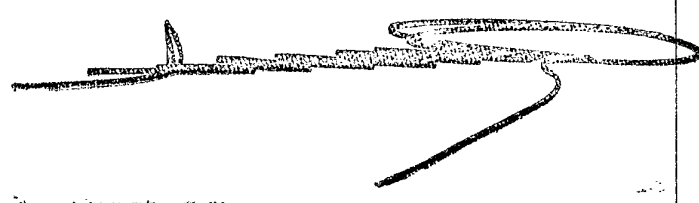
A fig. 11, esquematicamente, certos componentes da forma preferida de um processador S/370 e meios que o ligam a um processador e memória S/88;

A fig. 12, os componentes da fig. 11 com mais pormenor e vários componentes de uma forma de realização de um processador S/88;

A fig. 13, esquematicamente, o adaptador de linhas omnibus S/370;

As fig. 14A, 14B e 15A-C, teoricamente, a distribuição de tempos e o movimento de dados através dos canais de saída do adaptador de linhas omnibus S/370;

A fig. 16, esquematicamente, a interligação directa entre um processador S/370 e um S/88, com mais por



menor;

A fig. 17, teoricamente, o fluxo de dados entre um adaptador de linhas omnibus S/370 e um controlador do DMA (acesso directo à memória) da interligação da fig. 16;

A fig. 18, registadores DMAC para um dos seus quatro canais;

As fig. 19A, 19B e 19C (com a implantação da fig. 19), numa representação esquemática, a ilustração que mostra, com mais pormenor que a fig. 16, uma forma preferida da unidade de controlo das linhas omnibus que interliga um processador S/370 com um processador S/88 e a memória principal;

A fig. 20, um diagrama esquemático de uma forma preferida do desacoplamento lógico do processador S/88 do seu equipamento material do sistema associado e da lógica para tratar pedidos de interrupção provenientes do processador S/370 estranho para o processador S/88;

A fig. 21, teoricamente, a modificação da estrutura de interrupções do S/88 existente para um módulo que tem um certo número de processadores interligados S/370-S/88 de acordo com os ensinamentos da presente invenção;

As fig. 22, 23 e 24, diagramas de distribuição de tempos de ciclos de leitura, de escrita e confirmação de interrupção da forma preferida dos processadores S/88;

As fig. 25 e 26, diagramas de distribuição de tempos para estabelecimento de sincronização por permuta de informações ("handshaking") para os canais 0,1 da linha omnibus do adaptador, durante comandos de leitura da caixa de correio ("mailbox"), comandos de selecção Q, comandos de leitura e comandos de escrita BSM;

A fig. 27, uma esquema de blocos de uma forma preferida de um elemento central de processamento S/370;

As fig. 28 e 29, determinadas áreas da memória central e da memória de controlo S/370;

A fig. 30, uma forma preferida das linhas omnibus de interface entre o elemento central de processamento S/370, o adaptador de I/O, o controlador da memória temporária rápida (memória "cache"), da interface de controlo da me-



mória e linhas omnibus do sistema S/88 e o processador;

A fig. 31, um diagrama de blocos de uma forma preferida de um controlador da memória "cache" S/370;

As fig. 32A e 32B (com a implantação na fig. 32), esquematicamente, uma forma preferida da interface de controlo da memória, com mais pormenor;

A fig. 33, um diagrama de distribuição de tempos, ilustrando as fases da linha omnibus do sistema S/88 para a transferência de dados entre unidades pelas linhas omnibus;

A fig. 34, um esquema parcial que mostra os registadores "data in" (de entrada de dados) de uma interface de um controlo de memória emparelhado;

A fig. 35, formatos das palavras de comando e armazenamento armazenadas no FIFO (primeira entrada, primeira saída) da fig. 32B;

As fig. 36A-D comandos de armazenamento e busca provenientes do processador e adaptador S/370, que são executados na interface do controlo da memória;

A fig. 37, teoricamente, a forma de realização preferida do sistema global segunda a presente invenção, do ponto de vista do programador;

As fig. 38, 39 e 40, esquematicamente, formas preferidas da concepção dos microcódigos para a interface S/370 e S/88, da execução do comando de I/O S/370 e da partilha da interface entre os meios de programação EXEC 370 e o actuador de I/O S/370 (isto é o microcódigo ETIO + BCU + S/370), respectivamente;

As fig. 41A e 41B, teoricamente, interfaces e protocolos entre os meios de programação EXEC 370 e o microcódigo S/370 e entre o microcódigo CTIO e os meios de programação EXEC 370;

As fig. 41C-H o conteúdo da memória local BCU, incluindo memórias tampão de dados, memórias tampão de filas de espera de trabalho, filas de espera, áreas de comunicação de filas de espera e áreas de comunicação de equipamentos materiais, incluindo a lista de elos de ligação e o movimento



das memórias tampão de filas de espera através das filas de espera, elementos que compreendem o protocolo através do qual o microcódigo S/370 e os meios de programação EXEC 370 comunicam entre si;

A fig. 42, teoricamente, o movimento das memórias tampão das filas de espera de trabalho através da lista de elos de ligação e das filas de espera, em conjunção com os protocolos entre EXEC 370, ETIO, microcódigo S/370 e o equipamento material de acoplamento S/370-S/88;

A fig. 43, teoricamente, e execução de uma instrução S/370 Start I/O típica;

As fig. 44S-L, esquematicamente, os fluxos de dados de controlo para o microcódigo S/370 e EXEC 370 quando eles comunicam um com o outro para executar cada tipo de instrução S/370 I/O;

As fig. 45A-AG, informação de dados, de comandos e de estados sobre o endereço local e as linhas omnibus de dados na BCU durante as operações de transferência de dados dentro da BCU;

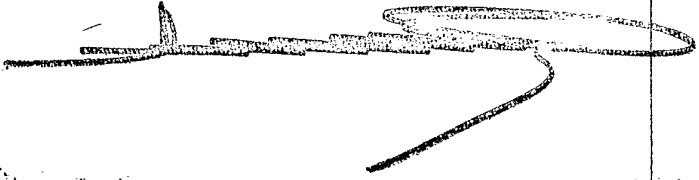
As fig. 46A-K, teoricamente, uma forma preferida do processo de emulação em disco por meio do qual o S/88 (através de BCU, ETIO e EXEC 370) armazena e busca informação num disco S/88 em formato S/370 em resposta a instrução S/370 I/O;

A fig. 47, teoricamente, o mapa da memória da fig. 10 juntamente com uma vista das entradas do mapa da memória S/88, algumas das quais são retiradas para se adaptar a uma área da memória S/370;

As fig. 48A-K, uma forma preferida da gestão das memórias virtual/física para o S/88 que podem interagir com sub-rotinas proporcionadas de novo durante o arranque do sistema e rotinas de reconfiguração para criar áreas de memória S/370 no interior da memória física S/88;

As fig. 49 e 50, esquemas parciais que ilustram uma certa parte da lógica usada para sincronizar pares de processadores S/370-S/99 e unidas associadas; e

As fig. 51 e 52, formas de realização



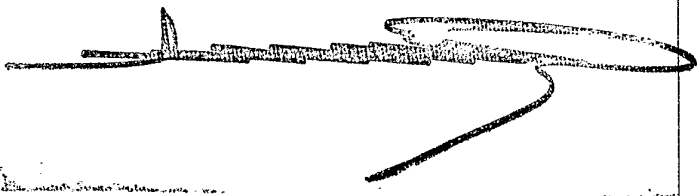
alternativas do aperfeiçoamento segundo a presente invenção.

Descrição da forma de realização preferida

Introdução

A forma de realização preferida para realizar praticamente a presente invenção compreende um sistema que tolera avarias. Os sistemas que toleram avarias têm tipicamente sido concebidos de alto a baixo para o funcionamento tolerante para as avarias. Os processadores, as memórias, os aparelhos de entrada/saída I/O e os sistemas operativos têm sido especificamente feitos por medida para proporcionar um ambiente tolerante para as avarias. Porém, a amplitude da sua base de clientes, a maturidade dos seus sistemas operativos, o número e a extensão dos programas de utilização disponíveis não são tão grandes como os dos sistemas de unidades centrais de processamento significativamente mais antigos de vários fabricantes, tais como o sistema 370 (S/370) comercializado pela International Business Machines.

Os sistemas de processamento de dados tolerantes para as avarias actuais oferecem muitas características avançadas que não estão normalmente disponíveis nos sistemas de unidades centrais de processamento tolerantes para as avarias mais antigas ou que não são suportadas pelos sistemas de unidades centrais de processamento. Algumas dessas características incluem: uma imagem única do sistema apresentada através de uma rede de computadores distribuída; a capacidade de processadores e controladores de I/O com fichas sob tensão (remoção e instalação de cartões com a energia ligada); detecção instantânea de erros, isolamento de avarias e remoção eléctrica para retirar de serviço componentes avariados sem interrupção da utilização do computador; unidades substituíveis pelo cliente identificadas por suporte de um serviço remoto; e reconfiguração dinâmica resultante de avaria de componentes ou adição de dispositivos complementares ao sistema, mantendo o sistema continuamente em funcionamento. Um exemplo de tais sistemas tole

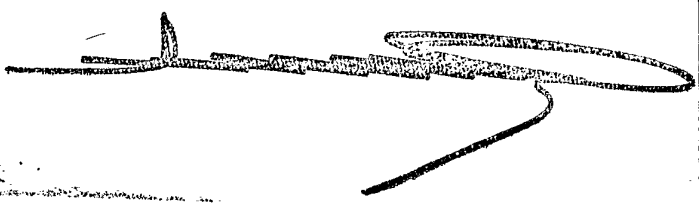


rantes para avarias é o sistema/88 (S/88) comercializado pela International Business Machines Co..

As propostas para incorporar as características atrás referidas no ambiente e na arquitectura S/370 consistiriam tipicamente numa importante alteração da redacção do ou dos sistemas operativos e dos programas de utilização e/ou de novos equipamentos materiais, desde a partida. Porém, a reformulação da escrita de um sistema operativo tal como VM, VSE, IX370, etc. é considerada por muitas pessoas como uma tarefa monumental, requerendo um grande número de programadores durante um período de tempo considerável. Usualmente, leva mais de cinco anos para amadurecer um sistema operativo complexo, tal como o IBM/S370 VM ou VMS. Até agora, a maioria das catástrofes de sistemas são consequência de erros do sistema operativo. Também são necessários vários anos para que os utilizadores criem a proficiência na utilização de um sistema operativo. Infelizmente, uma vez que um sistema operativo tenha amadurecido e tenha sido criada uma base de utilizadores ampla, não é uma tarefa simples modificar o código para introduzir novas funções, tais como a tolerância para as avarias, a imagem única do sistema e outras análogas.

Devido às complexidades e despesas para a migração de um sistema operativo maduro para o interior de uma nova arquitectura de uma máquina, os projectistas usualmente decidirão desenvolver um novo sistema operativo que pode não ser facilmente aceite pela comunidade dos utilizadores. Pode mostrar-se ser impraticável modificar o sistema operativo amadurecido para incorporar as novas características exemplificadas pelo sistema operativo desenvolvido de novo; porém, o novo sistema operativo pode nunca desenvolver uma base substancial de utilizadores e levará muitos anos de utilização de campo antes de estarem resolvidos a maioria dos problemas.

Por conseguinte, pretende-se que o presente aperfeiçoamento proporciona um ambiente e uma arquitectura tolerante para as avarias para um sistema de processamento e um sistema operativo normalmente não tolerante para as avarias, sem uma reformulação importante da escrita do sistema operativo.



Na forma de realização preferida, um modelo de sistema IBM S/88 é acoplado a um IBM S/370.

Um processo corrente de acoplamento de processadores e sistemas operativos diferentes é através de um certo tipo de controlador de comunicações adicionado a cada um dos sistemas, anexando actuadores de dispositivos aos sistemas operativos e usando um certo tipo de código de comunicação, tal como SNA (Systems Network Architecture) ou OSI para transportar dados. Normalmente, para obter a comunicação de dados entre computadores de nós terminais numa rede, é necessário que cada um dos nós terminais compreenda e aplique um conjunto consistente de serviços aos dados a permutar.

Para reduzir a complexidade da sua concepção, a maioria das redes são organizadas como uma série de camadas ou níveis, cada um deles construído sobre o seu predecessor. O número de camadas, o nome de cada camada e a função de cada camada diferem de rede para rede. Todavia, em todas as redes a finalidade de cada camada consiste em oferecer certos serviços às camadas mais elevadas, protegendo essas camadas dos pormenores de saber como são efectivamente realizados os serviços oferecidos. A camada n numa máquina realiza uma conversação com a camada n numa outra máquina. As regras e convenções usadas nesta conversação são colectivamente conhecidas como o protocolo da camada n. As entidades que compreendem as camadas correspondentes nas diferentes máquinas são designadas por processos emparelhados ("peer processes"), dizendo-se então que são os processos emparelhados que comunicam utilizando o protocolo.

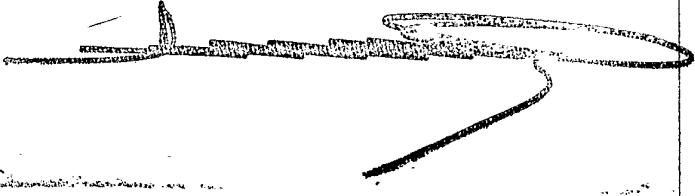
Na realidade, nenhum dado é directamente transferido da camada n numa máquina para a camada n numa outra máquina (excepto na camada mais baixa, ou camada física). Isto é, pode não haver qualquer acoplamento directo de programas de aplicação que operam nos sistemas distintos ou estranhos. Em vez disso, cada uma das camadas faz passar dados e informação de comando para a camada imediatamente abaixo de si, até ser atingida a camada mais baixa. Na camada mais baixa há comunicação física com a outra máquina, por oposição a comunicação virtual, usada pelas camadas mais elevadas.



Têm existido definições destes conjuntos de serviços num certo número de redes diferentes, como atrás se mencionou, e mais recentemente o interesse centrou-se na criação de protocolos para facilitar a comunicação de sistemas de fabricantes diferentes. Uma estrutura para o desenvolvimento destes protocolos é a estrutura definida pela ISO (International Standards Organization, modelo OSI (Open Systems Interconnect - Interligação de sistemas abertos), de sete camadas. Cada uma destas camadas neste sistema é responsável pelo fornecimento de serviços para a rede às camadas acima da camada em questão, enquanto requisita serviços à camada situada por baixo da mesma. Os serviços proporcionados em cada camada estão bem definidos, de modo que podem ser aplicados consistentemente por todas as estações na rede. Afirma-se que isso permite a interligação de equipamentos de fabricantes diferentes. A realização prática dos serviços de camada para camada no interior de um nó é específica dessa realização prática e permite a diferenciação dos fabricantes na base de serviços proporcionados no interior da estação.

É importante notar que toda a intenção de realizar praticamente um tal conjunto estruturado de protocolos consiste em efectuar a transferência de dados de extremo a extremo. As divisões principais dentro do modelo OSI podem compreender-se melhor se, como atrás se mencionou, compreendermos que o nó do utilizador está interessado no fornecimento de dados provenientes do programa de aplicação da fonte para o programa de aplicação do colector. Para fornecer estes dados, os protocolos OSI actuam nos dados em cada nível para fornecer tramas para a rede. As tramas são formadas como dados acoplados com preâmbulos correspondentes aplicados em cada nível OSI. Estas tramas são depois proporcionadas ao meio físico como um conjunto de bits que são transmitidos através do meio. Depois elas sofrem um conjunto inverso de procedimentos para proporcionar os dados ao programa de aplicação na estação de recepção.

Um processo corrente para o acoplamento dos processadores distintos e dos sistemas operativos distintos é através de um certo tipo de controlador de comunicações adicio



nado a cada sistema, anexando actuadores de dispositivos aos sistemas operativos e utilizando qualquer tipo de código de comunicações, tal como o SNA (Systems Network Architecture ou OSI (Open Systems Interconnect) para transportar os dados. A fig. 1 mostra uma interligação normalizada de dois sistemas computadores por meio de uma LAN (Local Área Network - Rede de área local). Em particular, está representado um sistema com a arquitectura IBM S/370 ligado a uma arquitectura do sistema IBM/88. Observar-se-á que em cada arquitectura um programa de aplicação opera através de uma interface com o sistema operativo para controlar um processador e aceder a um canal de I/O ou a uma linha omnibus. Cada dispositivo da arquitectura tem um controlador de comunicações para permuta de dados. Para comunicar, tem de utilizar-se um protocolo em várias camadas para permitir a permuta de dados entre os programas de aplicação correspondentes.

Um processo alternativo para permutar dados seria um processo de coprocessador no qual o coprocessador reside na linha omnibus do sistema, arbitra a linha omnibus do sistema e utiliza as mesmas I/O que o processador hospedeiro. O inconveniente do processo com coprocessador é a quantidade de remodelação da escrita do código necessária para suportar I/O não nativas (estranhas ao hospedeiro). Um outro inconveniente é que o utilizador tem de estar familiarizado com as arquitecturas de ambos os sistemas para fazer a comutação entre os sistemas operativos do coprocessador para o hospedeiro - um ambiente incómodo para o utilizador.

Um sistema de computador tolerante para as avarias da técnica anterior tem um módulo processador que contém uma unidade de processamento, uma unidade de memória de acesso aleatório, unidades de comando dos periféricos e uma estrutura única de linhas omnibus do módulo. A estrutura das linhas omnibus do sistema no interior de cada módulo processador inclui linhas omnibus correspondentes em duplicado, e cada unidade funcional no interior de um módulo processador tem também uma unidade correspondente em duplicado. A estrutura das linhas omnibus proporciona energia de operação às unidades de um módulo e sinais de distribuição de tempos do sistema provenientes



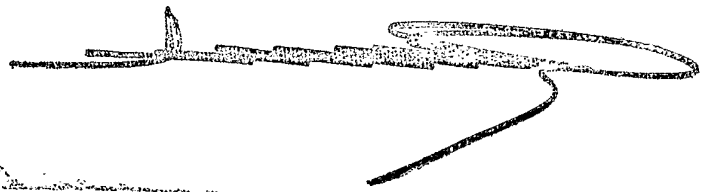
de um relógio principal.

A fig. 2 mostra, sob a forma de um diagrama funcional, a estrutura da parte da unidade do processador de um módulo processador. Utilizando processadores idênticos em parelhados montados num cartão comum de substituição e executando operações idênticas em sincronismo, podem fazer-se comparações para detectar erros de processamento. Cada cartão tem normalmente uma unidade correspondente redundante, de estrutura idêntica.

O sistema de computador proporciona a detecção de erros ao nível de cada unidade funcional no interior de todo o módulo do processador. Os detectores de erros seguem a evolução das operações nos equipamentos materiais no interior de cada unidade e verificam as transferências de informação entre unidades. A detecção de um erro faz com que o módulo do processador isole a unidade que provocou o erro e a interdicte de transferir informações para outras unidades, continuando o módulo o funcionamento utilizando a unidade parceira da unidade avariada.

Ao detectar-se um erro em qualquer unidade, essa unidade é isolada e colocada fora de linha, de modo que ela não possa transferir informação incorrecta para outras unidades. A parceira da unidade agora fora de linha continua a funcionar, permitindo assim que todo o módulo continue a funcionar. Um utilizador raramente se apercebe de uma tal detecção de um erro e da transição para o estado de fora de linha, excepto na visualização ou outra representação de um pedido de manutenção da unidade fora de linha. A disposição em cartões permite a desmontagem e a montagem fáceis.

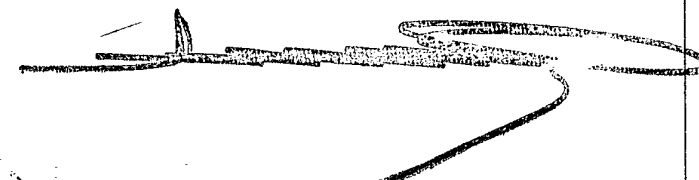
À unidade de memória está ainda atribuída a tarefa de verificar as linhas omnibus do sistema. Para isso, a unidade tem controladores de paridade que ensaiam os sinais de endereço e ensaiam os sinais de dados na estrutura. Após a determinação de que uma das linhas omnibus está avariada, a unidade de memória sinaliza para as outras unidades do módulo para que obedeçam apenas à linha omnibus não avariada. A unidade de alimentação do módulo processador utiliza duas fontes de



alimentação, cada uma das quais proporciona energia de operação a apenas uma unidade em cada par de unidades emparelhadas. Ao detectar-se uma tensão de alimentação errada, todas as linhas de saída vindas da unidade afectada para a estrutura das linhas omnibus são ligadas rigidamente ao potencial de terra para impedir que a avaria na alimentação provoque a transmissão de informação errada para a estrutura de linhas omnibus.

A fig. 3 mostra, sob a forma de diagrama funcional, a interligação de processadores S/370 emparelhados com processadores S/88 emparelhados, à maneira de uma estrutura tolerante para as avarias para permitir a permuta directa de dados. A semelhança com a estrutura S/88 anterior (fig. 2) é intencional mas é a única interligação por meio quer dos equipamentos físicos, quer pelos meios de programação que estabelece a operação da forma de realização preferida. Observar-se-á que os processadores S/370 estão acoplados a uma lógica de controlo da memória e a uma interface das linhas omnibus além da lógica de comparação do tipo S/88. Como será descrito, a lógica de comparação funcionará da mesma maneira que a lógica de comparação para os processadores S/88. Além disso, os processadores S/370 estão acoplados directamente e acoplados através das linhas omnibus do sistema a processadores S/88 correspondentes. Como para o processador S/88, os processadores S/370 estão acoplados aos pares, destinando-se os pares a ser montados em cartões de circuitos susceptíveis de ser substituídos no campo e ligados e desligados com a energia de alimentação ligada. As interligações pormenorizadas dos vários actuadores serão descritas com mais pormenor mais adiante.

A forma de realização preferida interliga vários processadores S/370 para executar as mesmas instruções S/370 simultaneamente sob o controlo de um sistema operativo S/370. Estes são acoplados a vários processadores S/88, aparelho de I/O e memória principal correspondentes, executando todos as mesmas instruções S/88, simultaneamente, sob o controlo de um sistema operativo S/88. Como será descrito mais adiante, incluem-se meios para desacoplar assincronamente os processadores S/88 dos seus aparelhos de I/O e da memória, para fazer pas



sar comandos de I/O S/370 e dados dos processadores S/370 para os processadores S/88 enquanto estes últimos estão desacoplados e para converter os comandos e os dados para uma forma utilizável pelos S/88 para processamento ulterior por processadores S/88 quando eles forem de novo acoplados aos seus aparelho de I/O e à memória principal.

1. Operação de um processador normalmente não tolerante para as avarias num ambiente tolerante às avarias

As características de tolerância às avarias atrás listadas são conseguidas numa forma de realização preferida acoplando processadores normalmente não tolerantes para as avarias, tais como processadores S/370 num primeiro par que executa as mesmas instruções S/370 simultaneamente sob o controlo dos sistemas operativos S/370. Proporcionam-se meios para comparar os estados de vários sinais num processador com os no outro processador para detectar erros instantaneamente num dos dois processadores.

Um segundo par de processadores S/370 correspondentes com meios de comparação é proporcionado para executar as mesmas instruções S/370 simultaneamente com o primeiro par e para detectar erros no segundo par. Cada processador S/370 está acoplado a um processador S/88 respectivo de um sistema tolerante às avarias tal como o sistema de processamento de dados S/88 tendo um primeiro e segundo pares de processadores parceiros, aparelhos de I/O S/88 e uma memória principal S/88. Cada processador S/88 tem com ele associado equipamento de meios físicos que o ligam com o aparelho de I/O e a memória principal.

Cada um dos processadores S/370 e S/88 respectivos tem as suas linhas omnibus dos processadores acopladas umas às outras por meios que incluem uma unidade de controlo das linhas omnibus. Cada unidade de controlo das linhas omnibus inclui meios que interagem com um programa de aplicação que é executado no processador S/88 respectivo para desacoplar assincronamente o processador S/88 dos seus equipamentos físicos e para o acoplar à unidade de controlo das linhas omnibus 1) pa

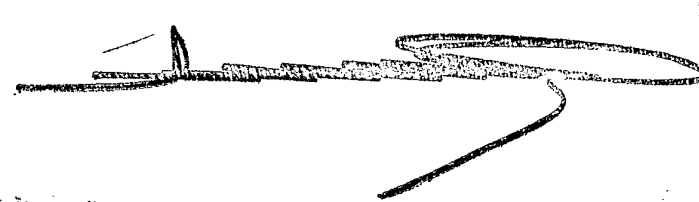


ra a transferência dos comandos S/370 e dados do processador S/370 para o processador S/88 e 2) para a conversão dos comandos e dados S/370 em comandos susceptíveis de ser executados e de dados susceptíveis de ser usados pelo S/88.

O sistema de processamento de dados S/88 processa depois os comandos e os dados sob o controlo do sistema operativo S/88. O sistema de processamento de dados S/88 também responde a sinais de erro em um qualquer dos pares de processadores S/370, ou no seu par de processadores S/88, respectivamente, acoplados para retirar os pares acoplados de serviço e permitir o funcionamento tolerante às avarias com os outros pares S/370, S/88 acoplados. Com esta disposição, executam-se programas S/370 pelos processadores S/370 (com a ajuda do sistema S/88 para as operações de I/O) num ambiente tolerante às avarias (FT) com as características vantajosas do S/88, tudo sem alterações significativas dos sistemas operativos S/370 e S/88.

Além disso, a unidade de gestão da memória do S/88 é controlada de modo a atribuir áreas dedicadas na memória principal S/88 a cada um dos pares de processadores duplexados S/370, e o sistema operativo, sem conhecimento por parte do sistema operativo S/88. Os processadores dos pares de processadores S/370 duplexados são acoplados individualmente à estrutura das linhas omnibus comuns do S/88 através de um aparelho gestor da memória e da interface das linhas omnibus S/68 para a busca e o armazenamento de instruções e dados S/370 a partir das respectivas áreas de memória dedicadas.

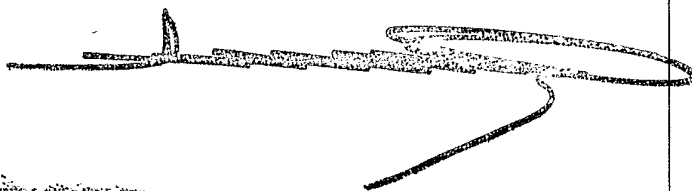
A forma de realização preferida proporciona um processo e meios para a realização da tolerância das avarias no equipamento físico S/370 sem reelaboração da escrita do sistema operativo S/370 ou das aplicações S/370. Proporciona-se uma redundância completa do equipamento físico CPU S/370 e a sincronização sem que o cliente conceba um processador para suportar a tolerância de avarias. Um sistema operativo S/370 e um sistema operativo tolerante das avarias (ambos sistemas de memória virtual) são executados simultaneamente sem uma reelaboração importante da escrita de qualquer dos sistemas operativos.



Proporciona-se uma interface de equipamentos físicos/microcódigo, na forma de realização preferida, entre os pares de processadores emparelhados, executando cada processador um sistema operativo diferente. Um processador é uma máquina IBM S/370 controlada por um microcódigo, que executa um sistema operativo IBM (por exemplo VM, VSE, IX370, etc.. O segundo processador da forma de realização preferida é uma máquina tolerante das avarias do equipamento físico que executa um sistema operativo capaz de controlar um ambiente tolerante das avarias do equipamento físico (por exemplo o sistema IBM S/88, que executa o S/88 VOS (sistema operativo virtual).

A interface do equipamento físico/microcódigo entre os pares de processadores permite que os dois sistemas operativos coexistam num ambiente observado pelo utilizador como um ambiente de um sistema único. Os recursos de equipamentos físicos/microcódigo (memória, linhas omnibus do sistema I/O para disco, fita, terminais I/O de comunicações, alimentação de energia e invólucros) actuam independentemente uns dos outros, enquanto cada sistema operativo trata a sua parte do funcionamento do sistema. Utilizam-se aqui intermutavelmente os termos memória e armazenamento. O ou os processadores tolerantes das avarias (FT) e o sistema operativo gerem a detecção/localização e recuperação das avarias. O ou os processadores não tolerantes das avarias (NFT) executam as instruções nativas sem qualquer conhecimento do processador FT. O processador FT aparece ao processador NFT como canais de I/O múltiplos.

A interface equipamento físico/microcódigo permite que ambos os processadores da memória virtual partilhem uma memória tolerante das avarias comum. Um bloco de memória contínuo a partir da tabela de atribuição da memória do processador FT é atribuído a cada processador NFT. A característica da translação dinâmica de endereços do processador NFT controla o bloco de memória que foi atribuído ao mesmo pelo processador FT. O processador NFT percebe que a sua memória começa no endereço zero através da utilização de um registador de desvios. Efectua-se uma verificação de limites para manter o processador NFT nos seus limites na memória. O processador FT pode

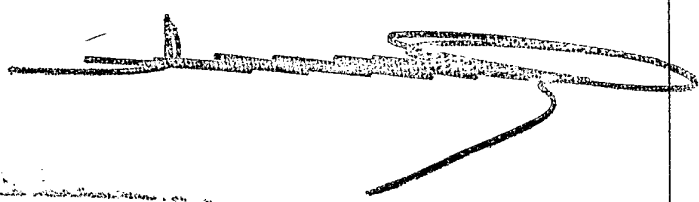


aceder à memória NFT e aos blocos DMA I/O de dados de entrada ou de saída do espaço de endereços NFT, enquanto que o processador NFT está impedido de aceder à memória fora do espaço de endereços que lhe está atribuído. A dimensão da memória NFT pode ser alterada por alteração da tabela de configuração.

2. Desacoplamento de um processador do seu equipamento físico associado para apresentar comandos e dados provenientes de um outro processador a si próprio

A adição de um novo dispositivo a um processador e sistema operativo existentes em geral exige a ligação de equipamentos físicos através de uma linha omnibus ou um canal, e a escrita de novos meios de programação de actuação do novo dispositivo ao sistema operativo. O "desacoplamento" aperfeiçoado permite que dois processadores distintos comuniquem um com o outro sem ligar um dos processadores a uma linha omnibus ou a um canal e sem arbitragem do controlo da linha omnibus. Os processadores comunicam sem modificação significativa do sistema operativo e sem a exigência do actuator tradicional do dispositivo. Pode dar-se a um utilizador a imagem de um sistema único quando se fundem dois processadores distintos e diferentes, embora cada processador esteja a executar o seu próprio sistema operativo nativo.

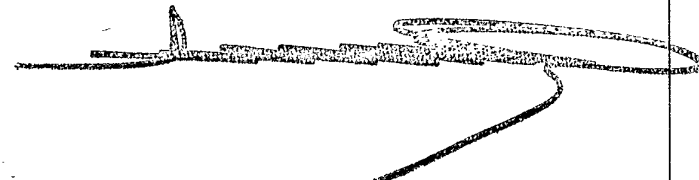
Esta característica proporciona um processo e meios de combinação das características especiais apresentadas por um sistema operativo de desenvolvimento mais recente com a visão dos utilizadores e a fiabilidade de um sistema operativo maduro. Ele faz o acoplamento dos dois sistemas (equipamentos físicos e meios de programação) entre si para formar um terceiro sistema novo. Será evidente para os entendidos na matéria que, embora a forma de realização preferida apresente um sistema S/370 acoplado a um sistema S/88, poderiam acoplar-se quaisquer outros dois sistemas distintos. Os critérios da concepção desta ideia são: alteração pequena ou nula do sistema operativo maduro, de modo que ele mantenha a sua fiabilidade, e um impacto mínimo no sistema operativo desenvolvido mais recentemente, devido ao tempo de desenvolvimento do código.



Esta característica implica um processo de combinação de dois sistemas diferentes, cada um deles com as suas características para obter um terceiro sistema com características de ambos. Uma forma preferida do processo exige uma lógica de acoplamento entre os sistemas que funcione preponderantemente como um controlador de acesso directo à memória (DMAC). O objectivo principal desta característica é o de proporcionar um programa de aplicação que passa num processador tolerante das avarias (por exemplo o S/88, na forma de realização preferida) e assente no sistema operativo tolerante das avarias, um processo de obtenção de dados e comandos provenientes de um processador estranho (por exemplo o S/370 na forma de realização preferida) e o seu sistema operativo. Existem em qualquer dos processadores mecanismos de defesa tanto do equipamento físico como dos meios de programação, para impedir a intrusão (isto é, um supervisor relativamente ao estado do utilizador, verificação do mapa da memória, etc.). Tipicamente, os sistemas operativos tendem a controlar todos os recursos do sistema, tais como interrupções, canais DMA, e dispositivos de I/O e controladores. Por conseguinte, acoplar duas arquitecturas diferentes e transferir comandos e dados entre estas máquinas sem ter projectado esta função desde a base é por muitos considerada uma tarefa monumental e/ou impraticável.

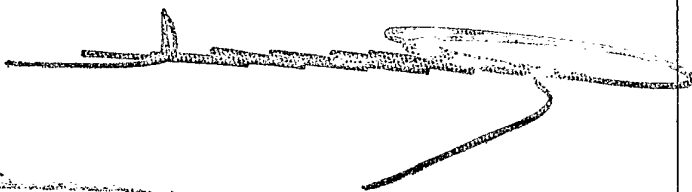
A fig. 4 representa esquematicamente um processador S/370 acoplado a um processador S/88 no ambiente da forma de realização preferida. Em contraste com o processador S/370 representado na fig. 1, a memória foi substituída por uma lógica de interface da linha omnibus S/88 e o processador de canais S/370 foi substituído por um adaptador de linhas omnibus e uma unidade de controlo da linha omnibus. Dá-se particular atenção à interligação entre a unidade de controlo da linha bus S/370 e o processador S/88, representada por uma linha dupla de traços e dois pontos.

Esta característica implica a ligação da lógica de acoplamento dos processadores à linha omnibus de endereços virtuais do processador tolerante das avarias S/88, bem como à linha omnibus de dados, à linha omnibus de controlo



ou ao canal de controlo, visto que estão ligados mais dispositivos. A linha de transmissão de impulsos que indica que um endereço válido está na linha omnibus de endereços virtuais do processador tolerante das avarias é activada alguns nanossegundos depois de serem activados os sinais de endereço. A lógica de acoplamento compreende o adaptador de linhas omnibus e a unidade de controlo das linhas omnibus determina se uma faixa pré-seleccionada de endereços é apresentada por um programa de aplicação antes de aparecer o sinal na linha de transmissão de impulsos referida. Se for detectada esta faixa de endereços, o sinal de impulsos do endereço é bloqueado e impedido de ir para os equipamentos físicos do processador tolerante de avarias S/88. Este sinal em falta impede que o equipamento físico e o sistema operativo tolerante de avarias tome conhecimento de um ciclo da máquina. A lógica de verificação tolerante de avarias no equipamento físico é isolada durante este ciclo e é omitida qualquer actividade que ocorra durante esse tempo. Toda a lógica de atribuição de posições de endereços virtuais na memória "cache" e processadores em vírgula flutuante nas linhas omnibus do processador deixarão de reconhecer que se verificou um ciclo da máquina. Isto é, são "congeladas" todas as funções da CPU do S/88, aguardando a asserção do sinal ADDRESS STROBE pelo processador S/88.

Este sinal ADDRESS STROBE ou sinal do impulso do endereço que foi bloqueado a partir da lógica do processador tolerante de avarias é enviado para a lógica de acoplamento. Isso dá ao processador tolerante de avarias S/88 o controlo completo sobre a lógica de acoplamento que é a interface entre o programa de aplicação especial tolerante de avarias e o processador S/370 associado. O sinal do impulso de endereço e o endereço virtual são usados para seleccionar a memória, os registadores e o DMAC locais que são componentes da lógica de acoplamento. A fig. 5 representa esquematicamente o resultado da detecção de uma interrupção proveniente da lógica de controlo das linhas omnibus S/370, que se determina ser um nível apropriado e correspondendo a um endereço apropriado. Portanto, no seu aspecto mais lato, o mecanismo de desacoplamento desliga um



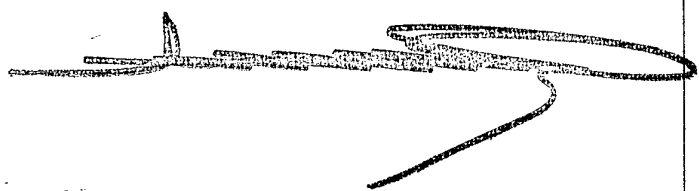
processador dos seus equipamentos físicos associados e liga o processador a uma entidade estranha para a transferência eficiente de dados para essa entidade.

A lógica de acoplamento tem uma memória local que é usada para formar uma fila de espera dos comandos S/370 recebidos e armazenar dados que vão para e vêm do S/370. Os dados e os comandos são transportados para a memória local por vários canais DMA na lógica de acoplamento. O programa de aplicação tolerante de avarias inicializa o DMCA e trata as interrupções provenientes do DMAC, o que serve para notificar o programa de utilização quando tiver chegado um comando ou tiver sido recebido ou emitido um bloco de dados. Para completar uma operação, a lógica de acoplamento tem de fazer o retorno de linhas de confirmação dos impulsos de dados antes da frente do impulso de relógio do processador para garantir que ambos os lados do processador tolerante de avarias se mantêm em sincronismo.

O programa de aplicação recebe comandos do tipo dos canais S/370, tais como "Start I/O", "Test I/O", etc.. O programa de aplicação converte depois cada comando de I/O S(370 num comando de I/O tolerante de avarias e inicia uma sequência de comando de I/O tolerante de avarias normal.

Pensa-se ser este um processo novo para fazer um bloco de dados contornar um sistema operativo para uma aplicação. É também uma maneira de permitir que uma aplicação trate uma interrupção, o que é uma função usualmente realizada por um sistema operativo. O programa de aplicação pode comutar o processador tolerante de avarias da sua função normal de processador para a função de controlador de I/O, à vontade, e numa base por ciclo, precisamente pelo endereço virtual que ele selecciona.

Assim, dois sistemas de processamento de dados com arquitecturas diferentes de instruções e de endereçamento da memória estão estreitamente acopladas de modo a permitir a um sistema aceder efectivamente a qualquer parte do espaço da memória virtual do outro sistema sem o outro sistema tomar conhecimento da existência do primeiro. O código de aplica-



ção especial no outro sistema comunica com o primeiro sistema através do equipamento físico, colocando endereços especiais na linha omnibus. O equipamento físico determina se o endereço é um endereço especial. Se assim for, os impulsos de relógio são bloqueados para não serem detectados pelos circuitos do outro sistema e re-orientados de modo tal que a CPU do outro sistema possa controlar equipamento físico especial e um espaço de memória acessível a ambos os sistemas.

O outro sistema pode controlar completamente o primeiro sistema, quando necessário, por exemplo para tarefas de inicialização e de configuração. O primeiro sistema não pode, de modo algum, controlar o outro sistema, mas pode apresentar pedidos de serviço ao outro sistema, da maneira seguinte:

O primeiro sistema apresenta comandos de I/O e/ou dados no formato do primeiro sistema num espaço de memória de acesso comum e, utilizando equipamento físico especial, apresenta uma interrupção ao outro sistema a um nível especial, pondo em acção o programa de aplicação especial.

Este último é dirigido para o espaço da memória que contém a informação apresentada e processa a mesma para converter o seu formato para a forma nativa do outro sistema. Depois o programa de aplicação dirige o sistema operativo nativo do outro sistema para executar operações de I/O nativas nos comandos e dados convertidos. Assim, tudo o referido anteriormente ocorre de maneira completamente transparente para os sistemas operativos nativos dos dois sistemas e sem qualquer alteração significativa nesses sistemas operativos.

3. Apresentação de interrupções a um sistema transparentes para o sistema operativo

Os programas mais correntes são executados em um de dois (ou mais) estados, o estado supervisor e o estado utilizador. Os programas de aplicação são passados no estado do utilizador, e funções tais como as interrupções no estado de supervisor.

Uma aplicação ocupa uma porta de I/O,



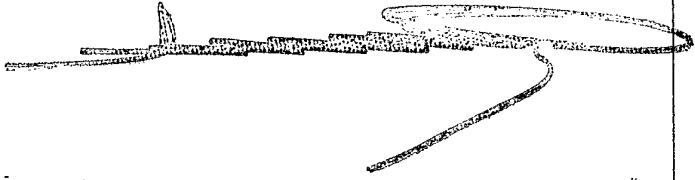
depois abre a porta, envia um pedido de I/O sob a forma de uma leitura, de uma escrita ou de um controlo. Nesse instante, o processador ocupará um comutador de tarefa. Quando o sistema operativo receber uma interrupção significando o fim de uma I/O, o sistema operativo colocará esta informação numa fila de espera disponível, ordenada por prioridade, para os recursos do sistema.

O sistema operativo reserva todos os vectores de interrupção para o seu próprio uso, não ficando nenhum disponível para novas características tais como uma interrupção exterior significando um pedido de I/O de uma outra máquina.

No S/88 da forma de realização preferida, a maioria dos vectores de interrupção disponíveis estão actualmente inutilizados, sendo ajustados para provocar a transferência para um processador de erros comum para interrupções "não inicializados" ou "espúrias", prática comum nos sistemas operativos. A forma de realização preferida do presente aperfeiçoamento substitui um subconjunto destes vectores, de outro modo não utilizados, por vectores apropriados para processadores de interrupções especiais para interrupções da lógica de acoplamento S/370. O sistema operativo S/88 modificado é depois de novo unido para ser usado com os vectores de novo integrados no seu lugar.

O sistema/88 da forma de realização preferida tem oito níveis de interrupção e usa auto-vectores em todos os níveis excepto no nível 4. O aperfeiçoamento segundo a presente invenção utiliza um destes níveis de auto-vectores, o nível 6, que tem a prioridade a seguir à prioridade mais elevada. Este nível (6) é normalmente usado pelo sistema/88 para interrupções de perturbações da alimentação de corrente alternada.

A lógica que faz o acoplamento do S/370 ao S/88 apresenta interrupções ao nível (6) combinando num circuito OU os pedidos de interrupção com os das perturbações da corrente alternada de alimentação. Durante a inicialização do sistema, números apropriados de vectores para os processadores de interrupções especiais para as interrupções da lógica de acoplamento



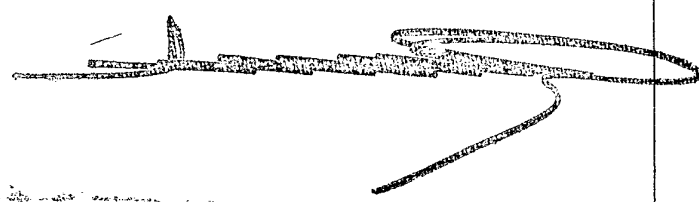
plamento (alguns, por exemplo, em registadores DMAC) por um programa de aplicação, transparente para o sistema operativo S/88.

Quando qualquer interrupção for recebida pelo S/88, ele inicia uma confirmação de interrupção (IACK), utilizando apenas equipamento físico e operações internas do processador S/88 para processar a interrupção e fazer a busca da primeira instrução do processador de interrupções. Não é necessária a execução de qualquer programa. Porém, o número do vector tem também de ser obtido e apresentado de uma maneira transparente. Isso consegue-se na forma de realização preferida desacoplando o processador S/88 do seu equipamento físico associado (incluindo o mecanismo de apresentação de interrupções de perturbações da alimentação de corrente alternada) e acoplando o processador S/88 à lógica de acoplamento S/370-S/88 quando for apresentada uma interrupção do nível 6 pela lógica de acoplamento.

Mais especificamente, o processador S/88 coloca o código da função e o nível de interrupção nas suas saídas e também declara ADDRESS STROBE (AS) e DATA STROBE (DS) no início do ciclo IACK. O ADDRESS STROBE é bloqueado a partir do equipamento físico S/88, incluindo o mecanismo de interrupção por perturbações da alimentação de corrente alternada, se estiver activo o sinal de apresentação de interrupção da lógica de acoplamento, sendo enviado AS para a lógica de acoplamento para ler o número de vector apropriado, que é encaminhado por uma porta para o processador S/88 pelo DATA STROBE. Devido ao facto de o DATA STROBE estar bloqueado pelo equipamento físico S/88, o ciclo de máquina (IACK) é transparente para o sistema operativo S/88 relativamente à obtenção no número de vector da interrupção da lógica de acoplamento.

Se o sinal de interrupção da lógica de acoplamento não estivesse activo no início do ciclo IACK teria sido considerada uma interrupção do nível a S/88 normal.

4. Partilha de uma memória real entre dois ou mais processadores que executam sistemas operativos de memória virtual diferente.

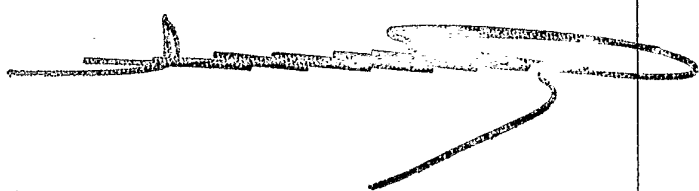


Esta característica faz o acoplamento de um sistema tolerante das avarias a um processador e o sistema operativo estranhos que não têm código para suportar uma memória tolerante das avarias, isto é, um código para remover e inserir cartões de memória através de uma ligação com a alimentação ligada, detecção instantânea de dados corrompidos e a sua recuperação se for apropriado, etc..

Esta característica proporciona um processo e meios pelos quais pode fazer-se que dois ou mais processadores que executam sistemas operativos virtuais diferentes compartilhem uma memória real única, de uma maneira transparente para os dois sistemas operativos e nos quais um processador pode aceder ao espaço de memória do outro processador de modo que possam verificar-se transferências entre estes múltiplos processadores.

Esta característica combina dois ambientes de sistemas operativos aparentes para o utilizador para dar a aparência ao utilizador de um sistema operativo único. Cada um dos sistemas operativos é um sistema operativo virtual que normalmente controla o seu espaço completo da memória real. A presente invenção tem apenas um único espaço de memória real que é compartilhado pelos dois processadores através de uma linha omnibus dos sistemas comum. Nenhum dos sistemas operativos é substancialmente re-escrito e nenhum dos sistemas operativos sabe que o outro existe. Esta característica utiliza um programa de aplicação que é executado num primeiro processador para fazer a busca na fila de espera de atribuição de memória do primeiro sistema operativo. Quando se tiver encontrado um espaço de memória contíguo, suficientemente grande para satisfazer as necessidades do segundo sistema operativo, este espaço de memória é então retirado, por indicadores de manipulação, da tabela de atribuição da memória do primeiro sistema operativo. O primeiro sistema operativo deixa de utilizar (por exemplo a capacidade de sobre-atribuir) esta memória retirada, a menos que a aplicação devolva a memória para o primeiro sistema operativo.

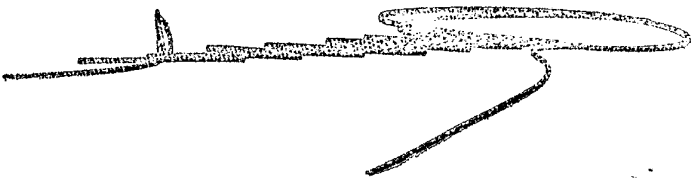
O primeiro sistema operativo é subserviente relativamente ao segundo sistema operativo numa prespecti-



va de I/O e responde ao segundo sistema operativo como um controlador de I/O. O primeiro sistema operativo é o dono de todos os recursos do sistema e na forma de realização preferida é um sistema operativo tolerante de avarias. O primeiro sistema operativo inicialmente atribui e retira a atribuição de memória (excepto a memória que foi "roubada" para o segundo sistema operativo) e processa todas as falhas de equipamento físico associadas e a recuperação. O objectivo é combinar os dois sistemas operativos sem alterar o código do sistema operativo em grau importante. Cada um dos sistemas operativos tem de acreditar que está a controlar toda a memória do sistema, visto que é um recurso único que está a ser usada pelos dois processadores.

Quando se liga a energia de alimentação ao sistema, o primeiro sistema operativo e o seu processador assumem o controlo do sistema, mantendo o equipamento físico o segundo processador numa condição de reposição. O primeiro sistema operativo inicializa o sistema e determina a quantidade de memória que realmente existe. O sistema operativo eventualmente organiza toda a memória em blocos de 4KB (4096 octetos) e lista cada bloco disponível numa fila de espera de atribuição de memória. Cada bloco de 4KB listado na fila de espera aponta para o bloco 4KB disponível seguinte. Qualquer memória usada pelo primeiro sistema ou é removida ou é adicionada em blocos de 4KB do cimo da fila de espera, ajustando-se adequadamente os indicadores dos blocos. Quando os utilizadores pedirem espaço de memória a partir do sistema operativo, os pedidos são satisfeitos atribuindo, a partir da fila de espera, um número necessário de blocos de 4KB da memória real. Quando a memória deixar de ser necessária, os blocos serão repostos na fila de espera.

A seguir, o primeiro sistema operativo executa uma lista de funções denominada arranque dos módulos, que configura o sistema. Uma aplicação que é executada pelo arranque dos módulos é uma nova aplicação usada para capturar memória do primeiro sistema operativo e atribuir a memória ao segundo sistema operativo. Este programa explora a lista completa de atribuições da memória e encontra uma cadeia contígua de blocos de memória de 4KB. Então, o programa de aplicação altera

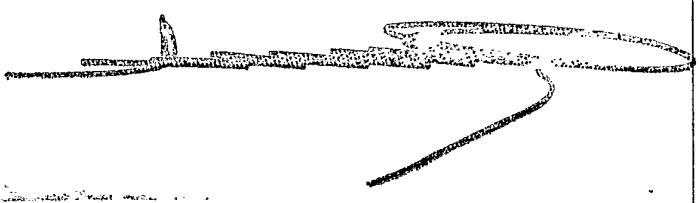


os indicadores na parte da fila de espera correspondente à cadeia contígua de blocos, removendo desse modo um bloco contíguo de memória da lista de atribuições de memória do primeiro sistema operativo. Na forma de realização preferida, o indicador do bloco de 4KB anterior ao primeiro bloco de 4KB retirado é modificado para apontar para o bloco de 4KB imediatamente seguinte à cadeia contígua de blocos retirada.

Nesta altura, o primeiro sistema operativo não tem controlo nem conhecimento deste espaço da memória real, a menos que o sistema seja de novo inicializado ou a aplicação reponha os indicadores da memória. É como se o primeiro sistema operativo considere um segmento da memória real atribuído a um processo que é executado em si próprio e que não pode ser de novo atribuído porque os blocos foram retirados da tabela, não simplesmente atribuídos a um utilizador.

O espaço de endereços removido é então transmitido para o segundo sistema operativo. Há uma lógica de desvio de equipamento físico que faz com que o bloco de endereços, roubado do primeiro sistema operativo e dado ao segundo sistema operativo, aparente começar no endereço zero para o segundo sistema operativo. O segundo sistema operativo controla então a memória roubada do primeiro sistema operativo como se ele fosse a sua própria memória real, e controla a memória através do seu próprio gestor da memória virtual, isto é, traduz endereços virtuais emitidos pelo segundo sistema operativo em endereços reais no interior do espaço de endereços da memória real atribuídos.

Um programa de aplicação que é executado no primeiro sistema operativo pode deslocar dados de I/O para e do espaço da memória do segundo processador, mas o segundo processador não pode ler nem escrever fora do seu espaço atribuído porque o segundo sistema operativo não tem conhecimento da memória adicional. Se se verificar um mau funcionamento do sistema operativo, no segundo sistema operativo, um dispositivo de rejeição no equipamento físico impedirá o segundo sistema operativo de inadvertidamente escrever no espaço do primeiro sistema operativo.



A quantidade de espaço de memória atribuída ao segundo sistema operativo é definida numa tabela no programa de arranque dos módulos pelo utilizador. Se o utilizador desejar que o segundo processador tenha 16 MB, então ele de finirá isso na tabela do programa de arranque de módulos e a aplicação adquirirá esse espaço a partir do primeiro sistema operativo. Um SVC ("service call" - chamada de serviço) permite ao programa de aplicação ter acesso à zona de supervisão do primeiro sistema operativo, de modo que os indicadores podem ser modificados.

Uma razão importante por que é desejável que os dois sistemas operativos partilhem a mesma memória é que a memória é tolerante de avarias no primeiro processador; e o segundo processador pode utilizar a memória e a I/O tolerante de avarias do primeiro processador. O segundo processador é feito para ser tolerante de avarias por duplicação de certo equipamento físico e por comparação de certos endereços, dados e linhas de controlo. Utilizando estas técnicas, o segundo processador é de facto uma máquina tolerante de avarias, embora o segundo sistema operativo não tenha capacidade para tolerância de avarias. Podem acoplar-se mais de um processador e um sistema operativo do segundo tipo estranhos ao primeiro sistema operativo com uma área de memória real separada proporcionada para cada processador estranho.

Na forma de realização preferida, o primeiro sistema operativo é o do S/88 tolerante das avarias e o segundo sistema operativo é um dos sistemas operativos S/370 e os primeiro e segundo processadores são, respectivamente o S/88 e o S/370. Esta característica não só permite que um sistema normalmente tolerante de avarias utilize uma memória tolerante de avarias que é mantida por um sistema tolerante de avarias, como também permite que o sistema não tolerante de avarias: 1) partilhe o acesso aos aparelhos de I/O tolerantes de avarias mantidos pelo sistema tolerante de avarias e 2) permuta dados entre os sistemas de uma maneira mais eficiente sem atrasos significativos de um acoplamento canal-a-canal.

•
•
•



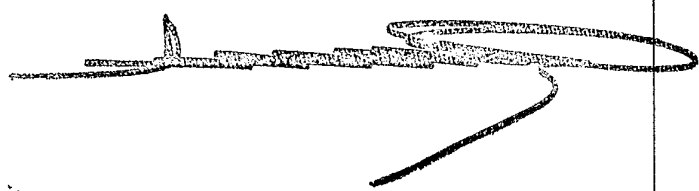
5. Imagem de um sistema único

A expressão "imagem de um sistema único" é utilizada para caracterizar as redes de computadores nas quais o acesso a dados e recursos (por exemplo impressores, ficheiros em suporte permanente, etc.) remotos aparece ao utilizador como sendo igual ao acesso a dados e recursos no terminal local a que está ligado o teclado do utilizador. Assim, o utilizador pode aceder a um ficheiro de dados ou a recursos simplesmente pelo nome, sem ter de conhecer a localização física na rede.

Introduz-se aqui o conceito de "imagem derivada de um sistema único" como uma nova expressão, que se pretende que seja aplicada a elementos do computador de uma rede que não têm meios para se ligar directamente a uma rede que tem a imagem de um sistema único, mas utiliza recursos de equipamento físico e de meios de programação dessa rede para se ligar directamente à mesma com uma imagem efectiva de um sistema único,

Para os fins desta discussão, a ligação directa de um sistema de computador para desenvolver efeitos de "imagem derivada de um sistema único" pode ser efectuada com vários graus de acoplamento entre esse sistema e elementos da rede. O termo "acoplamento frouxo", como aqui é usado, significa um acoplamento efectuado através de canais de I/O do computador de derivação e o computador "nativo" que faz parte da rede. "Acoplamento apertado" é um termo aqui usado para descrever uma relação entre os computadores de derivação e nativo estabelecida através de meios físicos especiais que permitem comunicar um com o outro numa base directa (isto é, sem utilizar os canais de I/O existentes de qualquer deles).

Um tipo especial de acoplamento apertado aqui contemplado, designado por "acoplamento apertado transparente" implica a adaptação do equipamento físico de acoplamento para permitir que cada computador (os computadores de derivação e nativo) utilize recursos do outro computador de modo tal que o sistema operativo de cada computador não tenha conhecimento dessa utilização. O acoplamento apertado transparente, como

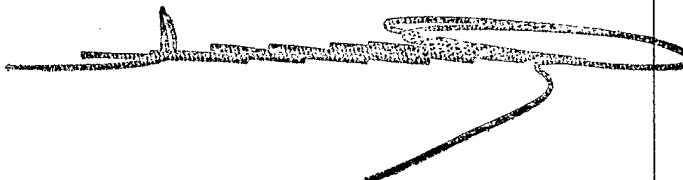


se acaba de definir, constitui uma base para conseguir vantagens de custo e eficácia na rede acoplada. O custo dos equipamentos físicos de acoplamento, não obstante a complexidade da sua concepção, devem ser mais que compensados pelas economias feitas por se evitarem modificações extensas dos meios de programação do sistema operativo que de outro modo seriam necessárias. As vantagens de maior eficácia resultam das ligações mais rápidas devido ao acoplamento directo e à interferência reduzida de larguras de banda na interface de acoplamento.

O termo "rede", como é usado nesta secção, é mais restrito do que o conceito actualmente prevalente de uma rede que é um esquema de ligações mais alto internacional de teleprocessamento/satélites, ao qual podem ligar-se tipos diferentes de máquinas se estiverem em conformidade com algum protocolo específico. Em vez disso, "rede" utiliza-se nesta secção aplicada a um complexo ligado de processadores S/88 ou, em alternativa, a um complexo ligado de outros processadores com características de imagem de um sistema único.

Utilizar-se-ão vários termos cuidadosamente definidos para explicar mais o conceito de uma imagem de um sistema único, como é aqui considerado, e partir-se-á da hipótese de que a forma de realização preferida específica do aperfeiçoamento será usada como base para a clarificação:

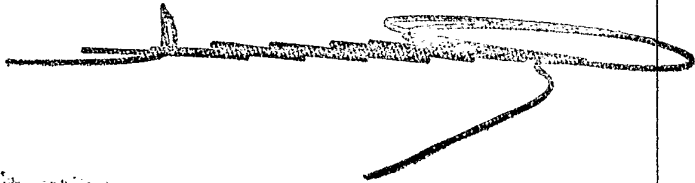
- a. HSDI ("High Speed Data Interconnection"- Interligação de dados a alta velocidade) refere-se a um subsistema de equipamentos físicos (e cabos) para a transferência de dados entre unidades de equipamento físico separadas.
- b. Elo de ligação ("link") refere-se a uma construção ou objecto de meios de programação que consiste inteiramente em indicadores multi-partes para um outro objecto de meios de programação e que tem muito do carácter de um nome suposto.
- c. MÓDULO refere-se a uma unidade de processamento auto-sustentada constituída pelo menos por um de cada um dos elementos: invólucro, alimentação de energia, CPU, memória e dispositivo de I/O. Um MÓDULO pode ser ampliado fixando entre si vários invólucros para alojar dispositivos periféricos adicionais, criando um módulo único maior. Algumas unidades de



I/O (terminais, impressoras) podem ser exteriores e ligadas ao invólucro por cabos; são consideradas como fazendo parte do módulo único. Um MÓDULO pode ter apenas um complexo de CPU.

- d. COMPLEXO de CPU refere-se a um ou mais cartões simples ou duplos de processadores no interior do mesmo invólucro, gerido e controlado por meios de programação do sistema operativo para operar como uma CPU única. Independentemente do número actual de cartões de processador instalados, qualquer programa ou aplicação de utilizador é escrito e executado como se estivesse presente apenas uma CPU. A carga de trabalho de processamento é partilhada grosseiramente entre os cartões CPU disponíveis, podendo executar-se simultaneamente múltiplas tarefas, mas sendo cada programa de aplicação apresentado com uma "IMAGEM DE CPU ÚNICA".
- e. OBJECTO refere-se a uma colecção de dados (incluindo programas executáveis) armazenados no sistema (disco, fita), que pode ser identificada univocamente por um nome hierárquico. Um ELO é um indicador designado de maneira única para algum outro objecto, sendo assim ele próprio um OBJECTO. Uma PORTA de I/O é uma construção de meios de programação designada de maneira única que aponta para um dispositivo I/O específico (uma fonte ou um destino de dados), sendo assim também um OBJECTO. O sistema operativo impede efectivamente a duplicação de NOMES DE OBJECTOS.

Visto que a expressão "imagem de sistema único" não é usada consistentemente na literatura, será descrita com mais pormenor para clarificação do presente aperfeiçoamento de uma "imagem de sistema único derivado". Na definição e descrição da expressão IMAGEM DE SISTEMA ÚNICO, a "imagem" refere-se ao ponto de vista, do programa de aplicação, do sistema e do ambiente. "Sistema", neste contexto, significa os equipamentos físicos (complexo de CPU) e os meios de programação (sistema operativo e os seus serviços) combinados, para os quais o programador da aplicação dirige as suas instruções. "Ambiente" significa todos os dispositivos de I/O e outros meios ligados que são susceptíveis de ser endereçados pelo sistema operativo



e desse modo acessíveis ao programador, mediante pedidos de serviços ao sistema operativo.

Um computador independente verdadeira-mente único com o seu sistema operativo tem então de proporcionar uma IMAGEM DE SISTEMA ÚNICO ao programador. É apenas quando ele deseja ligar sistemas múltiplos entre si, a fim de partilhar dispositivo de I/O e distribuir processamento, que esta "imagem" vista pelo programador começa a modificar-se; a interligação ordinária de duas máquinas por via de linhas de teleprocessamento (ou mesmo por cabos) força o programador a compreender - e a aprender a lidar com - o ambiente dual, para tirar proveito das facilidades alargadas.

Geralmente, para ter acesso a facilidades no outro ambiente ele tem de pedir o seu sistema operativo local para comunicar os seus requisitos em pormenor. Tem então se poder aceitar os resultados do seu pedido de maneira assíncrona (e na sequência própria) depois de uma demora arbitrariamente longa. O tratamento e o controlo das múltiplas mensagens e transferências de dados entre máquinas constituem um dispêndio geral significativo de processamento em ambas as máquinas; pode ser pesado, ineficiente e difícil para o programador num tal ambiente de DUAL-SYSTEM. E quando aumentar o número de máquinas convencionalmente ligadas, pode aumentar rapidamente a complexidade para o programador.

O projecto original do sistema/88 incluía os meios para simplificar esta situação e proporcionar ao programador a IMAGEM DE SISTEMA ÚNICO, isto é, a ligação HSDI entre MÓDULOS, e meios de programação de accionamento HSDI dentro do sistema operativo em cada MÓDULO. Aqui, num sistema de dois MÓDULOS, por exemplo, cada um dos sistemas operativos "tem conhecimento acerca de" todo o ambiente e pode aceder a facilidades através do HSDI sem a intervenção activa do "outro" sistema operativo. É considerável o dispêndio em comunicações.

Pode interligar-se por via do HSDI um grande número de MÓDULOS de várias dimensões e tipos de modelos para criar um sistema complexo que aparece ao programador como
• um ambiente (expansível). O seu produto, um programa de aplica-
•
•

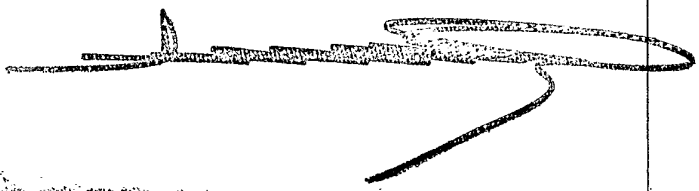


ção, pode ser armazenado num disco neste sistema complexo, executado em qualquer das CPU no complexo, controlado ou monitorado a partir substancialmente de um qualquer dos terminais do complexo, e pode transferir dados para e de quaisquer dispositivos de I/O do complexo, tudo sem considerações especiais de programação e sem um rendimento de execução melhorado em relação aos processos antigos.

O sistema operativo e as suas várias características e facilidades são escritos de modo a assumir nativamente o ambiente distribuído e operar dentro desse ambiente sem que o utilizador tenha necessidade de ter em atenção ou ter controlo sobre onde residem as várias entidades (serviços, aplicações, dados, linguagem, processadores, etc.). A chave para tornar tudo isto possível é a regra reforçada de que cada OBJECTO tem de ter um nome único; e esta regra estende-se facilmente a todo o sistema complexo visto que o qualificador de nomes é o nome dos MÓDULOS, que por sua vez tem de ser único no interior do complexo. Por conseguinte, a localização de qualquer OBJECTO em todo o complexo é tão simples como dar nomes correctos no mesmo. A nomeação de um OBJECTO é por sua vez simplificada para o programador pela provisão de ELOS DE LIGAÇÃO ("LINKS"), que permitem a utilização de indicadores aleatórios muito curtos para (substituir nomes de) OBJECTOS com nomes muito compridos e complicados.

Para conseguir realizar o conceito de uma "imagem de um sistema único derivada" no interior deste complexo de módulos S/88 interligados, acopla-se um certo número de processadores S/370 aos processadores S/88 de modo a proporcionar aos utilizadores dos processadores S/370 pelo menos alguns dos aspectos das características de imagem de um sistema único dos S/88. E isso embora os processadores e os sistemas operativos dos processadores S/370 não proporcionem essas características.

Proporcionam-se um ou mais processadores S/370 no interior do MÓDULO S/88. Um processador S/88 está acoplado unicamente a cada um dos processadores S/370. Como se verá, cada processador S/370 é duplicado e controlado por meios



de programação S/88 para o funcionamento tolerante de avarias. O acoplamento único e directo dos processadores S/88 e S/370, de preferência pelos mecanismos de desacoplamento e interrupção atrás descritos, torna as transferências de dados entre os processadores transparentes para ambos os sistemas operativos S/370 e S/88. Nenhum dos sistemas operativos tem conhecimento da existência do outro processador ou do outro sistema operativo.

Todos os processadores S/370 utilizam o complexo do sistema S/88 tolerante das avarias para proporcionar completamente a unidade de memória principal S/370, canal ou canais I/O S/370 e dispositivos de I/O emulados. OS S/370 não têm quaisquer memórias principais, canais ou dispositivos de I/O que não sejam parte do S/88, sendo todas estas facilidades tolerantes das avarias, por construção.

No instante da configuração do sistema, a cada processador S/370 é atribuído um bloco dedicado contíguo de 1 a 16 MB da memória principal dos recursos comuns dos S/88; este bloco é retirado das tabelas de configuração do S/88, de modo que, mesmo indadvertidamente, o sistema operativo S/88 não tem acesso ao mesmo. Registadores em equipamentos físicos tolerantes das avarias mantêm o indicador do bloco de memória para cada S/370, de modo tal que o S/370 não tem meios de acesso a qualquer memória principal que não seja a que lhe foi atribuída. O resultado é uma visão inteiramente convencional de sistema único da sua memória principal pelos S/370; o aspecto de tolerância das avarias da memória é completamente transparente. Um programa de aplicação (EXEC370) no S/88 emula o ou os canais e o ou os dispositivos de I/O S/370, utilizando dispositivos S/88 e chamadas do sistema operativo S/88 actuais. Ele tem a visão de imagem de sistema único do complexo S/88, visto que é um programa de aplicação; assim, esta visão estende-se a todo "pseudo-canal" S/370.

Do ponto de vista oposto, o do sistema operativo S/370 (e, por extensão, dos programas de aplicação), ele pode ajudar a visualizar uma "janela" (o canal) através da qual têm lugar todas as operações de I/O. A janela não é alterada no seu carácter - nenhum dos programas S/370 necessita de



ser alterado - mas a "vista" através da janela é alargada para incluir os atributos de IMAGEM DE UM SISTEMA ÚNICO. Um pequeno passo conceptual torna visível um grande número de S/370 compartilhando de maneira eficiente uma base de dados única, a que é gerida pelo S/88.

Uma consequência desta técnica de ligação é a reconfigurabilidade dinâmica relativamente simples e rápida de cada S/370. A "janela" do canal é bidireccional, estando o programa EXEC370 de controlo do S/88 do outro lado da mesma; EXEC370 tem capacidade total para parar, repor, reinicializar, reconfigurar e arrancar de novo a CPU S/370. Assim, por emulação transparente das facilidades de I/O s/370, usando outras facilidades que possuem o atributo de IMAGEM DE UM SISTEMA ÚNICO (I/O e sistema operativo S/88), este atributo é alargado e fornecido ao S/370.

Portanto, o S/370 foi dotado com independência de localização de objectos. Os seus utilizadores podem aceder a um ficheiro de dados ou outro recurso pelo nome, um nome que lhe é atribuído no catálogo do sistema operativo.

O utilizador não precisa de conhecer a localização do ficheiro de dados no complexo dos módulos S/370-S/88.

Os comandos de I/O S/370 enviados por uma unidade de processamento S/370 num módulo (9) são processados por uma unidade de processamento S/88 associada, estreitamente acoplada à unidade de processamento S/370 no mesmo módulo (ou por outras unidades de processamento S/88 interligadas no módulo (9) e controladas pela mesma cópia do sistema operativo virtual S/88 que suporta o multiprocessamento) para aceder a ficheiros de dados e similares residentes no mesmo ou em outros módulos ligados. Ele pode devolver os ficheiros a que acedeu para a unidade de processamento S/370 solicitante ou enviá-los para outros módulos, por exemplo para os fundir com outros ficheiros.

6. Sumário

Assim, as funções dos dois sistemas ope

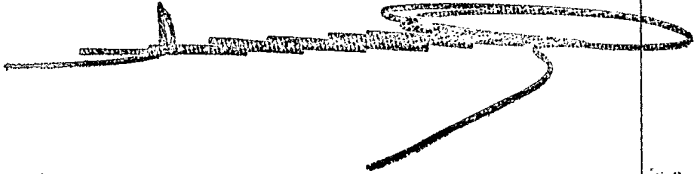


rativos virtuais (por exemplo VM, VSE ou IX370 do S/370 e OS do S/38) fundem-se num sistema físico. O processador S/88 executa o OS S/88 e trata os aspectos de tolerância de avarias do sistema. Ao mesmo tempo, ligam-se por fichas um ou mais processadores S/370 no bastidor dos S/38 e são atribuídos pelo OS S/88 de 1 a 16 MB de memória contígua por cada processador S/370. Cada sistema operativo virtual S/370 pensa que a sua atribuição de memória começa no endereço 0 e gere a sua memória através de técnicas normais de atribuição e paginação da memória dinâmica S/370. O S/370 é verificado quanto aos limites para impedir o seu acesso ao espaço de memória S/88. O S/88 tem de ter acesso ao espaço de endereços S/370, visto que o S/88 tem de deslocar dados de I/O para as memórias tampão de I/O S/370. O sistema operativo S/88 é o senhor de todos os meios físicos e dispositivos I/O do sistema. Os pares de processadores emparelhados executam os respectivos sistemas operativos num ambiente de sistema único sem alterações significativas da escrita de cada um dos sistemas operativos.

Introdução - Sistema S/88 da técnica anterior

Descrever-se-ão os aperfeiçoamentos segundo a presente invenção relativamente a uma forma preferida na qual unidades de processamento IBM System/370 (S/370) (que executam instruções S/370 sob o controlo de um qualquer dos sistemas operativos S/370, tais como VM, VSE, IX370, etc.) estão acopladas estreitamente a unidades de processamento IBM System/88 (S/88) (que executam instruções S/88 de uma maneira tolerante de avarias sob o controlo de um sistema operativo S/88 num ambiente tolerante de avarias) de uma maneira que permite o funcionamento tolerante de avarias das unidades de processamento S/370 com as características de imagem de um sistema único S/88, possibilidade de ligação de unidades com a energia de alimentação ligada, detecção instantâneo de erros, distribuição da carga de I/O e isolamento das avarias e reconfigurabilidade dinâmica.

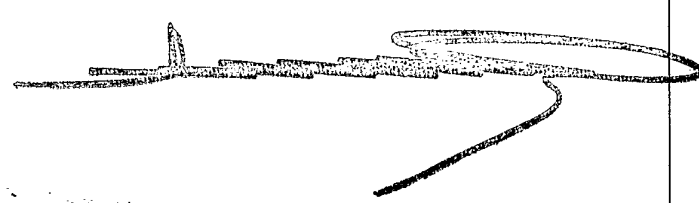
O IBM System/88, comercializado pela In



ternational Business Machines Corp., está descrito genericamente no IBM System/88 Digest, 2ª Edição, publicado em 1986, e noutras publicações de cliente S/88 disponíveis. O sistema computador S/88 que inclui o módulo (10), fig. 6A, é um sistema de elevada disponibilidade concebido para satisfazer as necessidades de clientes que exigem processamento em linha de elevada fiabilidade. O sistema S/88 combina uma arquitectura de equipamentos físicos duplexados com meios de programação sofisticados do sistema operativo para proporcionar um sistema tolerante das avarias. O sistema S/88 proporciona também a ampliação horizontal mediante a ligação de módulos múltiplos do sistema S/88 (10a, 10b, 10c), através de interligações de dados a grande velocidade (HSDI), fig. 6B, e módulos (10d-g) através da rede S/88, fig. 6C.

O sistema S/88 está concebido para detectar uma avaria num componente quando e onde ela ocorra, e para evitar que erros e interrupções provocadas por tais avarias sejam introduzidos no sistema. Como a tolerância das avarias faz parte da concepção dos meios de equipamento físico do sistema S/88, ela não exige programação pelo projectista da aplicação. A tolerância de avarias consegue-se sem aumento dispendio nos meios de programação nem degradação da eficácia. O sistema S/88 consegue a tolerância de avarias mediante a duplicação de componentes mais importantes, incluindo processadores, dispositivos de memória de acesso directo (DASD) ou discos, memória e controladores. Se se avariar um componente duplicado, o seu parceiro duplo automaticamente continua o processamento e o sistema mantém-se disponível para os utilizadores finais. Proporcionam-se também alimentações de energia em duplicado, com apoio por bateria, para a retenção da memória durante as falhas de energia de curta duração. O sistema S/88 e os seus produtos de meios de programação proporcionam uma ampliação fácil, a partilha de recursos entre os utilizadores e soluções para requisitos complexos, mantendo no entanto uma imagem de sistema único para o utilizador final.

Uma imagem de sistema único é um ambiente de processamento distribuído constituído por muitos processa



dores, cada um deles com os seus próprios ficheiros e I/O, interligados através de uma rede ou de uma LAN ("Local Area Network" - Rede de área local), que dá ao utilizador a impressão de estar registado numa máquina única. O sistema operativo permite que o utilizador mude de uma máquina para outra justamente mudando um catálogo de endereços.

Com um planeamento apropriado, a capacidade de processamento do S/88 pode ser aumentada, enquanto o sistema S/88 está em funcionamento ao mesmo tempo que mantém para o utilizador final a imagem de um sistema único. A ampliação horizontal é obtida combinando múltiplos módulos de processamento para obter sistemas que utilizam a HSDI do sistema S/88, e combinando múltiplos sistemas numa rede, utilizando a rede do sistema S/88.

Um módulo de processamento S/88 é um computador completo e independente, como se vê na fig. 6A dos desenhos. O sistema S/88 é ou um módulo único ou um grupo de módulos ligados numa rede local com a HSDI IBM, como se vê na fig. 6B. A rede S/88, que utiliza equipamentos de transmissão a distância, é a facilidade usada para interligar os múltiplos sistemas para formar uma imagem de sistema único para o utilizador final. Podem interligar-se dois ou mais sistemas por linhas de comunicações para formar uma rede de transmissão a grande distância. Esta ligação pode ser através de um cabo directo, de uma linha telefónica alugada ou de uma rede X.25. A rede do sistema S/88 detecta referências a recursos remotos e encaminha mensagens entre módulos e sistemas, de maneira completamente transparente para o utilizador.

A possibilidade de ligações por ficha, com a alimentação ligada, permite muitas substituições de equipamentos físicos feitos sem interrupção do funcionamento do sistema. O sistema S/88 retira do serviço um componente avariado, continuando o serviço com o parceiro duplo e acende um indicador no componente avariado - tudo sem intervenção de qualquer operador.

O pessoal do cliente ou de serviço pode retirar e substituir um cartão duplicado avariado, enquanto o



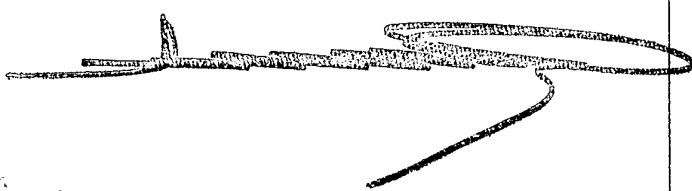
processamento contínua. Os benefícios para um cliente incluem reparações a tempo e reduzidos custos de manutenção.

Embora o sistema S/88 seja uma máquina tolerante de avarias, de funcionamento contínuo há alturas em que é necessário interromper o funcionamento da máquina. Alguns exemplos são: melhorar o sistema operativo do sistema S/88, alterar a configuração dos meios físicos (adicionar memória principal) ou efectuar certos procedimentos de serviço.

Os componentes duplicados do sistema S/88 e os meios de programação do sistema S/88 ajudam a manter a integridade dos dados. O sistema S/88 detecta uma avaria ou um erro transitório no momento da avaria e não a propaga através da aplicação ou dos dados. Os dados são protegidos da corrupção e mantêm-se a integridade do sistema. Cada componente contém a sua lógica de detecção e diagnóstico dos erros. A lógica de detecção dos erros compara os resultados de operações em paralelo em todo o ciclo da máquina.

Se o sistema detectar um mau funcionamento de um componente, esse componente é automaticamente retirado de serviço. O processamento continua no parceiro duplo enquanto o componente avariado é verificado por meios de diagnóstico internos. As funções de detecção de erros efectuem automaticamente um diagnóstico do componente retirado de serviço, enquanto o processamento continua no seu parceiro duplo. Se o diagnóstico determinar que certos componentes necessitam de ser substituídos, o sistema S/88 pode automaticamente chamar um centro de apoio para participar o problema. O cliente beneficia de reparações rápidas e baixos custos de manutenção.

O sistema S/88 baseia-se genericamente nos sistemas processadores do tipo descrito em pormenor na patente americana 4 453 215, intitulada "Central Processing Apparatus for Fault Tolerant Computing", publicada em 5 de Junho de 1984, concedida a Robert Reid e nas patentes americanas relacionadas 4 486 826, 4 597 084, 4 654 857, 4 750 177 e 4 816 990, sendo as referidas patentes aqui incorporadas por referência, na sua totalidade, como se fossem aqui apresentadas completamente. Nas fig. 7 e 8 do presente pedido de patente representam-se

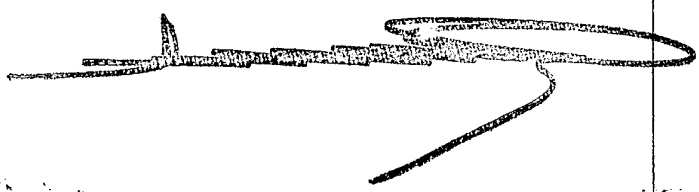


esquemáticamente partes da patente 4 453 215, de Reid.

Este sistema computador das fig. 7 e 8 do presente pedido de patente tem um módulo processador (10) com uma unidade de processamento (12), uma unidade de memória de acesso aleatório (16), unidades de controlo dos periféricos (20), (24), (32), e uma estrutura de linhas omnibus única (30) que proporciona todas as transferências de informação entre as várias unidades do módulo. A estrutura das linhas omnibus no interior de cada módulo processador inclui linhas omnibus parceiras em duplicado (A,B) e cada unidade funcional (12), (16), (20), (24), (32) tem uma unidade parceira em duplicado. Todas as unidades diferentes das unidades de controlo que funcionam com dispositivos periféricos assíncronos, normalmente operam com sincronização por arranque-e-paragem com a sua unidade parceira. Por exemplo as duas unidades de memória parceiras (16,18) de um módulo processador normalmente accionam as duas linhas omnibus parceiras (A,B) e são ambas accionadas pela estrutura (30) das linhas omnibus, em sincronismo completo.

O sistema computador proporciona detecção de avarias ao nível de cada unidade funcional dentro do módulo processador. Para obter esta característica, detectores de erros vigiam o funcionamento dos equipamentos físicos, dentro de cada unidade, e transferem a informação de verificação entre as unidades. A detecção de um erro faz com que o módulo processador isole a linha omnibus ou a unidade que provocou o erro para não transferir informação para as outras unidades, continuando o módulo em funcionamento. O funcionamento continuado utiliza o parceiro da unidade ou da linha omnibus avariadas. Quando a detecção do erro preceder uma transferência de informação, o funcionamento continuado pode executar a transferência ao mesmo tempo que teria ocorrido na ausência de avaria. Se a detecção do erro coincidir com uma transferência de informação, a operação continuada pode repetir a transferência.

O sistema computador pode efectuar a referida detecção de avarias e a acção de lhe dar remédio rapidamente, isto é, dentro de uma fracção de um ciclo de funcionamento. O sistema computador tem no máximo apenas uma informação



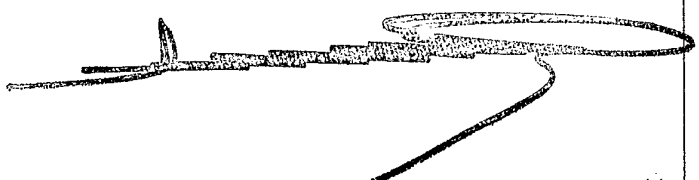
única que é de validade questionável e que exige a repetição para garantir a validade total dos dados.

Embora um módulo processador tenha uma redundância importante de meios físicos para proporcionar o funcionamento tolerante de avarias, um módulo sem unidades duplas é não obstante completamente operacional.

A redundância das unidades funcionais permite que o módulo continue a funcionar na eventualidade de uma avaria em qualquer unidade. Em geral, todas as unidades de um módulo processador funcionam continuamente, e com sincronismo seleccionado, na ausência de qualquer avaria detectada. Após a detecção de uma avaria, que se manifesta por um erro, em qualquer unidade, essa unidade é isolada e colocada fora de linha de modo que ela não pode transferir informação para outras unidades do módulo. A parceira da unidade fora de linha continua a funcionar, normalmente substancialmente sem qualquer interrupção.

Além da duplicação de unidades funcionais emparelhadas no interior de um módulo para proporcionar o funcionamento tolerante de avarias, cada unidade no interior de um módulo processador tem geralmente um duplicado de meios físicos que está implicado numa transferência de dados. A finalidade dessa duplicação, no interior de uma unidade funcional, é fazer ensaios, independentemente das outras unidades, para detecção de avarias no interior de cada unidade. Outra estrutura no interior de cada unidade de um módulo, incluindo a estrutura de detecção de erros, não é em geral duplicada.

A estrutura comum de linhas omnibus que serve todas as unidades de um módulo processador de preferência utiliza uma combinação dos dois níveis anteriores de duplicação e tem três conjuntos de condutores que formam uma linha omnibus (A), uma linha omnibus (B) que duplica a linha omnibus (A) e uma linha omnibus (X). As linhas omnibus (A) e (B) transportam cada uma um conjunto idêntico de sinais de definição de ciclos, de endereço, de dados, de paridade e outros, que podem ser comparados para a indicação de aviso de transferência de informação errada entre unidades. Os condutores da linha omnibus (X),

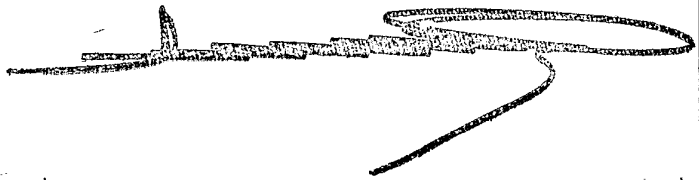


que não são duplicados, em geral transmitem sinais respeitantes a todo o módulo e outros sinais de operação, tais como sinais de distribuição de tempos, condições de erro e energia eléctrica. Uma linha omnibus adicional (C) é proporcionada para a comunicação local entre unidades emparelhadas.

Um módulo processador detecta e localiza uma avaria por uma combinação de técnicas no interior de cada unidade funcional, que incluem a comparação do funcionamento das secções duplicadas da unidade, a utilização de outros códigos de detecção e correcção de erros e vigiando parâmetros de funcionamento, tais como as tensões de alimentação. Cada unidade central de processamento tem duas secções de processamento redundantes e, se a comparação der um resultado inválido, isola a unidade de processamento para não transferir informação para a estrutura de linhas omnibus. Isso isola outras unidades funcionais do módulo processador de qualquer informação errada que possa provir da unidade de processamento em questão. Cada unidade de processamento tem também um andar para proporcionar o funcionamento de memória virtual, que não é duplicado. Em vez disso, a unidade de processamento utiliza técnicas de controlo de paridade para detectar uma avaria neste andar.

A unidade de memória de acesso aleatório (16) está disposta com duas secções de memória não redundantes, cada uma das quais está disposta para o armazenamento de octetos diferentes de uma palavra de memória. A unidade detecta uma avaria, quer em cada secção da memória, quer no conjunto composto das duas secções, com um código de correcção de erros. Mais uma vez, o detector de erros desactiva a unidade de memória para não transferir informação potencialmente errada para a estrutura das linhas omnibus e daí para as outras unidades.

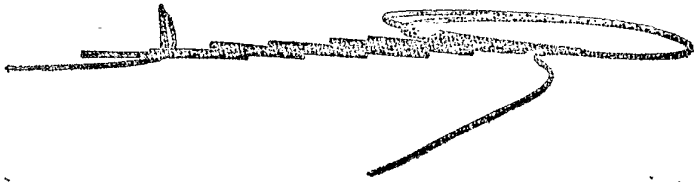
À unidade de memória (16) está também atribuída a tarefa de verificar os condutores das linhas omnibus duplicadas, isto é a linha omnibus (A) e a linha omnibus (B). Para isso, a unidade tem verificadores de paridade que ensaiam os sinais de endereços e que ensaiam os sinais de dados na estrutura de linhas omnibus. Além disso, um comparador compara todos os sinais na linha omnibus (A) com todos os sinais na



linha omnibus (B). Ao determinar deste modo que uma das linhas omnibus está avariada, a unidade de memória sinaliza nas outras unidades do módulo, através da linha omnibus (X), para obedecer apenas à linha omnibus não avariada.

As unidades de controlo dos periféricos para um módulo processador utilizam uma secção de interface com as linhas omnibus para ligação com a estrutura comum das linhas omnibus, secções de controlo duplicadas denominadas "excitação" ("drive") e "verificação" ("check"), e uma secção de interface dos periféricos que comunica entre as secções de controlo e os dispositivos de entrada/saída dos periféricos que a unidade serve. Há unidades de controlo de discos (20,22) para operação com memória de discos (52a,52b), uma unidade de controlo de comunicações (24,26) para a operação, através de painéis de comunicações (50) com dispositivos de comunicação que incluem terminais, impressoras e modem, e unidades de controlo da HSDI (32) e (34), para a interligação de um módulo processador com um outro num sistema multiprocessadores. Em qualquer dos casos, a secção de interface de linhas omnibus fornece sinais de entrada para as secções de controlo de excitação e de verificação a partir da linha omnibus (A) e/ou da linha omnibus (B), ensaios para detectar erros lógicos em certos sinais de entrada provenientes da estrutura de linhas omnibus e ensaios de identidade de sinais saídos dos canais de excitação e de verificação. A secção de controlo de excitação em cada unidade de controlo de periféricos proporciona funções de controlo, de endereço, de estado e de manipulação de dados, apropriados para o dispositivo de I/O que a unidade serve. A secção de controlo de verificação da unidade é substancialmente idêntica para a finalidade de verificar a secção de controlo de excitação. A secção de interface dos periféricos de cada unidade de controlo inclui uma combinação de dispositivos de paridade e comparadores para ensaiar sinais que passam entre a unidade de controlo e os dispositivos periféricos para detectar erros.

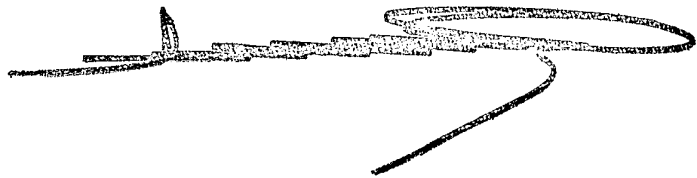
Uma unidade de controlo dos periféricos que opera com um dispositivo de I/O síncrono, tal como uma unidade de controlo de comunicação (24), opera com sincronização



por arranque e paragem com a sua unidade parceira. Porém, as unidades de controlo de discos emparelhadas (20,22) operam com memórias de discos diferentes não sincronizadas e por conseguinte operam com sincronismo limitado. As unidades de controlo de discos emparelhados (20,22) efectuam operações de escrita simultaneamente, mas não com sincronismo exacto, visto que as memórias de discos operam assincronamente uma em relação à outra. A unidade de controlo (32) e a sua parceira também operam tipicamente com este grau de sincronismo limitado.

A unidade de alimentação de energia para um módulo utiliza duas fontes de energia de grande capacidade, cada uma das quais fornece a energia de funcionamento para apenas uma unidade em cada par de unidades parceiras. Assim uma unidade de alimentação de grande capacidade alimenta a parte duplicada da estrutura de linhas omnibus, uma de duas unidades de processamento central parceiras, uma de duas unidades de memória parceiras e uma unidade em cada par de unidades de controlo de periféricos. As unidades de alimentação de grande capacidade também fornecem energia eléctrica para unidades não duplicadas do módulo processador. Cada unidade do módulo tem um andar de alimentação de energia que recebe energia de operação de uma unidade de alimentação de grande capacidade e que, por sua vez, gera as tensões de funcionamento exigidas por essa unidade. Esse andar de alimentação vigia além disso as tensões de alimentação. Ao detectar uma tensão de alimentação errada, o andar de alimentação produz um sinal que fixa ao potencial de terra todas as linhas de saída provenientes dessa unidade para a estrutura de linhas omnibus. Esta acção impede que uma falha de alimentação em qualquer unidade provoque a transmissão de informação errada para a estrutura de linhas omnibus.

Algumas unidades do módulo processador executam cada transferência de informação com um ciclo de operação que inclui uma fase de distribuição de tempos de detecção de erros antes da transferência de informação propriamente dita. Uma unidade que proporciona esta operação, por exemplo uma unidade de controlo de um dispositivo periférico; faz assim um ensaio para detectar uma condição de avaria antes de efectuar uma



transferência de informação. A unidade inibe a transferência de informação no caso de se ter detectado uma avaria. Mas o módulo pode continuar a operação - sem interrupção nem atraso - e efetuar a transferência de informação a partir da unidade parceira não inibida.

Outras unidades do módulo processador, que incluem geralmente pelo menos a unidade de processamento central e a unidade de memória, para as quais o tempo de operação tem maior importância, executa cada transferência de informação simultaneamente com a detecção de erro pertinente para essa transferência. No caso de ser detectada uma avaria, a unidade imediatamente produz um sinal que alerta outras unidades de processamento para descartar a transferência de informação imediatamente anterior. O módulo processador pode repetir a transferência de informação a partir do parceiro da unidade que informou a condição de avaria. Este modo de funcionamento produz uma velocidade de funcionamento ótima dado que cada transferência de informação é executada sem a demora destinada à detecção do erro. Um atraso só surge nos casos relativamente raros em que é detectada uma avaria. Um dispositivo de arbitragem de linhas omnibus é proporcionado para determinar qual a unidade que obtém o acesso às linhas omnibus do sistema quando há várias unidades a pedir esse acesso.

Módulo S/370 tolerante de avarias (9) interligado através de HSDI/redes

A fig. 7 ilustra, na parte por cima do módulo (10) da técnica anterior, a interligação de pares de processadores duplexados S/370 e S/88 (unidades emparelhadas) (21, 23) que, quando substituem as unidades S/88 duplexadas (12,14) no módulo (10), criam um módulo S/370 (9) novo e único. Quando tais módulos únicos (9) são interligados por HSDI S/88 e por redes, de uma maneira idêntica à representada nas fig. 6B,6C para os módulos (10), eles criam um complexo S/370 (em vez de um complexo S/88) com características S/88 de tolerância de avarias, de imagem de um sistema único, de possibilidade de ligação com



a alimentação ligada, de partilha de carga de I/O entre várias unidades de processamento S/88 no interior do mesmo módulo, etc..

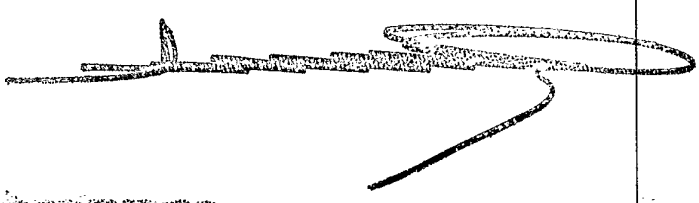
Em especial, os processadores nas unidades emparelhadas (21,23) dos módulos únicos (9) executam instruções S/370 sob o controlo do seu sistema operativo S/370; os processadores S/88 interligados efectuam todas as operações de I/O S/370 em conjunção com a sua memória S/38 respectiva e as unidades periféricas S/88, sob o controlo do sistema operativo S/88 em conjunção com um programa de aplicação S/88.

Além disso, outras unidades emparelhadas (25,27) e (29,31) do processador S/370-S/88 podem ser incorporadas dentro do novo módulo (9) para permitir um ambiente de processadores S/370 plural no interior do módulo único (9). Além disso, os processadores no interior das unidades emparelhadas (21,23) e (25,27) e (29,31) podem operar cada uma sob um sistema operativo S/370 diferente por par de parceiros.

Descrição geral das unidades emparelhadas (21,23) do processador duplicadas

A fig. 8 ilustra uma forma preferida de interligação de processadores S/370 e S/88 no interior da unidade (21). A parte inferior da unidade (21) compreende um processador central (12), substancialmente idêntico ao processador (12) da patente de Reid atrás mencionada, excepto na utilização de um elemento processador único em cada um dos elementos (60, 62) do par de processadores. Na patente de Reid, eram proporcionados processadores duais em (60) e em (62) para executar respectivamente o código do utilizador e o código do sistema operativo.

No presente pedido de patente, ambas as funções são executadas por um microprocessador único, de preferência um microprocessador Motorola MC68020, descrito no MC68020 Users Manual, 3ª Edição (ISBN-0-13-567017-9) publicado pela Motorola, copyright 1989, 1988. A referida publicação é aqui incorporada por referência, como se fosse aqui apresentada com-



pletamente.

Assim, cada elemento processador (PE) - (60) e (62) - de preferência compreende um microprocessador Motorola 68020. Os multiplexadores (61,63) ligam os elementos processadores (60,62) à estrutura de linhas omnibus (30) através de linhas omnibus de endereços/controlo de dados (A) e (B) e dos emissores-receptores (12e), de uma maneira descrita em pormenor na patente de Reid. Proporcionam-se um controlo local (64,66) e um mapa da memória virtual (12c) para os elementos (60,62). Um comparador (12f) verifica as avarias que produzem erros, por comparação dos sinais nas linhas de controlo, de dados e de endereços para e da linha omnibus (30) e para e dos elementos processadores (60,62). A não coincidência dos sinais produz um sinal de erro a partir do comparador (12f) para os circuitos de controlo comuns (86) que envia sinais de erro pela linha omnibus (X) da estrutura de linhas omnibus (30) e desactiva excitadores (não representados) nos emissores-receptores (12e) para pôr fora de linha a unidade de processamento (12). Circuitos de sujeição (88,90) respondem a uma falha de energia na unidade (12) para sujeitar ao potencial de terra todas as linhas de saída da unidade (12) para a estrutura de linhas omnibus (30). Estes componentes são descritos com mais pormenor na patente de Reid.

A porção superior da fig. 8 ilustra uma forma preferida da ligação de um par de elementos de processamento S/370 (85,87) à estrutura das linhas omnibus S/88 e aos elementos de processamento S/88 (60,62). Os elementos de procesamento (85,87) estão ligados à estrutura de linhas omnibus (30) através de multiplexadores (71,73) e emissores-receptores (13), de uma maneira logicamente semelhante à dos elementos (60,62) ligados à mesma estrutura de linhas omnibus (30).

Um circuito de comparação (15) (descrito mais completamente nas fig. 32A,B), circuitos de sujeição (77) e (79) e controlos comuns (75) são proporcionados e funcionam de maneira semelhante aos componentes correspondentes na unidade (12). O circuito de controlo (86) está acoplado ao mecanismo de interrupções S/88 dos elementos de processamento (60,

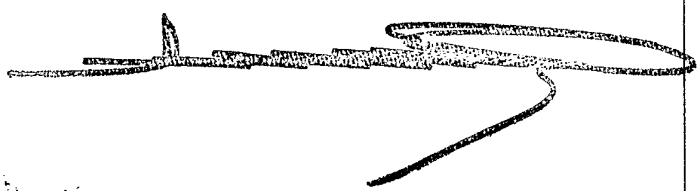


62). Os processadores S/370 (85,87) e os seus dispositivos de meios físicos com eles relacionados utiliza o S/88 para processar o tratamento e a recuperação dos erros. Assim, o circuito de controlo comum (75) está acoplado ao circuito de controlo comum (86), através da linha (95), para permitir que este último trate erros detectados pelo circuito de comparação (15). Esta linha de acoplamento (95) também permite que os controlos comuns (75) e (86) ponham ambos os seus pares de processadores respectivos (85,87) e (60,62) fora de linha, no caso de um erro em qualquer par de processadores.

Uma forma preferida de unidades de processamento S/370 na unidade (21) inclui as unidades (81,83) de gestão da memória dos elementos de processamento central (85, 87) e as interfaces (89,91) entre processador e processador (por exemplo S/370 para S/88). As unidades de gestão da memória (81,83) acoplam a memória principal (16) dos elementos processadores (85,87) para S/88, através de multiplexadores (71,73), dos emissores-receptores e da estrutura das linhas omnibus (30).

Um circuito de comparação (15) (descrito mais completamente nas fig. 32A,B), circuitos de sujeição (77) e (79) e controlos comuns (75) funcionam de uma maneira análoga aos componentes correspondentes da unidade (12). O circuito de controlo (86) está acoplado ao mecanismo de interrupção S/88 dos elementos de processamento (60,62). Os processadores S/370 (85,87) e os seus equipamentos físicos relacionados utilizam o S/88 para o tratamento e a recuperação dos erros do processo. Assim, o circuito de controlo comum (75) é acoplado ao circuito de controlo comum (86) através da linha (95) para permitir que este último trate os erros detectados pelo circuito de comparação (15). Esta linha de acoplamento (95) também permite que os comandos comuns (75) e (86) coloquem ambas os seus pares respectivos de processadores (85,87) e (60,62) fora de linha, na eventualidade de um erro em qualquer dos pares de processadores.

Uma forma preferida das unidades de processamento S/370 da unidade (21) inclui as unidades (81,83) de gestão das memórias dos elementos de processamento central e as



interfaces (89,91) processador-processador (por exemplo S/370 para S/88). As unidades de gestão das memórias (81,83) acoplam os elementos de processamento (85,87) à memória principal (16) do S/88 através de multiplexadores (71,73), emissores-receptores (13) e uma estrutura de linhas omnibus (30).

As interfaces (89,91) acoplam as linhas omnibus dos elementos de processamento (85,87) do S/370 respectivamente às linhas omnibus dos processadores dos elementos de processamento (62,60) do S/88.

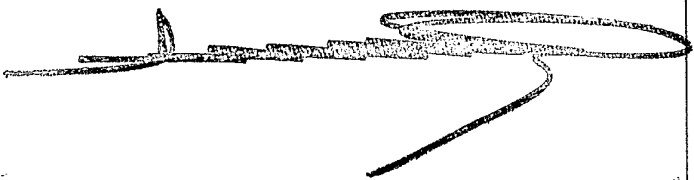
A unidade de processador parceira (23) é idêntica à unidade de processador (21). Recordar-se que, relativamente à descrição anterior, os dois elementos de processamento (60,62) na unidade (21) e os dois elementos correspondentes (não representados) na unidade (23) todos operam normalmente com sincronização por arranque e paragem uns com os outros para executar simultaneamente instruções idênticas sob o controlo do mesmo sistema operativo S/88.

Analogamente, os elementos processadores (85,87) na unidade (21) e os seus elementos correspondentes (não representados) na unidade (23) operam em sincronização de arranque e paragem uns com os outros para executar simultaneamente instruções idênticas sob o controlo do mesmo sistema operativo S/370.

No caso de um erro na unidade (21) ou na (23), essa unidade é retirada de serviço para permitir a continuação do funcionamento tolerante das avarias pela outra unidade.

Embora se descrevam alguns pormenores de uma forma de realização prática específica de uma unidade de processamento, mais adiante, compreender-se-á que podem usar-se outras formas de realização conhecidas que sejam compatíveis com os requisitos descritos em IBM System/370 Principles of Operation (publicação Nº. GA22-7000-10, 11ª edição, Setembro de 1987) publicada por e disponível na International Business Machines Corporation. A referida publicação é aqui incorporada por referência como se fosse apresentada completamente.

As fig. 9A e 9B mostram uma forma de



acondicionamento físico dos componentes S/370 e S/88 para a unidade de processador (21) da fig. 8. Os componentes S/370 que incluem os elementos de processamento emparelhados (85,87) estão montados num cartão (101) e os componentes S/88 que incluem os elementos de processamento emparelhados (60,62) estão montados num outro cartão (102). Os dois cartões (101) e (102) estão rigidamente fixados um ao outro para formar um par em sanduiche (103) e estão adaptados para a inserção em duas ranhuras do painel traseiro (não representado) do módulo (9), fazendo uma cablagem convencional do painel traseiro o acoplamento dos componentes dos cartões (101) e (102) uns aos outros e à estrutura de linhas omnibus (30), como se ilustra na fig. 8 e se descreveu na patente de Reid.

Antes de descrever os pormenores do acoplamento directo de um processador S/370 a um processador S/88, será uma ajuda proporcionar uma breve referência ao mecanismo que permite que o S/370: 1) utilize uma parte da memória principal S/88 e 2) permuta comandos e dados com o S/88 utilizando um certo espaço de memória virtual S/88. Estes mecanismos serão descritos mais tarde com mais pormenor.

Assim, a fig. 10 é usada para ilustrar uma forma preferida da atribuição de posições na memória virtual S/88 para a memória real (16), por uma unidade de gestão da memória (105) para um módulo (9). O espaço de endereços virtuais (106) está dividido no espaço (107) do sistema operativo S/88 e o espaço de aplicação dos utilizadores (108). Dentro do espaço (107) há uma área (109) (endereços 007E0000 a 007EFFFF) reservada para o equipamento físico e o código usados para acoplar cada elemento do processador S/370 a um elemento respectivo do processador S/88, numa unidade de processador tal como a (21). O espaço de endereços (109) é tornado transparente para o sistema operativo S/88 durante o processamento normal do sistema.

Durante a inicialização do sistema, a unidade (105) de gestão da memória atribui dentro da unidade (16) da memória principal S/88 uma área de memória principal S/370 para cada conjunto de quatro elementos do processador



S/370 em unidades emparelhadas, tais como (21) e (23). Assim, proporcionam-se três áreas de memória principal S/370 (162), (163) e (164) para unidades emparelhadas (21,23) e (25,27) e (29,31), respectivamente. Os elementos do processador S/88 no interior das unidades parceiras têm acesso às partes restantes da unidade de memória (16) da maneira descrita na patente Reid.

As áreas de memória S/370 (162-164) são atribuídas, como se descreverá mais tarde, de uma maneira tal que o sistema operativo S/88 não saiba que estas áreas foram "roubadas" e não são susceptíveis de ser de novo atribuídas a utilizadores S/88 pela unidade de gestão da memória, a menos que voltem para o espaço S/88. Como os sistemas S/370 são sistemas virtuais, eles acedem à respectiva área de memória principal por translação de endereços. A unidade de memória principal S/88 parceira exige áreas de memória principal S/370 idênticas (não representadas). Cada elemento de processamento S/370 pode aceder apenas à sua área de memória principal S/370 respectiva e produz um sinal de erro se tentar aceder ao espaço de memória principal S/88. Porém, cada elemento de processamento S/88 pode aceder (ou orientar o acesso) à área de memória principal S/370 do seu elemento de processamento S/370 durante as operações de I/O S/370, quando o elemento de processamento S/88 actuar como controlador I/O para o seu elemento de processamento S/370.

Acoplamento de elementos de processamento S/370 e S/88 (85,62)
(fig. 11 e 12)

A fig. 8 ilustra esquematicamente a provisão de quatro elementos de processamento S/370, tais como (85), dois em cada uma das unidades emparelhadas (21,23) e quatro elementos de processamento S/88, tais como (62), dois em cada unidade (21,23) acoplados de tal modo que os elementos de processamento S/370 executam simultaneamente instruções S/370 idênticas e todos os elementos de processamento S/88 executam simultaneamente instruções S/88 idênticas. Assim, todos os quatro elementos de processamento S/370 actuam como uma unidade de



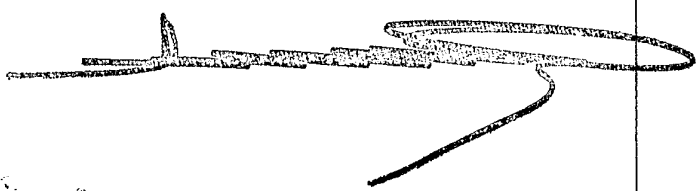
processamento S/370, no que respeita à execução de programas. Analogamente, todos os quatro elementos de processamento S/88 actuam com uma unidade de processamento S/88.

Por conseguinte, para facilidade de ilustração e explicação, as partes seguintes dos desenhos e da memória descritiva dirigem-se primariamente a um elemento de processamento S/370 (85) e a um elemento de processamento S/88 (62) e aos seus meios físicos e código de programas, excepto quando a duplicação de componentes exigir explicações adicionais.

Analogamente, omitir-se-á o acoplamento de elementos de processamento à estrutura de linhas omnibus (30), por exemplo através de multiplexadores (61), (63), (71) e (73) e de emissores-receptores (12e) e (11) da descrição seguinte, para facilitar a ilustração e a explicação. Far-se-á uma breve referência a este acoplamento, com referência à fig. 32.

Portanto, a fig. 11 representa o elemento de processamento (85) acoplado às linhas omnibus do sistema (30) e à memória S/88 (16), através de um primeiro trajecto que inclui a sua linha omnibus do processador (170) e a unidade de gestão da memória S/370 (81). O elemento de processamento (85) está representado acoplado à linha omnibus do processador (161) do elemento de processamento (PE) (62), através de um segundo trajecto que inclui o elemento de processamento para a interface dos elementos de processamento (89). O PE (85) utiliza o primeiro trajecto durante a execução de programas S/370 para buscar (e armazenar) dados e instruções a partir da sua área de memória principal (162) S/370 na memória (16). O PE (62) executa operações de I/O S/370 para o PE (85) através do segundo trajecto que inclui a interface (89).

Numa forma de realização preferida, um conjunto (150) de micropastilhas S/370 (fig. 11) inclui micropastilhas funcionais individuais para o elemento processador (85), um relógio (152), um controlador de memórias "cache" (153) com uma tabela separada de consulta do catálogo de endereços (DLTA) (341), um adaptador das linhas omnibus (154) um elemento de co-processamento em vírgula flutuante (151), optativo, e uma



memória de controlo (171) para armazenar um conjunto de microcódigo que suporta a arquitectura S/370. Este conjunto de micropastilhas S/370 pode estar adaptado para ser operado por um qualquer dos sistemas operativos S/370 existentes (tais como VSE/SP, VM/SP, IX/370, etc.) comercializados pela International Business Machines Corporation.

O controlador de memória "cache" (153), juntamente com uma interface de controlo da memória (STCI) (155) formam a unidade de gestão da memória S/370 (81). O adaptador de linhas omnibus (154) e a unidade de controlo de linhas omnibus (BCU) (156) constituem a interface entre PE e PE.

Na forma de realização preferida, cada uma das CPU S/370, tal como a PE (85) é um microprocessador de 32 bits com um fluxo de dados de 32 bits, uma unidade aritmética/lógica (ALU) de 32 bits, registadores de 32 bits e uma memória local de dados de três portas, e uma memória tampão de instruções S/370 de 8 octetos. As instruções S/370 ou são executadas em equipamentos físicos ou são interpretadas por microinstruções. A micropastilha (153) proporciona memória "cache" para instruções de programas S/370 e dados, juntamente com funções de controlo de memória associadas. A micropastilha (153) trata todos os pedidos de memória que são emitidos pelo PE (85) e executa as instruções dos seus programas. A micropastilha (153) também trata os pedidos provenientes do adaptador de linhas omnibus (154) quando se transferem dados de I/O.

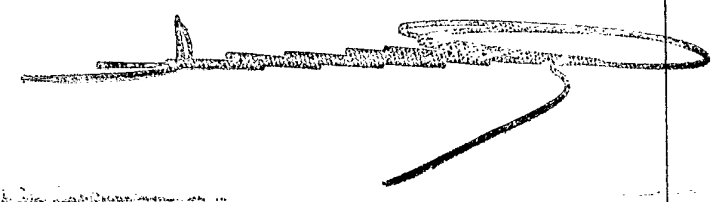
O adaptador de linhas omnibus (154) e a unidade de controlo das linhas omnibus (156) (BCU) proporcionam a lógica e o controlo para interligar directamente (ou rigidamente) a linha omnibus interna do processador S/370 à linha omnibus (161) do processador S/88 durante as operações de entrada/saída (I/O). A unidade BCU (156) é o mecanismo primário para acoplar directamente as linhas omnibus dos processadores PE (85) e PE (62) entre si. É o mecanismo de equipamentos físicos que interage com o elemento processador S/88 (62) quando o elemento processador (62) é "desacoplado" do seu equipamento físico do sistema associado para a transferência de dados e comandos entre o PE (62) e o PE (85), como se descreverá mais adiante.



A micropastilha do relógio (152) utiliza uma lógica centralizada para a geração dos sinais de relógio e aplica sinais de relógio apropriados individualmente a cada uma das outras micropastilhas (85), (151), (153) e (154). O relógio (152) é, por sua vez, controlado pelos sinais de relógio provenientes da linha omnibus do sistema S/88 (30), para sincronizar os dois elementos de processamento S/370 (PE85) e S/88 (PE62).

Uma parte integrante da fusão das duas arquitecturas distintas de equipamentos físicos S/370 e S/88, fora do equipamento físico de acoplamento/desacoplamento dos processadores, é um meio de ligar de maneira síncrona o equipamento físico anteriormente não tolerante das avarias à estrutura de linhas omnibus (30) tolerante das avarias. Numa forma de realização preferida, esta interface é tratada pela lógica STCI (155) que tem de comunicar entre o controlador de memórias "cache" S/370 (153) e a linha omnibus (30) do sistema. Além disso, os equipamento físicos não tolerantes das avarias têm de ser duplicados no cartão, como se representa na fig. 8 para produzir uma lógica de "verificação" e "excitação" susceptível de funcionar com sincronização por arranque e paragem, entre si e com uma unidade emparelhada. Assim a CPU "única" constituída por componentes do sistema em cartões (101) e (102) tem de funcionar em sincronismo por arranque e paragem com a respectiva unidade parceira duplexada. A tarefa da realização prática dos requisitos anteriores, ao mesmo tempo que se mantém uma eficácia e uma funcionalidade óptimas implica a sincronização de fontes de sinais de relógio separadas.

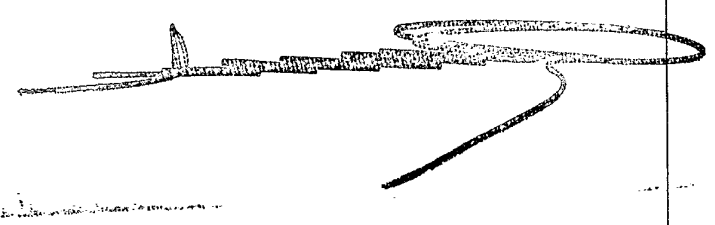
Na forma de realização preferida, o relógio (38) (fig. 7) do sistema S/88 é recebido por todos os dispositivos ligados à estrutura comum de linhas omnibus (30) definindo-se dois ciclos de sinais de relógio S/88 por cada ciclo das linhas omnibus (30). Este relógio (38) do sistema garante a comunicação síncrona na linha omnibus e pode ser usado por processadores/controladores individuais para produzir fontes internas de frequências de relógio baseadas no relógio do sistema. O equipamento físico S/370 utiliza uma entrada de oscilador na micropastilha de relógio S/370 (152), que gera então um conjunto



de sinais de relógio único para cada uma das outras micropastilhas S/370 (85), (151), (153), (154) e (155). Esta micropastilha de relógio (152) tem um atraso inerente que pode variar com base em vários parâmetros, tais como a temperatura de funcionamento, as variações de fabrico, etc.. Esta variação do atraso pode ser inaceitável, quer para manter a sincronização de arranque e paragem entre a lógica de verificação e excitação redundantes, bem como para manter plena capacidade de canalização de informações ("pipelining") entre a interface STCI (155) e a estrutura de linhas omnibus (30).

Como se ilustra nas fig. 12 e 19c, a forma de realização preferida utiliza a lógica de sincronização (158) e (158a), não representada, para a unidade do processador S/370 emparelhada) do relógio redundante, para permitir que os dois lados de verificação e excitação do processador de um cartão (101) funcionem com sincronização de arranque e paragem depois de uma reposição (isto é, religação da energia ou outra), enquanto se sincroniza o ciclo do processador S/370 com o ciclo das linhas omnibus (30) do S/88. Os sinais de relógio provenientes do relógio (38) S/88 são aplicados através da estrutura de linhas omnibus (30) à lógica de sincronização (158) e à interface STCI (155), para a sincronização S/88-S/370 e para aceder à memória principal através das linhas omnibus (30) do sistema.

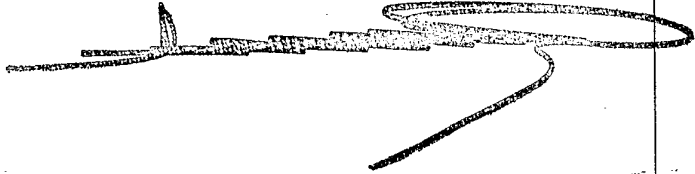
Esta sincronização é obtida na lógica de sincronização do relógio (158) multiplicando primeiramente o relógio do S/88 para obter a frequência de entrada do oscilador S/370 desejada para a micropastilha (152) do relógio S/370. Neste caso, ela é duas vezes a frequência dos ciclos de relógio S/88 e S/370. Em segundo lugar, faz-se a exploração de amostragem de um impulso de retroacção na linha (159), que representa o início do ciclo S/370, com os impulsos de relógio S/88 que representam os bordos dianteiro e traseiro de um período superior em um período de atraso do retentor do registador ao período do relógio de entrada do oscilador S/370 que, por sua vez, é igual à duração de meio ciclo S/88. No caso de uma reposição na qual o impulso de retroacção do relógio S/370 explorado, na linha (159), cai fora da janela explorada, ou sobrepondo-se ao início



do relógio S/88, então a entrada do oscilador S/370 é negada durante um ciclo S/370. Isto serve para "prolongar" o ciclo S/370 de modo que, na forma de realização preferida, a amostragem seguinte do impulso (na linha (159)) de retroação do relógio S/370 caia seguramente dentro da janela desejada. Toda a lógica de comparação (15) (fig. 8), representada com mais pormenor na fig. 32 (por exemplo (402 a-g)), é ignorada durante este tempo para permitir que o equipamento físico de verificação e excitação se sincronize.

Portanto, está assegurado que o ciclo do processador S/370 se inicia dentro do período do meio ciclo S/88 do início do período de relógio S/88. Todas as distribuições de tempo para as transferências entre a estrutura de linhas omnibus (30) e o controlador de memória "cache" S/370 (153) assume o atraso no pior dos casos para este meio ciclo. Além disso, a lógica de comparação (15) é alimentada apenas por linhas exploradas para amostragem com impulsos de relógio S/88, garantindo a sincronização da lógica "interrompida" (403) (fig. 32) com o cartão (102) do processador S/108 que as acompanha. Portanto, embora os equipamentos físicos de verificação e excitação S/370 possam de facto estar ligeiramente dessincronizados, devido a variações dos atrasos na lógica de geração dos respectivos impulsos de relógio, ambos os lados funcionam com sincronização por arranque e paragem, relativamente ao relógio (38) S/88 corrente, comum à estrutura de linhas omnibus (30), e nunca mais de meio ciclo depois do arranque do ciclo do relógio S/38. A lógica de sincronização (158) continua a monitorar o retorno dos impulsos de relógio S/370 na linha (159) para garantir que não há nenhuma deriva superior a meio ciclo. É necessário um máximo de um ciclo das linhas omnibus (30), na forma de realização preferida, para levar os dois lados ao sincronismo durante qualquer reposição do sistema; porém, qualquer desvio do atraso total fora da reposição, que faz com que um dos lados "prolongue" os seus impulsos de relógio S/370, terá como consequência uma condição de cartão "interrompido", isto é, uma avaria.

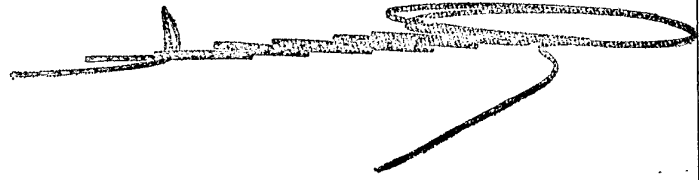
A fig. 12 representa o dispositivo da fig. 11 com mais pormenor. A memória (171) de controlo S/370 es



tá representada ligada ao PE (85). A memória de controlo (171) consiste, na forma de realização preferida, numa memória de 16 KB, de acesso aleatório, para armazenar micro-instruções que controlam a execução de instruções de programas e operações de I/O no interior do PE (85). A memória de controlo (171) também inclui em si um bloco de 64 KB (186) (fig. 29), que é usado como uma memória tampão para conter um microcódigo transitório carregado com base num pedido proveniente de uma área objecto interna (IOA) (187) (fig. 28), que faz parte da memória (162) dedicada S/370 no interior da unidade de memória principal (16). Nesta figura, a estrutura de linhas omnibus (161) do PE (62) está representada dividida numa linha omnibus de endereços virtuais (161A) e na linha omnibus de dados (161D). O PE (62) tem com ele associados equipamentos físicos que incluem um processador em vírgula flutuante FFP (172), uma memória "cache" (173), uma unidade (174) de armazenamento de microcódigo (174), que é usada para armazenar o microcódigo de acoplamento, aqui designado por ETIO. Quer o microcódigo, quer um programa de aplicação armazenado na memória "cache" (173), como se verá mais adiante, são usados para controlar o PE (62) e a lógica da unidade de controlo da linha omnibus BCU (156) para efectuar operações de I/O para o PE (85).

Os equipamentos físicos PE (62) também incluem um mecanismo de tradução de endereços (175). Uma canalização de escrita ("write pipe") (176) armazena temporariamente dados durante um ciclo de escrita para aplicação desses dados nas linhas omnibus do sistema (30) durante o ciclo seguinte, para acelerar o funcionamento do sistema S/88. A lógica das linhas omnibus do sistema/88 (177) do tipo descrito na patente de Reid acopla a unidade de tradução (175) e a canalização de escrita (176) à linha omnibus do sistema (30) de uma maneira descrita genericamente na referida patente de Reid. Uma unidade lógica da linha omnibus S/88 semelhante (178) acopla a interface de controlo da memória (155) à linha omnibus do sistema (30).

Uma memória tampão (180), uma memória fixa programável (181), uma memória (182) e um conjunto de registadores (183) estão acoplados ao PE (62) para serem usados duran



te a inicialização do sistema S/88 e do sistema S/370. A memória PROM (181) tem um código do sistema e um IDCODE necessários para inicializar o sistema a partir de uma sequência de ligação da energia. A memória PROM 181 tem o código de sincronização para o S/88. O registador (183) tem o registador do estado do sistema e do controlo.

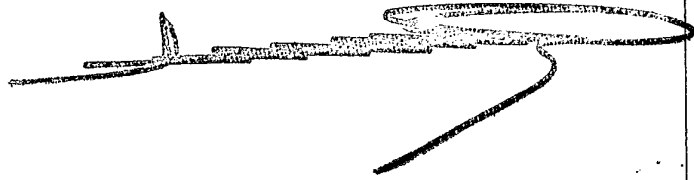
Dois dos conjuntos de micropastilhas S/370 estão montados no mesmo cartão físico, que é sincronizado, e executa programas com sincronização por arranque e paragem, para proporcionar autoverificação ao cartão. A linha omnibus da STC (157) e uma linha omnibus do canal (0,1) será vigiada relativamente a falhas de tensão, de modo que o processador S/370 não pode propagar um erro para outra unidade substituível do campo.

A unidade de controlo da linha omnibus (BCU) (156) e o adaptador de interface (89) permite que cada processador (PE 62) e (PE 85) tenha controlo apropriado sobre o outro processador, de modo que nenhum dos sistemas operativos tem o controlo completo do sistema. Cada uma das funções dos processadores é controlada em parte pela interface (89) e o microcódigo que passa em cada processador.

Interface (89) entre processador e processador

1. Adaptador I/O (154)

O adaptador (154) (fig. 13) tem uma interface com o processador S/370 (85) para a BCU (156), através dos seus canais de saída (0,1). Os canais incluem duas linhas omnibus de dados (250,251) assíncronos, (A/D), com a largura de dois octetos. As linhas omnibus (250,251) estão acopladas ao trajecto de dados síncronos, com a largura de dois octetos, na linha omnibus do processador (170) através de duas memórias tampão de 64 octetos (259,260). Os dados são transferidos da BCU (156) para o adaptador (154) (e para a memória principal S/370 (162)) através da linha omnibus (251) e do adaptador (154) para a BCU (156) através da linha omnibus (250).

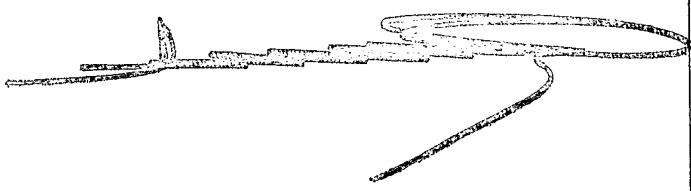


O adaptador (154) inclui os registadores seguintes:

1. O registador de base (110) contém o comprimento dos endereços de base e de filas de espera usados para as filas de espera e para o endereçamento para a caixa de correio.
2. Os registadores (111) e (112) de indicadores de leitura (RPNTR) e de indicadores de escrita (WPNTR) contém o desvio do endereço de base em relação à entrada seguinte na fila de espera a que deve aceder-se para uma leitura ou para uma escrita, respectivamente. O seu valor será carregado juntamente com o comando no registador de emissão quando os comando/endereço devem ser transferidos para o controlador da memória "cache" (153), através da linha omnibus (170).
3. O registador de estados (IOSR) (118) contém todos os pedidos Pu-BCU e BCU-PU, o estado da fila de espera de mensagens de entrada e o estado da interface BCU.
4. Se um bit no registador de activação de excepção (ER) (119) for 1 e o bit IOSR correspondente for 1, é excitada uma excepção no elemento processador (PE) (85).
5. O registador de palavras de controlo (CW) (120) controla o ajustamento/reposição de alguns bits IOSR.
6. O registador dos limites de verificação de endereços (ACBR) (121) retém o endereço da página inicial da área objecto interna (IOA) (187).
7. Os registadores das chaves de endereços (ADDR/KEY) (122) e (123) são normalmente carregados pela BCU (156) através das linhas omnibus de endereços/dados (250) e (251) para aceder a uma posição na memória (162). Estes registadores podem ser carregados pela PE (85) para fins de ensaio.
8. Os registadores de comandos (CMD0,1) (124) e (125) são normalmente carregados com uma contagem de comandos e de octetos pela BCU (156). Os registadores podem ser carregados pela PE (85) para fins de ensaio.

O adaptador (154) é a interface entre a PE (85) e a BCU (156). Localmente, o adaptador (154) proporciona os seguintes serviços à BCU (156):

- acesso à memória principal S/370 (162)

- 
- acesso a uma caixa de correio e a uma fila de espera de mensagens na memória S/370 (162)
 - um mecanismo de pedido/resposta entre a PE (85) e a BCU (156).

A BCU (156) tem acesso a toda a memória (162), incluindo a área IOA (187) (fig. 28). O adaptador (154) realiza a verificação dos limites de endereços (verificação ACB entre a área IOA (187) e a área do utilizador (165)), enquanto a verificação da chave é feita pelo controlador da memória "cache" (153) depois de receber a chave, o comando e os dados do endereço da memória (162), através da linha omnibus do processador (170) a partir do adaptador (154). Se a linha de dados endereçada a armazenar estiver contida na memória "cache", então os dados são armazenados na memória "cache". Caso contrário, o controlador (153) transfere os dados para a memória principal (162). Para as buscas de dados aplica-se o mesmo mecanismo no controlador da memória "cache" (153).

As transferências de comando I/O e de mensagens entre a PE (85) e a BCU (156) são feitas através de posições pré-definidas da memória (162) (área de caixa do correio (188) e fila de espera de mensagens entradas (189)) representada na fig. 28.

A BCU (156) busca comandos I/O provenientes da área de caixa do correio (188) de 16 octetos. O endereço para os acessos à caixa do correio é calculado da seguinte maneira:

endereço de base + comprimento da fila de espera de mensagens + desvio na caixa do correio

Os dois primeiros termos são fornecidos pelo registador de base (110) do adaptador (154) e o último pela BCU (156). O comprimento da fila de espera é ajustado por dois bits no registador de base (110) para 1, 2, 4 ou 8 KB (isto é, 64 a 512 entradas). A sua base é ajustada no registador de base (110) a um limite de duas vezes a dimensão da memória tampão (isto é, 2 - 16 KB, respectivamente).

A fila de espera das mensagens entradas (189) armazena todas as mensagens entradas através da BCU (154)



por ordem cronológica. Cada entrada tem um comprimento de 16 octetos.

Os indicadores de leitura (RPNTR) e de escrita (WPNTR) nos registadores (111) e (112) são usados pela BCU (156) para ler entradas de, e escrever entradas para a fila de espera (189). A PE (85) acede ao indicador de leitura por uma operação de exploração. O endereço de base no registador (110) mais WPNTR aponta para a entrada seguinte da fila de espera a escrever e o endereço de base mais RPNTR aponta para a entrada seguinte da fila de espera a ler.

Estes indicadores são actualizados depois de cada operação na fila de espera:

$WPNTR + 16 = WPNTR$ depois de uma escrita

$RPNTR + 16 = RPNTR$ depois de uma leitura

Resultam as condições seguintes da comparação dos indicadores:

$RPNTR = WPNTR$ A fila de espera está vazia

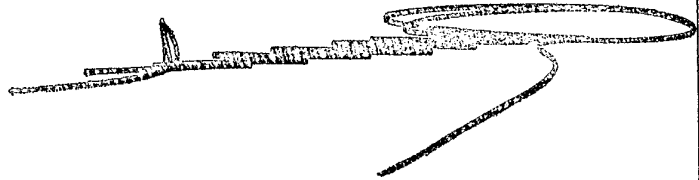
$RPNTR = WPNTR + 16$ A fila de espera está cheia; se a BCU (156) pedir uma escrita à fila de espera: memória tampão não disponível (BNA) emitido para a BCU através da linha omnibus de estados.

A validade dos dados armazenados na área de caixa do correio (188) é sinalizada da PE (85) para a BCU (156) e vice-versa, pelos mecanismos seguintes:

O sinal "PU to BCU request" na linha (256a) (fig. 16) é ajustado pela PE (85) com uma microinstrução de controlo. Esse sinal avisa a BCU (156) para fazer a busca de ordem proveniente da caixa de correio (188) e para a executar. O pedido é repostado pela BCU depois da execução da ordem. O estado do pedido pode ser detectado pela PE (85).

A BCU faz um pedido quando surge um problema, ou durante a execução de uma ordem iniciada pela PE (85), ou em qualquer outro instante. Ela provoca uma excepção na PE (85), se não for selectivamente mascarada.

O adaptador (154) adapta a velocidade de transferência dos canais assíncronos (0,1) do adaptador à li



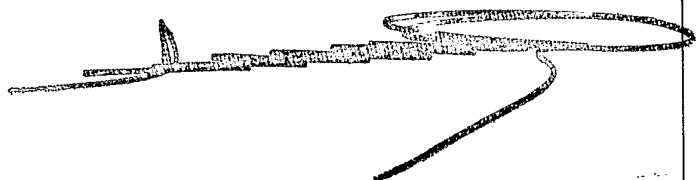
nha omnibus síncrona (170) do processador. Portanto, a BCU (156) é apoiada por memórias tampão de dados de 64 octetos (259) e (260), no adaptador (154), para a transferência de dados para e da BCU (156), respectivamente. O agregado tem uma porta de 4 octetos para a linha omnibus do canal (0,1) e para a linha omnibus (170) do processador.

Os registadores síncronos (113) e (114) armazenam temporariamente dados transferidos entre a BCU (156) e os agregados de memórias (260) e (269).

Os registadores de recepção e de emissão da linha omnibus armazenam dados recebidos de e transferidos para a linha omnibus (170) do processador, respectivamente.

Uma operação de armazenamento (armazenamento de dados de I/O, fila de espera Op) é iniciada pela emissão da BCU (156) para o adaptador (154) da contagem de comandos/octetos, da chave de protecção e do endereço da memória, através da linha omnibus do canal. A contagem de comandos/octetos é recebida na linha omnibus de comandos (252) (fig. 13) e armazenada no registador (125) de comandos. Os dados da chave e do endereço são recebidos da BCU (156) através da linha omnibus (251) de endereços/dados (fig. 13) e armazenados no registador (123) da chave/endereço (123). O agregado de indicadores de endereços de escrita e leitura são ajustados nos seus valores iniciais no registador (128). O número de transferências de dados (2 octetos de cada vez) na linha omnibus (251) é determinado pela contagem de octetos. Com uma operação de armazenamento, podem transferir-se até 64 octetos de dados. O endereço de memória de qualquer octeto dentro de uma operação de armazenamento pode não ultrapassar o limite de endereços de 64 octetos.

O ciclo de comando/endereços é seguido por ciclos de dados na linha omnibus (251). Todos os dados são recolhidos na memória tampão de 64 octetos (260). Depois de terem sido recebidos os últimos dados da BCU (156), o adaptador (154) efectua uma primeira verificação (não representada) de prioridade interna para as duas memórias tampão de dados (259) e (260) e depois pede o controlo (não representado) da linha omnibus (170) do processador, tendo o adaptador (154) aí o grau



de prioridade mais elevado.

No caso de ambas as memórias (259) e (260) pedirem uma transferência ao mesmo tempo, o controlo interno de prioridade concede-a primeiro à memória tampão (259) e depois, sem ciclo de arbitragem, à memória (260), isto é: as leituras têm prioridade sobre as escritas.

Quando for concedido o controlo, transferem-se a contagem de comandos/octetos, a chave de protecção e o endereço inicial para o controlador da memória "cache" (153). O ciclo de transferência de comandos é seguido pelos ciclos de transferências de dados.

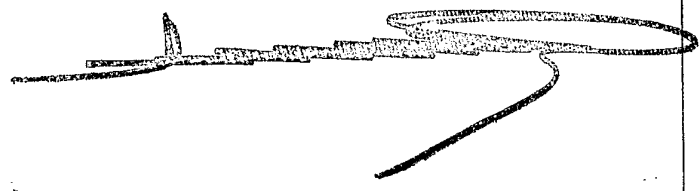
O controlador da memória "cache" (153) efectua a verificação da chave de protecção. Uma violação da chave será comunicada ao adaptador (154) no estado da linha omnibus (170). Outras condições de verificação pelo controlador da memória "cache" (153) e da memória principal (162) são comunicadas como estado ANY-CHECK. Uma violação da chave e condições de estado detectadas pelo adaptador (154) serão enviadas para a BCU (156) num ciclo de transferência de estados.

Há duas condições possíveis de estado detectadas pelo adaptador (154) que podem ser comunicadas à BCU (156). Para ambas as condições de verificação é suprimido o acesso à memória (162).

Cada endereço de memória principal recebido da BCU (156) é comparado com o endereço mantido no registador ACB para determinar se o acesso é para a área IOA (187) ou para a área de cliente (165) da memória (162). Um bit "cliente" recebido juntamente com cada comando da BCU (156) determina se o acesso à memória principal se destina à área IOA (187) ou à área de cliente ou utilizador (165) e controla os acessos impróprios.

Uma condição BNA (memória tampão não disponível) descrita mais adiante é comunicada apenas para operações de fila de espera.

As operações de leitura (leitura de I/O, leitura da caixa de correio) são iniciadas pela BCU (156) de uma maneira essencialmente idêntica à das operações de memória ou armazenamento. Logo que a contagem de comandos/octetos, a chave de



protecção e o endereço tenham sido recebidos da BCU (156), efectua-se a verificação de prioridade interna no adaptador (154) e é pedido o controlo da linha omnibus do processador (170). Se este controlo for concedido, a contagem de comandos/octetos, a chave de protecção e o acesso inicial da memória principal são transferidos para o controlador da memória "cache" (153) para iniciar o ciclo de leitura. O adaptador (154) carrega os dados pedidos primeiro na sua memória tampão (259) e depois, a pedido da BCU, através da linha omnibus (250), para a BCU (156). É comunicado o estado com cada transferência de dados.

As condições de estado e o mecanismo de comunicação para as operações de memória aplicam-se às operações de leitura.

A PE (85) pode aceder à maior parte dos registadores no adaptador (154) com operações de detecção (leitura) e de controlo (escrita) através da linha omnibus (170).

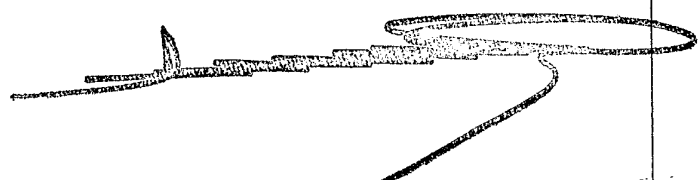
Para as operações de detecção, o comando é transferido para o adaptador (154) e retido no registador (129). No ciclo seguinte, selecciona-se o multiplexador de detecção (126) de acordo com o comando; e o comando é carregado no registador de emissão BSR (116) para ter os dados esperados válidos no ciclo seguinte da linha omnibus (170).

Se se detectar um erro de paridade interno no registador a detectar, o adaptador (154) emite dados com a paridade correcta para trás para a PE (85), mas provoca uma condição de verificação na linha omnibus da chave/estado. Esta função pode ser ensaiada com uma posição de código de detecção específica.

Para as operações de controlo, o comando BUS 170 será seguido por dados, que são carregados no registador de destino no ciclo seguinte.

Se se detectar um erro de paridade na linha omnibus (170) no ciclo de comando para operações de detecção ou de controlo ou no ciclo de dados para operações de controlo, o adaptador (154) força uma paragem do relógio.

O registador da base (110) contém o endereço base usado para endereçar a fila de espera e a mala de



correio e o código do comprimento da fila de espera. A fila de espera começa no endereço da base, a área da caixa de correio na base + comprimento da fila de espera.

Os registradores RPNTR e WPNTR (111) e (112) contêm o desvio do endereço da base para a entrada seguinte da fila de espera a que deve aceder-se para leitura e escrita, respectivamente.

Quando detectados, o indicador de leitura e o indicador de escrita são concatenados com o endereço de base pelo multiplexador de detecção (126) no adaptador (154). Portanto, a palavra devolvida pela operação detectada é o endereço completo da entrada seguinte da fila de espera a que deve aceder-se.

O registor de estados I/O contém os seguintes bits (além de outros, não descritos aqui):

Qualquer verificação ("Any Check") (bit 0) - ajustado a 1, se qualquer verificação em CHSR <0..24> e correspondente ao bit CHER for. 1. "Any Check" provoca ATTN-REQ. Se MODE-REQ <1> = 1, então activa-se o sinal Clock-Stop-Diana.

Enviado BNA ("BNA sent") (bit 6) - bit BNA (memória tampão não disponível) é 1, quando BCU tem de armazenar uma mensagem entrada na fila de espera e a fila de espera está cheia, isto é, RPNTR igual a WPNTR + 16. Este bit só pode ser repostado escrevendo um 1 no registor CW (120), bit 6.

Fila de espera não vazia (bit 7). Este bit é 1 se RPNTR não for igual a WPNTR. É 0 se RPNTR = WPNTR. É este o meio usado para notificar o processador (85) de que foi recebida uma nova mensagem.

Pedido de BCU para PU ("BCU to PU Request") (bits 10 e 14) - Ajustados pelo BCU (156) através do sinal na linha "BCU to PU Request" (256c) para os canais 0 e 1. A reposição dos bits 10 e 14 pelo PE (85) produz uma confirmação "BCU to PU" na linha (256d) para os canais 0 e 1.

Pedido de PU para BCU ("PU do BCU Request") - Ajustado na linha (226a) pela PE (85), ajustando o bit 11 do registor CW (120) para o canal 0 e o bit 15 para o registor CW (120) para o canal 1. Reposição pelo sinal "PU to

PCU Acknowledge na linha (256b).

Falta de energia em BCU ("BCU powerloss") (bit 13) - Este bit é ajustado para 1 pela BCU (156) quando ela perde a sua energia ou quando se verificar um "power on reset" (reposição da energia). É repostado em 0 se se inscrever um 1 no bit "Reset BCU powerloss" do registador CW (120) e a BCU já não estiver no estado de sem energia.

Permitir a arbitragem ("Allow Arbitration") (bit 29) - Este bit activa o sinal "Allow Arbitration" da linha omnibus do caso de o bit 3 do registador de modos do adaptador estar inactivo.

O bit de acesso do utilizador, que faz parte do comando/endereço recebido da BCU (156), determina se o acesso à memória será na área de memória IOA ou na de utilizador. Se o bit de acesso do utilizador for "0", o endereço da página para acesso à memória tem de estar dentro da área IOA (187). Não será feita qualquer verificação da chave para estes acessos, forçando por conseguinte o equipamento físico do adaptador a chave a ser zero (adapta-se a entradas de todas as chaves).

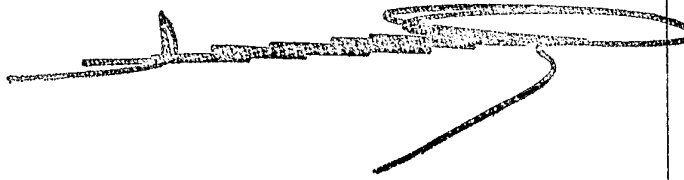
Se o acesso de utilizador for "1", o endereço da página para acesso à memória tem de estar no interior da área de memória do utilizador (165). Caso contrário é provocada uma condição de verificação ACB para o acesso.

A PE (85) utiliza comandos de mensagens para ler (detectar) ou para escrever (controlar) os registadores do adaptador (154).

O formato para estes comandos é o seguinte:

bits 0 - 7	CMD	= Tipo de comando
8 - 11	SRC	= pedido de endereço da unidade de linha omnibus
12 - 15	DST	= dados a transmitir no ciclo CMD
24 - 27	REG1	= número do registador para CONTROL
28 - 31	REG2	= número do registador para SENSE

O campo DST para a interface PU-BCU é X'8'. O adaptador (154) não descodificará o campo SRC e MSG visto que não há qualquer informação contida para execução de co-



mandos. Durante as operações de controlo e detecção, os bits REG1 e REG2 definirão respectivamente o registador no adaptador (154) a escrever para o mesmo e a ler do mesmo.

2. LINHA OMNIBUS DOS CANAIS 0 E 1 DO ADAPTADOR DE I/O (fig. 16)

O canal 0 do adaptador e o canal 1 do adaptador são interligações de alta velocidade do adaptador de I/O (154) para a unidade de controlo de linhas omnibus (156).

O canal 0 inclui:

Linhas omnibus de endereços/dados (250)
(bits 0-16, PO-P1)

Linha omnibus comando/estado (249)
(bits 0-3,P)

"Tag up" (BCU para Memória tampão) linha (262a)

"Tag Down" (Memória tampão para BCU) linha (262b)

"Pedido PU para BCU" linha (256a)

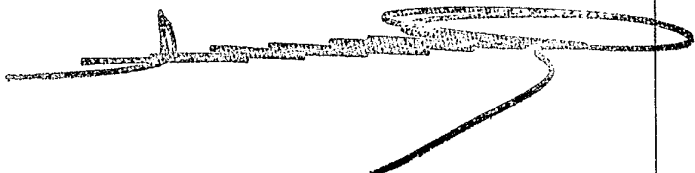
"Confirmação BCU para PU" linha (256b)

O canal 1 inclui uma linha omnibus de endereços/dados (251), uma linha omnibus de comando/estado (252) e linhas "Tag up" e "Tag down" (262c) e (262d).

O canal 0 é usado para transferências de dados da memória S/370 (162) (e PE (85)) para BCU (156) e o canal 1 é usado para transferências de dados da BCU (156) para a memória (162) (e PE (85)).

As linhas omnibus (249), (250), (251) e (252) dos canais originam-se no adaptador I/O (154), que é essencialmente um par de memórias tampão de dados que controla a lógica capaz de armazenar até 64 octetos de dados cada uma. As linhas omnibus terminam na BCU (156). O adaptador I/O (154) serve para acelerar a adaptação entre a linha omnibus (170) interna S/370 com o seu formato de palavras completas (32 bits) e as linhas omnibus mais lentas (249) a (252) com o seu formato de meias palavras (16 bits).

Cada canal está organizado em duas par-



tes, a linha omnibus de dados (250,251) do lado de dois octetos (meia palavra) e a linha omnibus (249,252) de comando/estado do lado de meio octeto (4 bits). Sinais de etiquetas ("Tag signals") proporcionam os meios para controlar as operações através de pedido/resposta e sinais especiais.

A transferência de dados por cada canal verifica-se sempre em dois ciclos para transferir quatro octetos por cada linha omnibus de dois octetos). Logicamente, toda transferência de dados é entre a memória principal S/370 (162) e o subsistema I/O que inclui a BCU (156). A BCU (156) é a unidade directora, isto é, ela inicia todas as operações de transferência uma vez que a PE (85) tenha assinalado a necessidade da mesma.

A linha omnibus de comandos/estados (249,252) é usada durante o ciclo de selecção para definir o sentido da transferência (busca/armazenamento) e a quantidade de dados a transferir. A linha omnibus de endereços/dados (250/251) serve para transferir o endereço da memória principal durante o ciclo de selecção e fornece dados durante um ciclo de transferência actual. É também usada para indicar áreas específicas (188,189) na memória (162) conhecidas por "caixa de correio" e "fila de espera de mensagens". Estas áreas permitem à PE (85) permutar certa informação com a BCU (156).

Durante uma operação de busca (a partir da memória (162)), o estado é transferido pela linha omnibus de comando/estado (249) juntamente com os dois primeiros octetos de dados pela linha omnibus (250). Este estado indica qualquer verificação de endereços, verificação da chave, etc. ou é zero para indicar uma operação sem êxito.

Se se efectuar uma operação de armazenamento (na memória (162)), segue-se um ciclo de estado depois de todos os dados terem sido fornecidos à memória principal (162).

As fig. 14A e 14B mostram a utilização lógica das partes da linha omnibus durante o subciclo 1 e o subciclo 2 de operações de busca e de armazenamento, respectivamente, sendo:

aaa... endereço do primeiro octeto (o octeto mais



à esquerda) no campo de dados

A: 1 = verificação de endereço

B: 1 = memória tampão não disponível

c: 1 para acesso à memória do utilizador (165),
0 para acesso à área de microcódigo (IOA 187)

ddd... 4 octetos de dados para/da memória

fff... comprimento do campo menos 1, em octetos (0...63 decimal)

kkkk 1 = verificação da chave

oooo: desvio dentro de 32 octetos área da caixa de correio

pp prioridade (0...3, 3 é a mais elevada)

.... não considerar

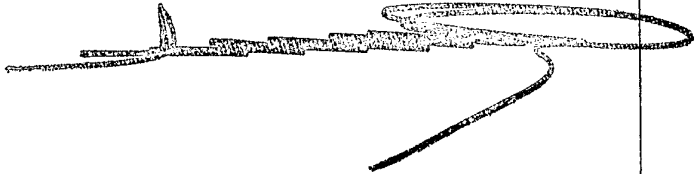
///: linha omnibus em flutuação (indefinida)

in "inbound" (entrada (BCU para memória tampão))

out "outbound" (saída) (memória tampão para BCU)

Utilizam-se as seguintes linhas de etiqueta para as operações de transferência:

1. A linha de pedidos de PU para BCU (256a) do adaptador de linhas omnibus (154) para a BCU (156) é usada pela PE (85) para indicar a necessidade de uma operação de I/O. Uma vez ajustado, o sinal mantém-se activo até ser repostado pela BCU (156).
2. A linha "Tag up" (etiqueta no sentido ascendente) (262a) da BCU (156) para o adaptador (154) é usada para pedir dados de saída do adaptador (154) ou para indicar que há dados de entrada disponíveis na linha omnibus. A linha "Tag up" (262c) funciona da mesma maneira.
3. A linha "Tag down" (etiqueta no sentido descendente) (262b) do adaptador (154) para a BCU (156) é usada para indicar uma falta temporária de dados para a BCU (156), se esta situação existir. O flanco descendente do sinal TAG DOWN indicará então a disponibilidade de dados de saída na linha om-



nibus. A linha "Tag Down" (262d) funciona da mesma maneira.

4. A linha de Confirmação BCU para PU (256b), da BCU (156) para o adaptador (154), é usada para repor o sinal de pedido de PU para BCU. Esta reposição é efectuada quando se tiver completado a operação de I/O para a caixa de correio.

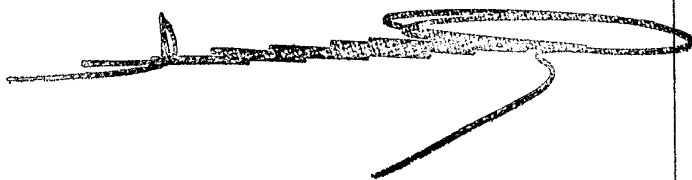
Quando a PE (85) detectar uma instrução Start I/O (iniciar entrada/saída) (SIO) na corrente de intruções, ela alerta o subsistema I/O, isto é a BCU (156), sobre a necessidade de uma operação I/O, activando a linha "PU to BCU Request" (pedido da PU para a BCU). Esta etiqueta faz com que a BCU (156) consulte a "caixa de correio" (188) no interior da memória (162) para verificar se esta operação é uma busca ou um armazenamento, quantos octetos é necessário transferir, etc.. A caixa de correio contém efectivamente o canal SIO, CUA, CAW e a palavra de comando (CCW) da operação I/O pertinente.

As operações de armazenamento são geralmente aquelas em que a BCU (156) envia dados para a PE (85). Estes "dados" são ou comando/chave/endereço, que são enviados no ciclo de selecção, ou dados I/O "reais" a armazenar na memória principal (162). Em ambos os casos a sequência de eventos é a mesma.

As fig. 1A a 15C ilustram esquematicamente, numa forma generalizada, para a descrição seguinte, a maneira como a informação de dados e de estados é encaminhada para dentro e para fora de memórias tampão/registadores de trinta e dois bits no adaptador (154) e na BCU (156) e como os bits, os de ordem superior (à esquerda) e os de ordem inferior (à direita), de informação são colocados nas linhas omnibus dos canais 0 e 1 de dezoito bits do adaptador (154).

As fig. 25 e 26 ilustram um conjunto específico de sinais para as transferências de dados entre a BCU (156) e o adaptador (154).

Com o início de um ciclo do relógio BCU, durante uma operação de armazenamento, fig. 15A, a BCU (156) coloca os dados para o primeiro ciclo na linha omnibus (251), Se este ciclo for um ciclo de dados para uma operação de dados da memória principal, coloca-se na linha omnibus de comandos/esta-

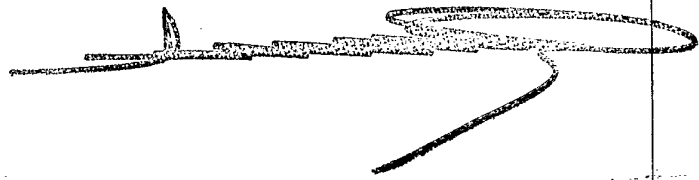


dos (252) e na linha omnibus de endereços e dados (251), respectivamente um comando, uma contagem de octetos, uma chave de ecesso e o primeiro octeto do primeiro endereço da memória principal. Se o ciclo for o ciclo de selecção para uma consulta da caixa do correio, não se coloca qualquer endereço da memória principal, visto que o comando indica a caixa do correio, que está numa posição fixa. Mantém-se válido o primeiro subciclo na linha omnibus para o tempo de dois subciclos.

Um ciclo de relógio da BCU após a colocação dos dados na linha omnibus (251) durante um ciclo de selecção, a BCU (156) activa a linha do sinal "Tag Up". A linha "Tag Up" (262a) faz com que o adaptador (154) armazene os dois primeiros octetos na metade da esquerda do registador (113). Com o início do ciclo de relógio seguinte, a BCU (156) coloca os dados (os segundos dois octetos) para o subciclo seguinte na linha omnibus de endereços/dados (251) para armazenamento, na outra metade do registador (113) do adaptador (154). Estes dados são ou o restante de um endereço da memória principal ou um desvio (se o lance pertence a um ciclo de selecção de consulta da caixa do correio). A BCU (156) retém os segundos dois octetos durante três ciclos do relógio da BCU, depois faz cair o sinal "Tag up".

As operações de busca são geralmente aquelas em que a BCU (156) pede dados do espaço de dados (162) da memória principal, da área de microcódigos na memória (162), ou da caixa de correio ou da fila de espera de mensagens. Em qualquer caso, um ciclo de selecção tem de preceder uma tal operação de busca para instruir a lógica do adaptador (154) acerca da operação que ele tem de executar. O ciclo de selecção é realizado colocando os comando/chave/endereço na linha omnibus (249) de uma maneira análoga à operação de armazenamento utilizando a linha omnibus (252), excepto que o comando na linha omnibus de comandos/estados (249) é um comando de "busca".

Com o início do ciclo de relógio seguinte (depois de se completar o ciclo de selecção) a BCU produz o sinal "Tag up" e mantém-no durante três ciclos de relógio BCU (fig. 15B). O sinal "Tag up" pede dados da memória tampão. Os



dados estarão disponíveis um ciclo mais tarde, se a memória tampão puder fornecer dados. Como o funcionamento é semi-síncrono, a BCU (156) admite que os dois primeiros octetos de dados sejam mantidos válidos na linha omnibus durante dois ciclos, há depois um tempo de comutação de um ciclo e depois disso podem encaminhar-se os dois segundos octetos de dados para a BCU (156).

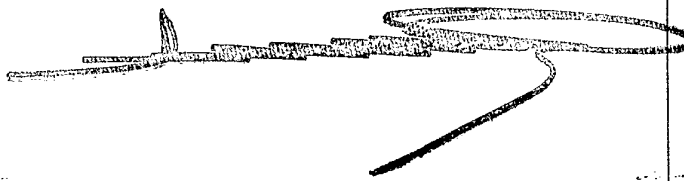
Contudo, há situações nas quais o adaptador (154) não tem dados disponíveis no instante em que surge o sinal "Tag up". Isso sucede tipicamente numa busca "inicial" de dados, isto é, quando se buscam dados a partir de um novo endereço levando algum tempo até que o pedido de busca seja processado através do controlador da memória "cache" (153) e do controlador da memória (155), voltando depois ao adaptador (154). Uma nova tentativa na memória principal (162) pode também provocar um atraso temporário.

Sempre que o adaptador (154) não possa fornecer dados (fig. 15C), excita a linha "Tag Down" logo que seja detectado o sinal "Tag up". A BCU (156) deve fazer a amostragem da linha "Tag Down" não depois de cinco ciclos após ser gerado o sinal "Tag up".

O adaptador (154) mantém o "Tag Down" até estar disponível a primeira palavra de dados (quatro octetos). Nesse instante, o adaptador (154) coloca os dois primeiros octetos na linha omnibus (250) e faz cair o sinal "Tag Down". O flanco descendente do sinal "Tag Down" dispara a lógica (253) da BCU.

A BCU (156) admite que os primeiros octetos são válidos durante os dois ciclos seguintes à queda do sinal "Tag Down" e depois ficam disponíveis os dois segundos octetos. Conforme o valor da contagem feita durante o ciclo de selecção até 60 octetos, podem seguir-se dois octetos de cada vez.

Quando todos os dados da caixa de correio que foram pedidos no ciclo de selecção tiverem sido recebidos, a BCU (156) estabelece o sinal "BCU to PU Acknowledge" (confirmação da BCU para a PU) na linha (256b) para o adaptador (154) para repor o pedido da PU para a BCU na linha (256a) que iniciou a operação.



A maior parte da transferência de informação entre a PE (85) e a BCU (156) é feita através de posições pré-definidas da memória (188,189) utilizando o endereço de base e o comprimento da fila de espera armazenados no registador de base (110) no adaptador (154). A fila de espera das mensagens entradas (189) armazena todas as mensagens emitidas pela BCU por ordem cronológica.

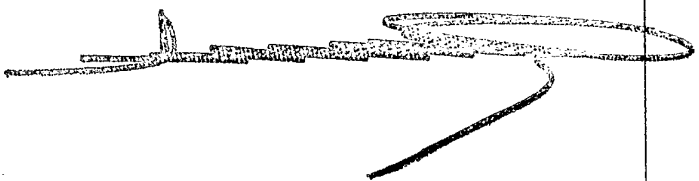
3. Unidade de controlo de linhas omnibus (156) - Descrição Geral (Figuras 16 e 17)

A unidade de controlo das linhas omnibus (BCU) (156) é o equipamento físico primário de acoplamento entre o processador S/370 (85) e o seu processador S/88 associado (62) que é utilizado para efectuar as operações de I/O S/370.

A BCU (156) inclui meios que interagem com um programa de aplicação (EXEC370) e um microcódigo (ETIO) que é executado no processador do S/88 (62) para apresentar interrupções ao processador (62) e para desacoplar assincronamente o processador (62) do seu equipamento físico associado e para acoplar o processador (62) à unidade BCU (156), tudo transparente para o sistema operativo S/88. As operações de interrupção e desacoplamento transparentes são utilizadas para permitir o acoplamento directo dos processadores S/370 e S/88 para a transferência eficiente de comandos e dados I/O S/370 do processador S/370 (85) para o processador (62) S/88 para a conversão dos comandos e dados para uma forma utilizável pelo processador S/88 (62) para efectuar as operações I/O S/370.

Compreender-se-á que o EXEC370 e o ETIO podem ambos ser um microcódigo ou um programa de aplicação e ser armazenados na memória (174) ou na memória "cache" (173).

A BCU (156), fig. 16, inclui a lógica da interface da unidade de controlo de linhas omnibus e registadores (205), um controlador do acesso directo à memória (DMAC) (209) e uma memória local (210). Linhas omnibus locais de endereços e de dados (247) e (223) acoplam a memória (210) ao endereço da PE (62), às linhas omnibus de dados (161A,161D) através



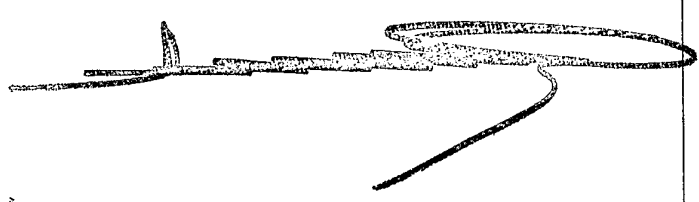
dos circuitos de excitação/recepção (DRVR/RCVR) (217,218) e à lógica de interface (205). O DMAC (209) está acoplado às linhas omnibus de endereços (247) através de retentores (233) e à linha omnibus de dados através dos circuitos de excitação/recepção (234).

O DMAC (209) tem quatro canais, 0 a 3, que estão acoplados à lógica de interface (205) (fig. 17) por trajectos respectivos de pedido e de confirmação, cada um deles dedicado a uma função especial: o canal 0 transfere comandos I/O S/370 de uma área de caixa do correio (188) (fig. 28) na memória S/370 (162) para a memória local (210) (LEITURA DA CAIXA DE CORREIO). O canal 1 transfere dados S/370 da memória (210) para a memória (162) (Leitura I/O S/370). O canal 3 transfere mensagens S/88 de elevada prioridade da memória (210) para a área (89) da fila de espera de mensagens (fig. 28) na memória (162) (ESCRITA DE MENSAGENS NA FILA DE ESPERA).

O adaptador de linhas omnibus (154) tem dois canais 0 e 1. O canal 0 do adaptador trata as funções de LEITURA DA CAIXA DE CORREIO e ESCRITA I/O S/370 dos canais 0,1 do DMAC (isto é, o fluxo de dados do S/370 para a BCU (156)). O canal 1 do adaptador trata as funções de LEITURA I/O S/370 e ESCRITA NA FILA DE ESPERA DE MENSAGENS dos canais 2, 3 da DMAC (isto é, o fluxo de dados de BCU (156) para S/370).

4. Controlador do acesso directo à memória (209)

O controlador do acesso directo à memória (DMAC) (209) é de preferência do tipo descrito (MC68450) no M68000 Family Reference Manual, FR68K/D, Copyright Motorola, Inc. 1988. O referido manual é aqui incorporado por referência como se fosse transcrito na sua totalidade. O DMAC (209) é concebido para complementar a eficácia e as capacidades arquitecturais dos microprocessadores da família M68000 Motorola (tal como o elemento processador M68020 (62)) deslocando blocos de dados de uma maneira rápida e eficiente com o mínimo de intervenção de um processador. O DMAC (209) efectua transferências de dados de memória para memória, de memória para um dispositivo e de um



dispositivo para uma memória.

Inclui quatro canais de acesso directo à memória (DMA) independentes, com prioridade programável, e utiliza a estrutura da linha omnibus M68000 assíncrona com uma linha omnibus de endereços de 24 bits e de dados de 16 bits. Pode ser endereçado explicita e implicitamente.

A finalidade principal de um DMAC tal como (209) em qualquer sistema é transferir dados a velocidade muito elevadas, usualmente muito mais rapidamente que o que um microprocessador pode tratar sob o controlo de meios de programação. A expressão acesso directo à memória (DMA) é usado para designar a capacidade de um dispositivo periférico aceder à memória num sistema da mesma maneira que o faz um microprocessador. A memória na presente aplicação é a memória local (210). O funcionamento do DMA pode verificar-se simultaneamente com outras operações que o sistema tenha necessidade de efectuar, reforçando assim grandemente a eficácia geral do sistema.

O DMAC (209) desloca blocos de dados com velocidades que se aproximam dos limites da linha omnibus local (223). Um bloco de dados consiste numa sequência de operandos, de um octeto, de uma palavra ou de uma palavra longa, que se inicia num endereço específico na memória com o comprimento do bloco determinado por um valor de contagem de transferência. Uma operação num canal único pode implicar a transferência de vários blocos de dados para ou da memória (210).

Qualquer operação que envolva o DMAC (209) seguirá os mesmos passos básicos: inicialização do canal por PE (62), transferência de dados e terminação do bloco. Na fase de inicialização, o processador PE (62) carrega os registadores do DMAC com informação de comando, indicadores de endereços e contagens de transferência, fazendo depois arrancar o canal. Durante a fase de transferência, o DMAC (209) aceita pedidos para transferências de operandos e proporciona controlo dos endereços e das linhas omnibus para as transferências. A fase de terminação ocorre depois de a operação estar completa, quando o DMAC indica o estado da operação no registador de estados CSR. Durante todas as fases de uma operação de transferência de



dados, o DMAC (209) estará num de três modos de operação:

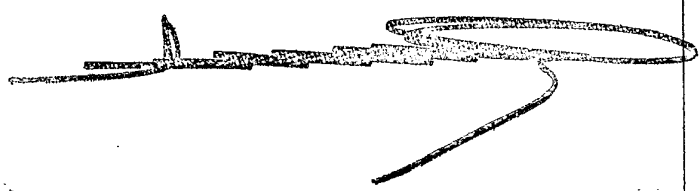
1. IDLE - Este é o estado que o DMAC (209) toma quando é repos to por um dispositivo exterior e aguarda a inicialização pe lo processador do sistema (62) ou um pedido de transferên cia de operandos de um periférico.
2. MPU - Este é o estado em que DMAC (209) entra quando é se leccionado ao nível da micropastilha por uma outra linha om nibus no sistema, que funciona como primária (usualmente o processador principal do processador (62)). Neste modo, os registadores internos do DMAC são gravados ou lidos, para controlar o funcionamento do canal ou verificar o estado de uma transferência de blocos.
3. DMA - Este é o estado em que entra o DMAC (209) quando está a actuar como uma linha omnibus considerada como primária para efectuar uma transferência de operandos.

O DMAC pode efectuar transferência de dados de endereços implícitos ou explícitos. Para as transferên cias explícitas, os dados são transferidos de uma fonte para um registador de retenção interno DMAC, e depois, no ciclo seguin te da linha omnibus, são deslocados do registador de retenção para o destino. As transferências implícitas exigem apenas um ciclo da linha omnibus porque os dados são transferidos directa mente da fonte para o destino, sem memorização intermediária in terna no DMAC.

Há três tipos de operações dos canais:

1) transferências de um bloco único, 2) funcionamento contínuo e 3) operações encadeadas. Quando se transferem blocos de dados isolados, são inicializados os registadores (MAR) e (DAR) de en dereços de memória e de endereços de dispositivos, pelo utiliza dor, para especificar a origem e o destino da transferência. É também inicializado o registador de contagem de transferências da memória para contar o número de operandos transferidos num bloco.

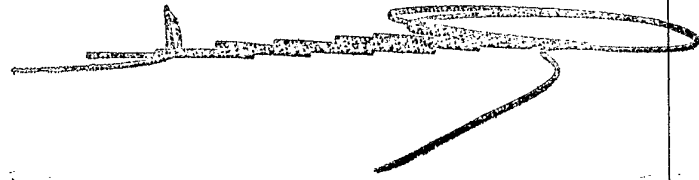
Os modos de encadeamento são: encadea mento de campos de dados ("array") e encadeamento de campos de dados ligados. O modo de encadeamento de campos de dados opera a partir de um campo de dados contíguo na memória (210) que con



siste em endereços de memória e valores de contagem de transferência. O registrador do endereço de base (BAR) e o registrador de contagem de base de transferência (BTC) são inicializados para apontar para o endereço inicial do campo de dados e o número de entradas do campo de dados, respectivamente. Quando cada transferência de blocos estiver completada, é buscada a entrada seguinte a partir do campo de dados, decrementa-se a contagem de transferências e incrementa-se o endereço de base para apontar para a entrada do campo de dados seguinte. Quando a contagem de transferências atingir zero, a entrada precisamente procurada é a transferência do último bloco definido neste campo de dados.

O modo de campos de dados encadeados ligados é semelhante ao modo de encadeamento de campos de dados, excepto que cada entrada do campo de dados na memória contém um endereço de ligação que aponta para a entrada seguinte no campo de dados. Isso permite um campo de dados na memória não contíguo. A última entrada contém um endereço de ligação ajustado a zero. O registrador BTC da contagem de transferências de base não é necessário neste modo. O registrador de endereços de base (BAR) é inicializado para o endereço da primeira entrada no campo de dados. O endereço de elos de ligação é usado para actualizar o registrador de endereços de base no início de cada transferência de blocos. Este modo de encadeamento permite que as entradas do campo de dados sejam facilmente deslocadas ou inseridas sem ter de reorganizar o campo de dados por ordem sequencial. Também, o número de entradas no campo de dados não precisa de ser especificado para o DMAC (209). Este modo de endereçamento é usado pelo DMAC (209), no presente pedido de patente, para aceder a blocos livres da fila de espera de trabalho a partir de uma lista de elos de ligação, de uma maneira que se descreve mais adiante com mais pormenor.

O DMAC (209) interromperá o PE (62) para um certo número de casos que ocorrem, tais como a completude de uma operação DMA, ou quando do pedido de um dispositivo usando uma linha PCL (57 a-d). O DMAC (209) retém os vectores de interrupção, em oito registadores de vectores na micropastilha,



para serem usados na estrutura de interrupções por vectores PE (62). Estão disponíveis por cada canal dois registadores de vectores, o vector de interrupção normal (NIV) e o vector de interrupção de erro (EIV).

A cada canal é atribuído um nível de prioridade de 0, 1, 2 ou 3, isto é, aos canais 0, 1, 2 e 3 são atribuídos os níveis de prioridade 0, 2, 2, 1, respectivamente (sendo o nível de prioridade 0 o mais elevado).

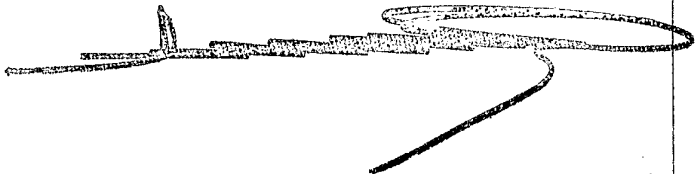
Os pedidos são gerados exteriormente por um dispositivo ou gerados internamente pelo mecanismo de auto-pedido do DMAC (209). Os auto-pedidos podem ser gerados ou a uma velocidade máxima, caso em que o canal tem sempre um pedido pendente, ou com uma velocidade limitada, determinada seleccionando uma parte da largura de banda da linha omnibus para ficar disponível para a actividade DMA. Os pedidos exteriores podem ser pedidos que roubam ciclos, que são gerados pelo sinal de pedido associado com cada canal.

O DMAC (209) contém 17 registadores (fig. 18) para cada um dos quatro canais mais um registador de controlo geral (GCR), todos eles sob o controlo dos meios de programação.

Os registadores do DMAC (209) contêm informação acerca das transferências de dados, tais como o endereço da origem e do destino e códigos de funções, contagem de transferências, dimensões dos operandos, dimensão das portas do dispositivo, prioridade do canal, endereço de continuação e contagem de transferências e o funcionamento da linha de controlo dos periféricos. Um registador CSR também proporciona informação de estado e de erros sobre a actividade do canal, das entradas dos periféricos, e de vários eventos que podem ter ocorrido durante uma transferência DMA. O registador de controlo geral GCR selecciona o factor de utilização da linha omnibus a usar em operações de auto-pedido DMA de velocidade limitada.

Os sinais de entrada e saída estão organizados funcionalmente nos grupos como adiante se descreve (ref. fig. 19A).

A linha omnibus de endereços/dados (A8-



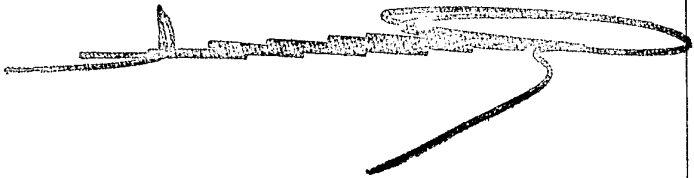
-A23,D0-D15) (248), uma linha omnibus de 16 bits, é multiplexada temporalmente para proporcionar saídas de endereços durante o modo de funcionamento DMA e é utilizada como linha omnibus de dados bidireccional para introduzir dados provenientes de um dispositivo exterior (durante uma escrita em PE (62) ou uma letura em DMAC) para fazer a saída de dados para um dispositivo exterior (durante uma leitura de PE (62) ou uma escrita em DMAC). Esta linha omnibus é uma linha omnibus de três estados e é desmultiplexada utilizando retentores e memórias tampão exteriores (233,234), controlados pelas linhas OWN e DDIR de controlo do multiplexador.

As linhas (A1) a (A7) dos endereços inferiores da linha omnibus (247) são linhas bidireccionais com três estados e são usadas para endereçar os registadores internos DMAC, no modo MPU, e para proporcionar as sete saídas dos endereços inferiores, no modo DMA.

As linhas de códigos de funções (FC0) a (FC2) são linhas de saída de três estados e são usadas, no modo DMA, para qualificar mais o valor na linha omnibus de endereços (247) para proporcionar espaços de endereços separados que podem ser definidos pelo utilizador. O valor colocado nestas linhas é retirado de um dos registadores internos de códigos de funções MFC, DFC, BFC, conforme o registador que proporciona o endereço usado durante um ciclo da linha omnibus DMA.

Linhas de controlo das linhas omnibus assíncronas controlam as transferências assíncronas de dados usando os sinais de controlo seguintes: selecção, impulsos de sincronização de endereços, leitura/escrita, impulsos de sincronização de dados superiores e inferiores e confirmação de transferência de dados. Estes sinais são descritos nos parágrafos seguintes.

A linha de entrada SELECT (selecção) (296) é usada para seleccionar o DMAC (209) para um ciclo da linha omnibus MOU. Quando for identificado, o endereço em A1-A7 e os impulsos de sincronização de dados (ou A0 quando se usar uma linha omnibus de 8 bits) selecciona o registador interno DMAC que será implicado na transferência. O sinal SELECT deve ser ge-



rado qualificando um sinal de descodificação do endereço com os impulsos de sincronização de endereços e de dados.

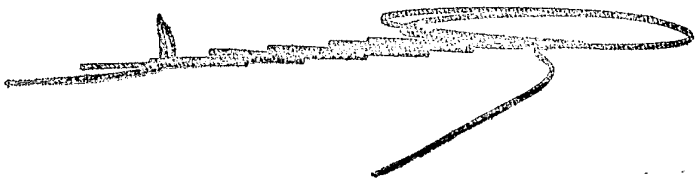
O sinal AS (ADDRESS STROBE) na linha (270b) é um sinal bidireccional usado como uma entrada, no modo DMA, para indicar que está presente um endereço válido na linha omnibus de endereços (161). Nos modos MPU ou IDLE, é usado como entrada para determinar quando o DMAC pode tomar o controlo da linha omnibus (se o DMAC tiver pedido e lhe tenha sido concedido o uso da linha omnibus).

READ/WRITE (leitura/escrita) é um sinal bidireccional (não representado) usado para indicar o sentido de uma transferência de dados durante um ciclo da linha omnibus. No modo MPU, um nível elevado indica que uma transferência é do DMAC (209) para a linha omnibus de dados (223) e um nível baixo indica uma transferência da linha omnibus de dados para o DMAC (209). No modo DMA, um nível elevado indica uma transferência da memória endereçada (210) para a linha omnibus de dados (223) e um nível baixo indica uma transferência da linha omnibus de dados (223) para a memória endereçada (210).

As linhas bireccionais UPPER AND LOWER DATA STROBE (não representadas) indicam quando os dados são válidos na linha omnibus e que partes da linha omnibus devem estar implicadas numa transferência (D8-15 ou D0-7).

A linha bidireccional DATA TRANSFER ACKNOWLEDGE (DTACK) (265) é usada para sinalizar que um ciclo da linha omnibus assíncrona pode ser terminado. No modo MPU, esta saída indica que o DMAC (209) aceitou dados da PE (62) ou colocou dados na linha omnibus para a PE (62). No modo DMA, esta entrada (265) é vigiada pela DMAC para determinar quando termina um ciclo da linha omnibus. Enquanto DTACK (265) se mantiver negado, o DMAC inserirá ciclos de espera no ciclo da linha omnibus e quando DTACK (265) for detectado, o ciclo da linha omnibus será terminado (excepto quando PCL (257) for usado como sinal de prontidão, caso em que ambos os sinais têm de ser detectados antes de o ciclo ser terminado).

Sinais de controlo multiplex nas linhas
OWN e DDIR são usados para controlar dispositivos anteriores



multiplexadores/desmultiplexadores (233,234) para separar a informação de endereços e de dados na linha omnibus (248) e para transferir dados entre as metades superior e inferior da linha omnibus de dados (223) durante certos ciclos da linha omnibus DMAC. A linha OWN é uma saída que indica que o DMAC (209) está a controlar a linha omnibus. É usado como sinal de activação para ligar excitadores exteriores de endereços e memórias tampão de sinais de controlo.

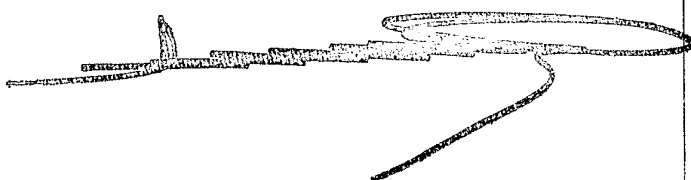
A linha BUS REQUEST (pedido da linha omnibus) (BS) (269) é uma saída identificada pelo DMAC para pedir o controlo da linha omnibus local (243,247).

A linha BUS GRANT BG (Atribuição de linha omnibus) (268) é uma entrada detectada por um árbitro externo de linhas omnibus (16) para informar o DMAC (209) que pode assumir o controlo da linha omnibus logo que esteja completado o ciclo corrente da linha omnibus.

Os dois sinais de controlo das interrupções IRQ e IACK nas linhas (258a) e (258b) formam uma sequência de transmissão com passagem de testemunho por pedido/confirmação das interrupções com PE (62) através da lógica de interrupções (212). INTERRUPT REQUEST (IRQ) na linha (258a) é uma saída identificada pelo DMAC (209) para pedir um serviço de PE (62). INTERRUPT ACKNOWLEDGE (IACK) na linha (258b) é identificada por PE (62) através da lógica (216) para confirmar que recebeu uma interrupção do DMAC (209). Em resposta à identificação de IACK, o DLAC (209) colocará um vector em D0-D7 da linha omnibus (223) que será usado pela PE (62) para buscar o endereço da rotina apropriada para processar a interrupção DMAC.

As linhas de controlo de dispositivos realizam a interface entre o DMAC (209) e dispositivos acoplados aos quatro canais DMAC. Quatro conjuntos de três linhas são dedicados a um canal DMAC único e ao seu periférico associado; as restantes linhas são sinais globais compartilhados por todos os canais.

As entradas REQUEST (REQ0 a REQ3) nas linhas (263a a d) são reclamadas pela lógica (253) para pedir uma transferência de operandos entre a memória principal (162)



e a memória (210).

As saídas ACKNOWLEDGE (ACK0 a ACK3) nas linhas 264 a-d são reclamadas pelo DMAC (209) para sinalizar que um operário está a ser transferido em resposta a um pedido de transferência anterior.

As linhas de controlo de periféricos, PERIPHERAL CONTROL LINES (PCLO a PCL3) (257 a-d, inclusive, são linhas bidireccionais entre a lógica da interface (253) e o DAMAC (209), que são ajustadas para funcionar como entradas de prontidão, aborto, regarga, estado, interrupção ou impulsão de relógio de activação, ou como saídas de impulsos de arranque.

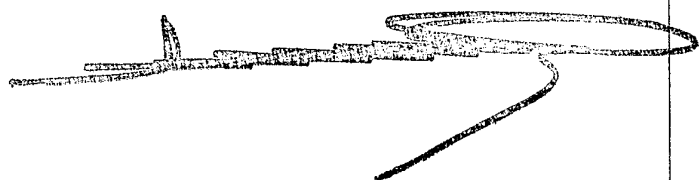
DATA TRANSFER COMPLETE (DTC), linha (267) é uma saída reclamada pelo DMAC (209) durante qualquer ciclo da linha omnibus do DMAC para indicar que os dados foram transferidos com êxito.

DONE. Este sinal bidireccional é reclamado pelo DMAC (209) ou um dispositivo periférico durante o ciclo da linha omnibus DMA para indicar que os dados que estão a ser transferidos são o último item de um bloco. O DMAC reclamará este sinal durante um ciclo da linha omnibus quando o registor da contagem de transferências da memória for decrementado até zero.

5. Unidade de controlo de linhas omnibus (156) - Descrição pormenorizada (Fig. 19 A-C e 20)

a) Registadores de interface para transferência de dados a alta velocidade

A lógica de interface da BCU (205) (fig. 16) foi dividida em várias unidades funcionais, para facilidade de ilustração e descrição nas fig. 19 A-C. Assim, a lógica (205) inclui um certo número de registadores de interface interpostos entre a linha omnibus de dados local (223) e os canais 0 e 1 do adaptador para aumentar a velocidade e a eficácia das transferências de dados entre o adaptador (154) e a BCU (156). A lógica dos equipamentos físicos (253) da interface (205), juntamente



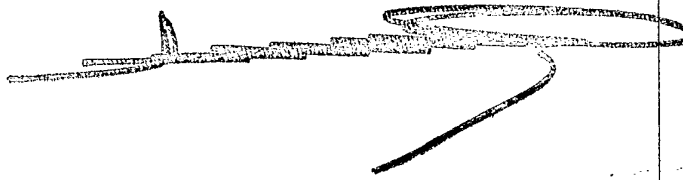
com o DMAC (209), a lógica de descodificação de endereços e arbitragem (216) e a lógica dos impulsos de sincronização de endereços (215) controlam as operações da BCU (156).

Os registradores de interface incluem um registrador do estado de leitura do canal 0 (229) e um registrador do estado de escrita do canal (1) (230), acoplados às linhas omnibus de comando/estado dos canais 0 e 1 (249,252), para reter o estado das transferências de dados entre o adaptador (154) e a BCU (156).

Os registradores de comando (214,215) dos canais 0 e 1 retêm o endereço S/370 para transferência para o adaptador (154) durante as transferências de dados I/O S/370. O registrador (227) também retém as palavras de dados I/O seguintes (até 4 octetos) das transferências de dados (até 64 octetos por transferência de endereços) para o adaptador (154), depois de cada transferência de endereço.

A memória tampão de leitura do canal 0 recebe dados I/O transferidos do adaptador (154) durante operações de leitura da caixa do correio DCU e de escrita I/O S/370.

Os contadores (220,222) "BSM WRITE/READ SELECT UP" dos canais 0,1 e os contadores (221,224) "BSM WRITE/READ BOUNDARY" dos canais 0 e 1 retêm as contagens de octetos para a transferência de dados da BCU (156) para o adaptador (154). Os dois contadores são necessários para cada um dos canais para impedir o cruzamento dos limites dos endereços de 64 bits S/370 pelas transferências de dados. Como mais adiante será descrito, os contadores (220,222) armazenam inicialmente a contagem total de octetos a transferir para uma operação de I/O (até 4 KB) e são usados para transferir valores de contagem para os registradores (214) e (225) para formar parcialmente um endereço inicial S/370 apenas para a transferência do último bloco (64 octetos), isto é, a última operação de transferência de comandos/dados. Os contadores de limites (221,224) são usados para apresentar (em parte) um endereço inicial S/370 sempre que for detectado um cruzamento dos limites pela BCU (156) para qualquer operação individual de transferência de comandos/dados ou quando o valor da contagem de octetos for superior a 64



KB.

Os contadores (220), (221), (222) e (224) são decrementados de maneira apropriada depois de cada transferência de dados, através dos canais 0 ou 1.

Um contador da fila de espera (254) (Q COUNT) proporciona uma função análoga para as transferências de mensagens (até 16 octetos) para o adaptador da memória S/370 (154).

Os endereços para seleccionar os registadores de interface anteriores estão no espaço de endereços da memória (210), fig. 23C, e são seleccionados descodificando o endereço na linha omnibus (247) de uma maneira bem conhecida.

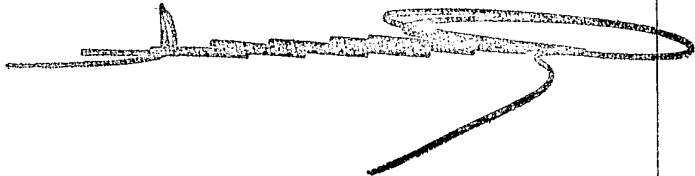
Um sinal na PU da BCU na linha (256a) proveniente do adaptador (154) para a lógica (253) notifica a BCU (156) de que está pronto um pedido de leitura da caixa de correio S/370. Este sinal não é repostado por um sinal de confirmação BCU PU na linha (256b) até a informação da caixa de correio ter sido armazenada na memória local (210).

As linhas TAG UP e TAG DOWN (262 a-d) são usadas para a sincronização de dados entre a BCU 156 e o adaptador (154) através dos canais 0 e 1 do adaptador.

Proporcionam-se sinais de transmissão com passagem de testemunho entre a lógica BCU (253) e o DMAC (209). A lógica BCU faz pedidos de serviços pelas linhas (263 a-d), um por cada canal DMAC. O DMAC responde com sinais de confirmação pelas linhas (264 a-d). Outras linhas tais como a de selecção (270), de confirmação da transferência de dados (265), linhas de controlo de periféricos (257 a-d), transferência de dados completada (267) foram descritas atrás, com referência ao DMAC (209).

b) Lógica de desacoplamento da BCU e de interrupção (215, 216) (Fig. 20,21)

Foi mais atrás referido que há duas características que são críticas para obter o acoplamento estreito dos processadores S/370 e S/88 de modo a proporcionar ao sis



tema S/370 as características de sistema único S/88, tais como o funcionamento tolerante das avarias e o ambiente com imagem de um sistema único. Designam-se aqui essas características por "desacoplamento" do sistema S/88 do seu equipamento físico associado e um mecanismo "de interrupção único". As duas características funcionam de uma maneira que é transparente para o sistema operativo S/88. O desacoplamento e a lógica de interrupção (215,216) são proporcionados na BCU (156).

A lógica de "desacoplamento" descodifica o endereço virtual aplicado à linha omnibus de endereços do processador S/88 (161A) durante cada ciclo de execução da instrução. Se forem detectados o bloco de endereços virtuais S/88 pré-seleccionados atribuídos à BCU (156) ou a sua memória (210), o sinal de impulsos de sincronização de endereços (AS) proveniente do processador S/88 (62) é encaminhado para a BCU (156) em vez de para o equipamento físico S/88. Esta acção impede que o sistema operativo e os equipamentos físicos S/88 tomem conhecimento de que teve lugar um ciclo de máquina, isto é, a acção é transparente para o S/88.

Porém, o processador S/88 (62) é acoplado para controlar a BCU (156) durante este ciclo de máquina, sendo o sinal AS e o endereço pré-seleccionado usados para seleccionar e controlar vários componentes na BCU (156) para efectuar uma função, relacionada com as operações I/O S/370.

Um código de aplicação especial (EXEC370) executado no processador S/88 inicia a comunicação com o processador S/370 (85) colocando estes endereços virtuais pré-seleccionados na linha omnibus S/88 (161A) para orientar a BCU (156) para efectuar operações para realizar a referida comunicação.

O DMAC (209) e outra lógica na BCU (156) apresentam interrupções ao S/88 num nível especificado (6), chamado esta aplicação especial para entrar em acção como for requerido. A apresentação de cada interrupção é transparente para o sistema operativo S/88.

Mais adiante será feita uma breve descrição do tipo de funções efectuadas por algumas poucas rotinas do dispositivo de tratamento de interrupções em resposta a es-



tas interrupções, com referência a um exemplo de uma vista geral dos meios de microprogramas "firmware" das operações I/O S/370.

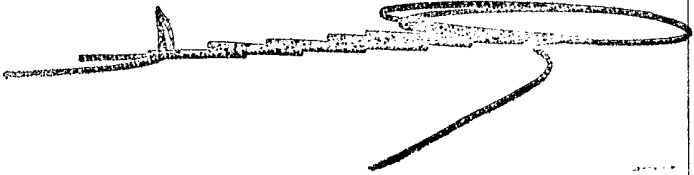
As modificações do mecanismo e do sistema operativo S/88 para tratar interrupções S/370 para S/88 através de controladores DMAC tais como (209), quer na base de uma unidade partilhada, quer num módulo com várias unidades partilhadas, vão agora ser descritas.

Recordar-se-á que uma unidade partilhada é uma sanduiche ligada de um cartão de processador S/88 dual modificado com um cartão de processador S/370 dual contendo memórias locais, DMAC e lógica de cliente duais. Os elementos iguais deste cartão dual ensanduichado operam em paralelo, em perfeito sincronismo (por arranque e paragem) por razões de detecção de avarias.

Esta sanduiche total tem normalmente uma sanduiche parceira idêntica, funcionando as parceiras sincronizadas por arranque e paragem, aparecendo assim como uma entidade tolerante das avarias. Para a discussão seguinte é suficiente considerar este equipamento físico duplicado como uma unidade operacional única, como se representa na fig. 21.

Numa forma de realização preferida, podem residir num invólucro único do módulo até oito destas unidades operacionais (295) a (295-8), compartilhando a memória principal, as facilidades I/O e as alimentações de energia, sob o controlo de uma cópia única do sistema operativo S/88. A unidade (295) (e cada uma das outras unidades (295-2) e (295-8)) correspondente a um par de cartões parceiros, tais como os cartões (21) e (23) da fig. 7. É importante notar que, nesta configuração de CPU múltiplas, as unidades do processador S/88 (62) e (62-8) funcionam com multiprocessadores que repartem a carga de trabalho S/88, mas as unidades S/370 (85) a (85-8) funcionam separada e independentemente e não intercomunicam. Cada unidade S/370 funciona sob o controlo do seu próprio sistema operativo e não tem "conhecimento" de qualquer outra CPU no ambiente (nem S/370 nem S/88).

Devido ao ambiente de multiprocessamen-

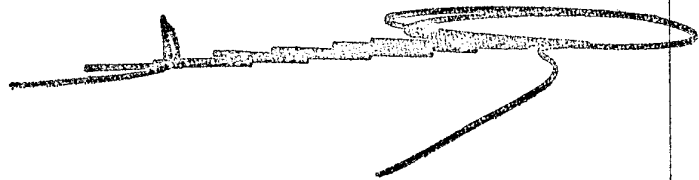


to e à arquitectura S/88, o tratamento de interrupções no sistema S/88 normal é repartido entre as unidades CPU (62) a (62-8). Numa visão simplificada, cada interrupção (proveniente das I/O, dos distribuidores de tempos, transferências de programas, etc.) é apresentada na linha omnibus comum (30) a todas as unidades do processador S/88 em paralelo; uma das unidades aceita a responsabilidade de satisfazer o serviço e faz com que todas as outras unidades o ignorem. Independentemente de qual a unidade CPU que aceita o serviço, há uma tabela de vectores única, um ponto de entrada único (por vector) no sistema operativo para o código processador, sendo a disposição da interrupção decidida e tratada pelo sistema operativo (único).

Numa configuração S/370 múltipla, todas as interrupções S/88 normais operam como atrás se descreveu; nenhum código de tratamento de interrupções S/88 é modificado. Alterações pouco importantes dos meios físicos para permitir a apresentação das interrupções do DMAC 209 ao 209-8 são inteiramente transparentes ao mecanismo de interrupções e aos meios de programação S/88.

Um requisito é que uma interrupção DMAC tem de ser tratada apenas pelo processador S/88 (62) a que está ligado esse DMAC, a BCU e o S/370, de modo que as unidades múltiplas S/370 (85) a (85-8) não podem interferir mutuamente. Para isso, a linha DMAC IRQ (258a) está cablada directamente para o processador S/88 (62) ao qual está ligado o DMAC 209 e não aparece na linha omnibus S/88 comum, como sucede para todas as linhas de pedido de interrupção S/88. Durante as faixas de tempo usurpadas do S/88 para suporte do S/370, um processador S/88 (65) dado é dedicado ao S/370 a que está directamente ligado.

Reservam-se oito localizações de vectores de utilizador dentro da tabela de vectores S/88 para utilizar pelos DMAC, sendo estes vectores endereços codificados rigidamente de meios de tratamento de interrupções DMAC que são adicionadas ao sistema operativo S/88. Estes oito meios de tratamento das interrupções são usados por todos os processadores S/88 para processar interrupções apresentadas por todos os DMAC para os processadores S/370 associados.

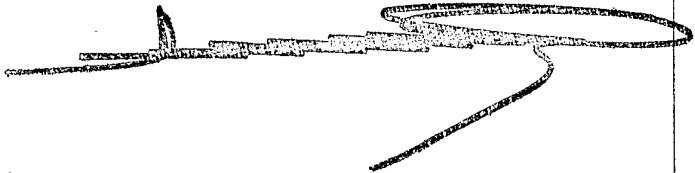


Cada DMAC, tal como (209), tem um único sinal de saída de pedido de interrupção (IRQ) e oito registadores internos de vectores (dois por canal, cada um para operações normais e erros detectados pelo DMAC). No instante da inicialização (descrita mais adiante), estes valores dos registadores de vectores DMAC são programados para corresponder às oito localizações da tabela de vectores principais reservados atrás referidos. Assim, um DMAC pode pedir uma das oito rotinas do meio de tratamento quando apresenta IRQ. Estes meios de tratamento acedem ao DMAC, aos meios físicos BCU, às filas de espera, às listas encadeadas e a todos os parâmetros de controlo apresentando endereços virtuais que se situam dentro da gama de endereços da memória local "escondida" (210). A concepção dos equipamentos físicos garante que cada processador S/88, tal como (62), pode aceder à sua própria memória tal como (210) e não a outras, embora uma "janela" de desacoplamento de endereços virtuais comum seja compartilhada entre as unidades S/370 múltiplas. Isto é, o espaço de endereços virtuais S/88 007EXXXX é usado por todos os multiprocessadores S/88-S/370 num módulo embora cada unidade emparelhada, tal como (21,23), tenha a sua memória física S/88 dedicada, como se mostra na fig. 10.

Nas configurações múltiplas S/370, todos os DMAC (209) a (209-8) são programados identicamente, no que respeita a estes oito registadores de vectores, e todos compartilham oito vectores reservados na tabela principal de vectores, bem como das rotinas dos meios de tratamento. A diferenciação, bem como o desacoplamento, verificam-se em cada acesso à memória, tal como (210).

A apresentação com cablagem rígida do DMAC IRQ ao seu próprio processador S/88, juntamente com o desacoplamento, garante a separação e a integridade das unidades do processador S/370 e a não interferência com o funcionamento normal do S/88. Excepto no que se refere ao tempo "perdido" da CPU S/88, o atendimento destas interrupções é transparente para o sistema operativo S/88.

A concepção completa das interrupções realiza assim o atendimento intermitente "dedicado por pedido"



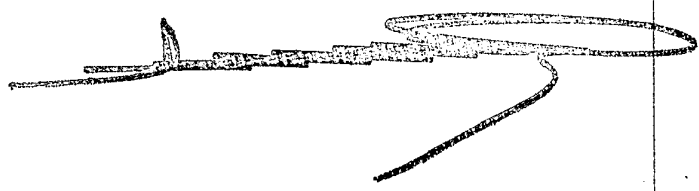
das interrupções DMAC S/370, com isolamento e protecção das unidades múltiplas S/370, usurpando facilidades individuais do processador de um ambiente de um sistema de multiprocessamento que utiliza uma filosofia diferente de atendimento das interrupções, sem substancialmente qualquer impacto no funcionamento do sistema de multiprocessamento e sem alterações significativas no sistema operativo do multiprocessamento.

Para um funcionamento mais pormenorizado de cada mecanismo de interrupção DMAC, faz-se referência às fig. 19A e 20. Quando um dispositivo periférico, tal como o DMAC (209), que possui vectores de selecção, apresenta um pedido de interrupção (IRQ) ao processador S/88 (62), é activada pelo dispositivo uma única linha IRQ (258a). Esta linha IRQ está cablada para um circuito de codificação (293), de uma maneira especificada pela arquitectura do processador S/88, de modo a apresentar um pedido de interrupção codificado ao processador S/88 (62) através dos pernos de entrada IPL0-IPL2, nível de prioridade específico 6.

O processador (62) decide efectivamente quando pode atender a interrupção, usando bits de máscara de prioridade mantidos no registador interno de estados. Quando pronto, o processador (62) inicia um ciclo especial de "confirmação de interrupção" (IACK).

No ciclo IACK, que é controlado internamente pelo processador (62), é apresentada uma única configuração de endereços na linha omnibus de endereços (161A) para identificar o tipo de ciclo e o nível de prioridade que estão a ser atendidos. Isto é também efectivamente um pedido de um número de vector a partir do dispositivo de interrupção. Todos os dispositivos que fazem pedidos comparam o nível de prioridade que está a ser atendido com o seu próprio, encaminhando o dispositivo cuja prioridade condiz com um número de vector de um octeto para a linha omnibus de dados (161D) para o processador (62) para ler.

Uma vez obtido o número de vector, o processador (62) conserva o estado interno básico numa pilha supervisora e gera depois o endereço do vector de excepção a ser



usado. Isso faz-se multiplicando internamente o número do vector do dispositivo por quatro e somando este resultado ao conteúdo do registador interno do vector base, dando o endereço da memória do vector de excepção. Este vector é o valor do novo contador de programa para o código do dispositivo de tratamento da interrupção.

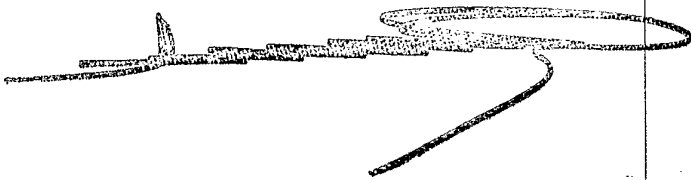
A primeira instrução é buscada utilizando este novo valor do contador do programa, retomando-se a descodificação e a execução normais da instrução, no estado de supervisor, com o registador do estado do processador (62) ajustado para o nível de prioridade agora corrente.

As fases anteriores, desde o início do ciclo IACK, passando pela busca da primeira instrução do dispositivo de tratamento da interrupção, são efectuadas por uma combinação de operações de meios físicos e internas no processador (62), não exigindo a execução de instruções do programa. O efeito líquido é a preempção transparente do programa que corria anteriormente (de prioridade inferior) para executar o tratamento da interrupção de prioridade mais elevada.

As interrupções DMAC (209) na forma de realização preferida são cabladas para o nível de prioridade 6, adaptando-se inteiramente à arquitectura do processador (62). O DMAC (209) tem oito números de vectores programados internamente, sendo usadas oito rotinas de tratamento separadas.

A lógica de descodificação e arbitragem (216) (fig. 19A) e a lógica de controlo AS (215) controlam esta função de interrupção durante o ciclo IACK além de proporcionar a função de desacoplamento do processador S/88 (62).

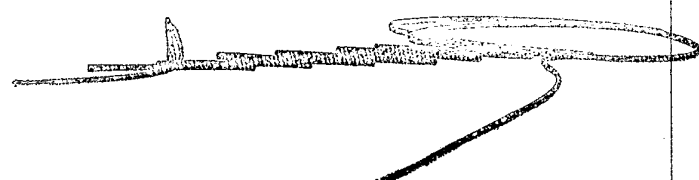
Vão agora descrever-se estas duas funções dos meios físicos detalhadamente, com referência à fig. 20, que representa pormenores da lógica (215) e da lógica (216) da fig. 19A. A linha de impulsos de sincronização de endereços (270) proveniente do PE (62) é acoplada a uma entrada da lógica de controlo (215). A lógica (216) tem dois circuitos de descodificação (280, 281). A saída (282) do circuito (280) está acoplada à lógica (215); a saída (283) do circuito (281) está também acoplada à lógica (215) através da porta E (291) e da linha (287).



Normalmente, durante a execução das instruções, os circuitos de descodificação (280) e (281) permitem que o sinal de impulsos de sincronização de endereços (AS) na linha (270) passe através da lógica (215) para a linha (270a) que é a linha de impulsos de sincronização de endereços normal para os meios físicos S/88 associados com PE (62).

Porém, se uma instrução executada pelo processador S/88 (62) aplicar um endereço virtual na linha omnibus de endereços (161A) com os quatro dígitos hexadecimais de ordem superior iguais a "007E" (implicando o desacoplamento do PE (62) do seu equipamento físico S/88 e o acoplamento do PE (62) à BCU (156) para uma função relacionada com uma operação I/O S/370), a lógica de descodificação (280) coloca um sinal na linha (282) para bloquear o sinal AS na linha (270a) e envia AS para a BCU (156) através da linha (270b). A lógica de descodificação (280) pode também ser concebida para detectar um código de função apropriado nas linhas FCO-2; porém, isto é simplesmente uma escolha de projecto. As fig. 22, 23 e 24 mostram o atraso entre os sinais de endereços na linha omnibus (161A) e os impulsos de sincronização de endereços na linha (270), que permite o bloqueio do sinal AS na linha (270a) antes do instante em que se faz subir o sinal AS. Compreender-se-á que podem usar-se meios diferentes de um grupo especial de endereços virtuais S/88 aplicados à linha omnibus de endereços para descodificar uma condição que indica o desacoplamento do PE (62) do seu equipamento físico S/88 associado e o acoplamento de PE (62) à BCU (156).

O sinal de bloqueio na linha (282) é aplicado a um circuito OU (284) para produzir um sinal de pedido da linha omnibus local do PE(62) na linha (190) para a lógica de arbitragem (285). A lógica (285) satisfará o pedido para PE (62) apenas se o DMAC (209) não tiver já feito um pedido na linha (269). A linha (191) de aceitação do pedido da linha omnibus de PE (62) é activada se não houver qualquer pedido do DMAC. O sinal de aceitação do pedido da linha omnibus do PE (62) na linha (191) eleva o nível nas linhas ENABLE (286 a,b, fig. 19A) através da lógica (253) para acoplar as linhas omnibus PE (62)



(161 A,D) às linhas omnibus locais (247,223) através dos excitadores (217) e dos excitadores/receptores (218), em preparação para uma operação PE (62) com BCU (156). Os dados e os comandos podem ser transferidos entre o PE (62) e elementos da BCU enquanto que as linhas omnibus do processador (161 A,D) são acopladas às linhas omnibus locais (247,223) sob o controle da instrução que está a ser executada pelo PE (62). O programa de aplicação EXEC370 e os microprogramas registadores em ROM ("firmware") contêm tais instruções.

Se existir um pedido DMAC na linha (269), a lógica (285) dá prioridade ao DMAC (209) sobre o pedido PE (62) na linha (190); o sinal de aceitação do pedido de linhas omnibus do DMAC na linha (268) é devolvido ao DMAC (209) e a linha omnibus local (247,223) é ligada ou entre a memória local (210) e os canais 0, 1 do adaptador, através dos registadores da interface de grande velocidade, ou entre o DMAC (209) e a memória local (210), em preparação para uma operação DMAC com a BCU (156).

Pode portanto ver-se que a lógica (215, 216) desacopla o processador S/88 (62) dos meios físicos associados (por exemplo (175), (176), (177) e acopla-o à BCU (256), quando se descodificar um endereço 007EXXXX pela lógica (280). Este desacoplamento é transparente ao sistema operativo S/88.

Analogamente, a lógica (281) de descodificação (e os meios físicos associados) bloqueia os impulsos de sincronização de endereços AS provenientes da linha (270a) e inicia um pedido da linha omnibus local para a lógica de arbitragem (285), durante uma sequência de interrupção DMAC (209) para o PE (62).

Mais especificamente, quando o DMAC (209) aplicar um sinal de interrupção na linha (258a), ele é aplicado ao PE (62) através dos circuitos OU (292a) e (292), da entrada Nível 6 da lógica de prioridade (293) de interrupções S/88 e das linhas IPLO-2. O PE (62) responde com um ciclo de confirmação de interrupção. Bists lógicos pré-determinados (que incluem o valor do nível de interrupção) são colocados na saída FCO-2 e na linha omnibus de endereços (161A) (bits A1-3, A16-19),

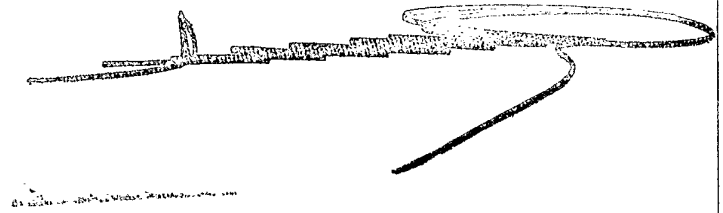


sendo esses bits decodificados pela lógica (281), para produzir uma saída na linha (283). Esta saída e o sinal de interrupção na linha (258c) fazem com que a porta E (291) aplique um sinal na linha (287) fazendo com que a lógica (215) aplique AS à lógica (253) da BCU através da linha (270b).

O sinal na linha (287) bloqueia AS proveniente da linha (270a) e coloca um pedido de linha omnibus PE (62) na linha (90) através do circuito OU (284) para a lógica de arbitragem (285). Devido ao facto de os impulsos de sincronização de endereços (AS) estarem bloqueados para não irem para os meios físicos S/88, esta interrupção é transparente para o sistema operativo S/88.

Quando se receberem os bits especiais IACK na linha omnibus (161A) e FCO-2 como atrás se descreveu, a lógica de decodificação (281) produz um sinal de saída na linha (283) para bloquear um sinal de impulsos de sincronização endereços na linha (270a) e para colocar um pedido PE (62) na lógica de arbitragem (285) através do circuito OU (284) e na linha (190). Se não houver qualquer pedido DMAC na linha (269), activa-se o nível do sinal de aceitação da linha omnibus PE (62) na linha (191) para a porta E (294-1). A porta E (294) produz um sinal IACK na linha (258) para o DMAC (209). Este sinal alerta o DMAC (209) para apresentar o seu vector de interrupção. O DMAC coloca então o vector na linha omnibus local e eleva o nível de "DTACK" na linha (265) para a lógica (253). A lógica (253), em resposta ao sinal AS na linha (270b), eleva o nível dos sinais ENABLE nas linhas (286a, 286b) para acoplar linhas omnibus do processador (161A e D) às linhas omnibus locais (248) e (223), através dos circuitos (217, 218), para ler o vector apropriado proveniente da DMAC (209) para o PE (62). O DMAC (209) apresenta vectores de interrupção do octeto menos significativo da sua linha omnibus de dados (248) (fig. 19A) para a linha omnibus de dados do processador S/88 (161D), bits 23-16, através do activador receptor (XCVR) (234) e os bits 23-16 da linha omnibus de dados local (223).

O número do vector emitido pelo DMAC (209) é usado pelo processador S/88) para saltar para um dos ou



tros dispositivos de tratamento de interrupções no microcódigo de interface S/88 ETIO.

DTACK na linha (265) e na lógica (253) activa DSACK 0,1 nas linhas (266 a,b) para terminar o ciclo PE (62), através de dois circuitos OU (288). As linhas (266 a,b) são combinadas num circuito OU com linhas DSACK S/88 normais (266 c,d), para formar as entradas DSACK definitivas (266 e,f) para o PE (62).

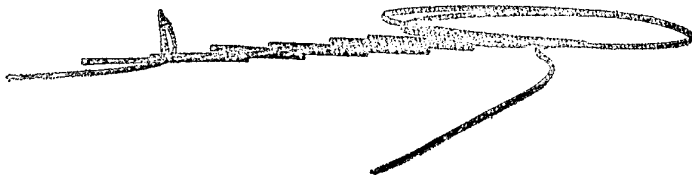
Os pedidos de interrupção aplicados ao circuito OU (292a) através das linhas (562) e (563) provenientes da Facilidade de Serviço Integrada (fig. 49) dão origem a uma sequência de operações semelhante às descritas atrás com referência a um pedido de interrupção DMAC. Dois circuitos E (294-2) e (294-3) (fig. 20) excitam sinais IACK nas linhas (258 d,e) para iniciar a transferência dos números de vectores apropriados da BCU (156) para a unidade de processamento S/88 (62) através da lógica (654) e (565) da fig. 49 e da linha omnibus de dados local (223).

Compreender-se-á que o pedido de interrupção de nível 6 S/88 pode receber uma prioridade superior à dos pedidos de interrupção DMAC e BCU (quando forem concorrentes), por uma alteração pouco importante da lógica. No entanto, correntemente, o tempo para reconhecer faltas de energia como fontes de interrupção secundárias é mais que adequado.

c) Atribuição de posições de endereços na memória na BCU

A memória local (210) (fig. 41C) tem uma dimensão fixa e tem as suas posições definidas no espaço de endereços virtuais S/88 do PE (62). A memória local (210) está dividida em três domínios de endereços para diferenciar três finalidades:

1. Leitura/escrita S/88 PE (62) directamente de/para memórias tampão de dados locais e estruturas de controlo incluindo listas de elos de ligação;
2. Comandos de leitura/escrita S/88 PE (62), estados de leitura para a/da BCU (156); os comandos são descodificados a



partir de endereços específicos; e

3. Registradores DMAC de leitura/escrita S/88 PE (62) (quer para as operações de inicialização, quer para as operações normais); os números dos registradores são descodificados a partir de endereços específicos.

O espaço de endereços da memória local inclui:

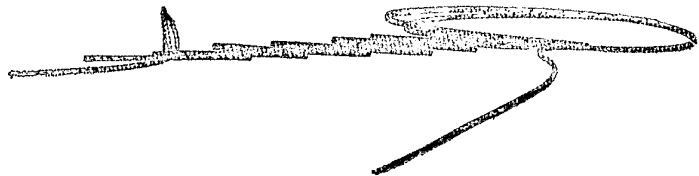
1. MEMÓRIAS TAMPÃO DE DADOS E ESTRUTURAS DE CONTROLO (64 K octetos menos 512; inclui listas de elos de ligação na memória física (210))
2. ÁREA DE COMANDOS BCU (comando de 256 octetos, descodificados a partir de endereço específico) e
3. ÁREA DE ACESSO DMAC (número de registador de 256 octetos descodificada a partir de endereço específico)

A unidade (216) de descodificação de endereços locais e de arbitragem das linhas omnibus detecta todos os endereços dentro do espaço desta memória local. O DMAC (209) pode, ao mesmo tempo, estar a apresentar um endereço no interior da área 1 atrás referida. O DMAC não pode endereçar as áreas 2 ou 3 atrás referidas; isso é garantido por microcódigo de inicialização.

A BCU 156 vigia todos os endereços na linha omnibus local e reorienta, através de indicadores de controlo, operações com endereços dentro dos domínios 2 e 3 para a unidade própria (BCU ou DMAC) em vez de para a memória local (210). Assim a área de endereços da memória local (210) representada pelos domínios 2 e 3 atrás referidos, enquanto presentes, nunca é aqui usada para memória.

Na forma de realização preferida, um quarto tipo de operação é também tratado pela unidade (215) de descodificação de endereços locais e de arbitragem das linhas omnibus:

O processador S/88 confirma interrupções DMAC (209) para PE (62) S/88 e completa todas as interrupções de acordo com a arquitectura MC 68020, como atrás se descreveu.



Esta operação especial é detectada por bits de endereço e de código de função que a PE (62) S/88 apresenta, com a diferença de que a descodificação (especial architectada) não é um endereço no domínio da memória local (210).

A unidade (216) de arbitragem das linhas omnibus locais tem portanto um descodificador especial para este caso, e auxilia a lógica a indicar ao DMAC para apresentar o seu vector de interrupção pré-programado. A operação é aliás semelhante à leitura de um registador DMAC pelo processador S/88.

A linha omnibus de endereços (247) é seleccionada pela PE (62) quando a descodificação dos dígitos de ordem superior der o valor hexadecimal (H) 007E.

Os quatro dígitos hexadecimais restantes proporcionam o domínio de endereços da memória local de 64 KB, que são atribuídos da seguinte maneira:

Dispositivo I/O	Descodificação do endereço
Seleccção (ou comando) do registador DMAC	007E0000-007E00FF (área 3 anterior)
Reposição BCU	007E0100 (área 2 anterior)
BSM Wr Sel Up	007E0104 (área 2 anterior)
BSM Rd Sel Up	007E0108 (área 2 anterior)
Ler estado BCU	007E010C (área 2 anterior)
Seleccção da memória local	007E0200-007EFFFF (área 1 anterior)

Os dados seguintes são colocados na linha omnibus de dados local (223) pelo processador S/88 (62) para um registador de contagem de transferências para a memória DMAC seleccionado, e para usar a BCU 156 num comando subsequente BSM Read/Write Select:

31 23 15 7 0
0000 Oqbb bbbb bbbb rspp kkkk CLXX XXXX

Os bits 31-16 (0000 Oqbb bbbb bbbb) da contagem de transferências de octetos são ajustados no contador de transferência de memória DMAC:

- 26 = Bit de ordem superior de contagem de octetos (= 1 para a contagem máxima de octetos (apenas 4 096)).
-



25-16 = Bits de ordem inferior da contagem de octetos. Bits 26-16 representam 1/4 da contagem actual de octetos (transferências de palavras duplas).

A BCU (156) captura os dados, da seguinte maneira, para um comando BSM Read/Write Select Up:

31-27 = Ignorados pela BCU

26 = Bit de ordem superior da contagem de octetos. Este bit será igual a 1, apenas quando se estiver a transferir a contagem máxima de octetos.

26-14 = A transferência dos bits da contagem de octetos (máximo 4 096) para o adaptador dos registadores (220) ou (222) exige uma contagem de 1111 1111 1111 para transferir 4 096 octetos (contagem de octetos 1). Portanto a BCU (156) decrementará os bits 26-16 de limite das palavras duplas, um vez, antes de os apresentar juntamente com os bits 15-14 do desvio de octetos (em blocos de 64 octetos) ao adaptador de linhas omnibus (154)

15-14 = Bits de ordem inferior da contagem de octetos. Estes bits representam o desvio de octetos menos 1 (para os requisitos do adaptador de linhas omnibus) a partir de um limite de palavras duplas. Estes bits não são usados pelo DMAC (209) ou pela BCU (156), visto que eles apenas transferem palavras duplas. Eles são retidos na BCU (156) até passarem para o adaptador de linhas omnibus (154), para apresentação ao BSM S/370 (162).

13-12 = Prioridade dos canais da linha omnibus do adaptador para o registador (219) ou (227).

11-08 = Chave da memória para o registador (219) ou (227)

07 = Bit do espaço Utilizador/IOA para o registador (219) ou (227)

06 = O processador S/88 activará este bit para BSM Write Select Up para indicar que é necessário um acesso adicional à memória local. Isso ocorrerá quando um endereço de memória local inicial não estiver num limite de palavra dupla. Como todos os acessos BCU têm de começar num limite de palavra dupla, o primeiro acesso

conterá o ou os octetos no endereço inicial designado, bem como o ou os octetos anteriores contidos nesse endereço de palavra dupla. O ou os octetos anteriores são descartados.

05-00 = Reservados

O seguinte será colocado na linha omnibus de dados local (223) pelo processador S/88 (62) para o registrador de contagem de transferências para a memória DMAC e pela BCU (156) para um comando Q Select Up subsequente:

0000 0000 0000 bbbb 0000 kkkk cxxx xxxx

O valor da contagem de transferências de octetos (bits 31-16) é ajustado no registrador MTC de contagem de transferência de memória pelo canal 3 DMAC.

A BCU (156) captura os dados para um comando Q Select Up subsequente da seguinte maneira:

31-20 = Ignorados pela BCU

19-16 = Contagem de octetos (máximo 64 octetos) para o registrador (220) ou (222)

15-12 = Ignorados pela BCU

11-08 = Chave da memória para o registrador (227)

07 = Bit do espaço Utilizador/IOA para o registrador (227)

06-00 = Ignorados pela BCU.

d) Operação das linhas omnibus de endereços e dados locais

Todas as operações de linhas omnibus locais são iniciadas através de pedidos de linhas omnibus provenientes do processador S/88 (62) ou do DMAC (209). As operações da linha omnibus local do processador S/88 (62) incluem:

Leitura/escrita da memória local (32 bits)

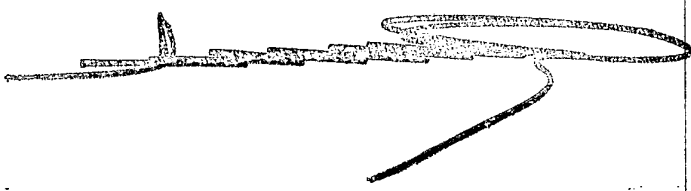
Leitura/escrita de registradores DMAC (8,16,32 bits)

Ciclo de confirmação de interrupção para DMAC (leitura do vector de interrupção de 8 bits)

Leitura do estado da BCU (leitura de 32 bits BCU)

Reposição da BCU programada

As operações das linhas omnibus locais DMAC (209) incluem:



Carga da lista de elos de ligação (16 bits)

Operações DMAC (32 bits)

Apenas proporciona endereços da memória local

Proporciona pedido de linha omnibus local

Interrupções

Proporciona o vector de interrupção normal para PE (62)
para 4 canais (8 bits)

Proporciona o vector de interrupção de erro para PE (62)
para operações DMAC ilegais e outros erros detectados no
DMAC (8 bits)

As operações da linha omnibus local BCU
(156) incluem:

Proporciona dados de leitura/escrita (32 bits) durante as
operações DMAC

Inicia o pedido de dados para DMAC (209)

Inicia o pedido de interrupções de leitura de caixa do
correio através da linha DMAC PCLO (257a).

Sempre que o processador S/88 (62) activa
a sua linha omnibus de endereços com uma descodificação da
linha omnibus local válida (007EXXXX) ou com um ciclo de confirmação
de interrupções dirigido para DMAC, a lógica BCU (156)
efectua o seguinte:

Bloqueia a linha ADDRESS STROBE (AS) pa-
ra S/88.

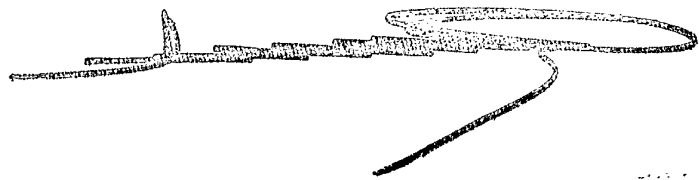
Activa um Pedido de Linha omnibus para
a lógica de contenção (216).

Se a linha omnibus local não estiver em
utilização, a linha omnibus de endereços (161A) e a linha omni-
bus de dados (161B) do processador S/88 são acopladas à linha
omnibus local (247,223), através dos receptores excitadores
(218,217). Efectua-se a operação de leitura, de escrita ou IACK.

As linhas DSACK (266 a,b) são activadas
pela lógica BCU, para fechar o ciclo:

32 bits DSACK para todos os comandos di-
rigidos à memória local e à BCU.

• 16 bits DSACK para todos os comandos di-
• rigidos ao registador DMAC.
•



16 bits DSACK para ciclos IACK.

A linha (269) de pedido da linha omnibus (BR) do DMAC proveniente do DMAC (209) é activada para uma sequência DMAC ou de carga da lista de elos de ligação. Quando isto se verifica, a BCU (156) efectua o seguinte:

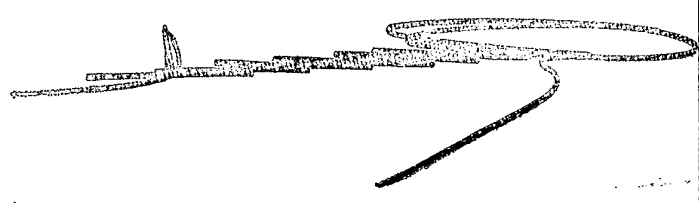
Se a linha omnibus local não estiver em utilização, o endereço DMAC (durante a carga de leitura/escrita DMAC ou da lista de elos de ligação) é encaminhado para a linha omnibus de endereços local (247). A lógica BCU (156) encaminha os dados (escrita DMAC para a memória local (210)) a partir de um registador DMAC para a linha omnibus de dados local (223). A memória local (210) encaminha os seus dados (leitura DMAC ou carga da lista de elos de ligação) para a linha omnibus local (223). É efectuada a operação de leitura/escrita.

A linha DTACK é activada pela lógica BCU (253) para o DMAC (209), para encerrar o ciclo.

e) Endereçamento do processador S/88 (62) e do DMAC (209) para/da memória (210)

As atribuições dos bits do endereço do processador S/88 (62) para a memória local (210) são as seguintes: bits de ordem inferior 0,1 (e IZ0,1 do PE (62), não representados) determinam o número e o alinhamento na linha omnibus dos octetos (1-4) a transferir. Os bits 2-15, inclusive, são os bits de endereço para o espaço de memória (210).

No modo de lista de elos de ligação, o bit A2 do endereço DMAC é usado como bit de ordem inferior do endereço (limite de palavra dupla) para a memória local (210). Como o DMAC (209) é um dispositivo orientado para palavras (16 bits) (A1 é o seu bit de ordem inferior do endereço) e como a memória local (210) acedem palavras duplas (32 bits), têm de proporcionar-se meios, nos equipamentos físicos, para permitir que o DMAC (209) leia dados para a sua lista interna de elos de ligação a partir de posições de memória local contíguas. Isso consegue-se lendo duas vezes a mesma posição de palavra dupla na memória (210), usando A2 como bit de ordem inferior do ende-



reço. Como então A1 é usado para seleccionar a palavra alta/baixa a partir da linha omnibus local. O deslocamento dos bits de endereço para a memória local (210) é feito no equipamento físico através dos bits de código de função DMAC. Qualquer código de função, excepto "7", proveniente do DMAC (209) fará com que os bits A15-A02 do endereço sejam apresentados à memória local (210). Este esquema permite que os dados da lista de elos de ligação da memória local sejam armazenados em posições contíguas na memória (210).

No modo de leitura/escrita da memória local, o bit DMAC é usado como bit de ordem inferior do endereço para a memória local (210). Os dados lidos são fornecidos à memória (210) a partir da memória tampão (226) de leitura do canal 0 da linha omnibus do adaptador. Os dados são escritos a partir da memória (210) para a memória tampão (228) de escrita do canal 1 da linha omnibus do adaptador. Como o DMAC é um dispositivo de 16 bits, pretende-se que o bit de ordem inferior do endereço represente um limite de palavra. Porém, todas as operações DMAC acedem a uma palavra dupla. Para permitir acessos a palavras duplas com um mecanismo de endereçamento de acesso a uma palavra, é necessário um deslocamento de endereço.

O deslocamento dos bits de endereço para a memória local (210) faz-se no equipamento físico através dos bits de código de função DMAC. Um código de função, de 7 provenientes do DMAC (209), fará com que os bits A14-A01 do endereço sejam apresentados à memória local (210). Para permitir a operação correcta, o DMAC é carregado com 1/4 do valor da contagem de octetos actual (1/2 da contagem actual de palavras). Para uma operação de escrita DMAC, há uma medida para permitir escritas de palavras controlando as linhas UDS e LDS (não representadas) provenientes do DMAC (209), embora todas as operações DMAC sejam normalmente acessos de palavras duplas. Os sinais UDS e LDS provocam o acesso das porções de ordem elevada (D31-D16) e de ordem baixa (D15-D0) da memória local (210),

No modo PE (62) para DMAC (209), o processador S/88 PE (62) escreverá nos registadores DMAC, em cada um dos quatro canais DMAC 0-3, para estabelecer os controlos in

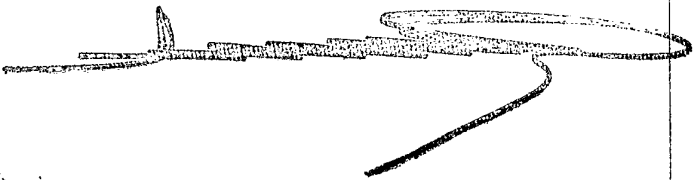


ternos para uma operação DMAC. A PE (62) tem também a capacidade de ler todos os registadores DMAC. O DMAC (209) devolve uma palavra (16 bits) DSACK numa linha omnibus (266) que tem duas linhas, DSACK 0, DSACK 1, permitindo dimensões de portas de 8, 16 ou 32 bits. Isso permite que o DMAC (209) tome tantos ciclos quantos os necessários para efectuar a carga DMAC de maneira apropriada.

As linhas SIZ0, SIZ1 (não representadas) e A0 do processador S/88 são usadas para gerar entradas UDS (Upper Data Strobe - Impulsos de sincronização de dados superiores) e LDS (Lower Data Strobe - Impulsos de sincronização de dados inferiores) (não representadas) para o DMAC (209). Isto é necessário para ter acesso a registadores com a dimensão de um octeto no DMAC (209), como se descreveu mais completamente na publicação DMAC atrás descrita. A linha LDS é gerada a partir da combinação lógica OU de NOT SIZ0 ou SIZ1 ou A0 da linha omnibus de endereços (161D). A linha UDS é gerada a partir da lógica NÃO de A0. A linha SIZ0 é usada para aceder ao octeto de ordem inferior quando se acede a uma registador com a dimensão de uma palavra (NOT ZIZ0). A linha SIZ1 é usada para aceder ao octeto de ordem inferior quando se está a aceder a uma registador com a dimensão de uma palavra, através de uma operação "três octetos restantes" do processador S/88. Isso ocorrerá apenas quando o processador S/88 estiver a executar uma operação de leitura/escrita de uma palavra dupla (32 bits) para o DMAC num limite de octeto ímpar. O bit A0 é usado para seleccionar o octeto superior ou o octeto inferior num registador de dois octetos. Os bits A0,A1 são usados para seleccionar octetos num registador DMAC de quatro octetos. Os bits A6,A7 da linha omnibus de endereços (161D) do PE (62) seleccionam um dos quatro canais DMAC.

f) Operação do contador de octetos BCU BSM RD/WR

O BCU (156) é capaz de aceitar um comando único proveniente do DMAC (209) que transferirá até 4 KB de dados através de cada linha omnibus do adaptador (250,251). Porém, cada linha omnibus pode tratar apenas blocos de 64 octetos



para uma operação de transferência de dados. Há outras restrições da linha omnibus do adaptador a que tem de obdecer o equipamento físico a fim de satisfazer os requisitos do protocolo. O que segue é uma descrição pormenorizada dos meios físicos BCU (156) que efectuam isso.

A BCU (156) contém dois contadores de palavras completas (11 bits) (220,222) e dois contadores (4 bits) (221,224) de limites, que são usados para operações de leitura BSM da linha omnibus e BSM de escrita do adaptador. Os contadores de limites (221,224) são usados para representar um endereço inicial para o adaptador de linhas omnibus, quando se detectar o atravessamento de um limite de 64 KB pela BCU (156) para qualquer operação individual de comando/transferência de dados, ou quando a contagem de octetos for maior que 64 octetos. Os conteúdos do contador de limites são apresentados ao adaptador de linhas omnibus (154) para todas as transferências excepto a do último bloco. Os conteúdos dos contadores de palavras completas são apresentados para a transferência do último bloco apenas (operação da última transferência de comando/dados).

O processador S/88 (62) coloca a contagem de octetos, a chave, e os bits de prioridade na linha omnibus local (223) (fig. 45F) para transferência para o registador (222) ou (220). O bit R (bit de contagem 1) representa limites de palavras (2 octetos) e o bit S (bit de contagem 0) representa limites de octetos. Os bits de contagem de palavras completas representam uma capacidade de transferência de 2KB-1 palavras duplas. Como todas as transferências são feitas numa base de palavras duplas, o bit 2 é o bit de decrematação de ordem inferior. Os bits R e S são retidos pela BCU e apresentados ao adaptador de linhas omnibus (154), na transferência final de 64 octetos.

Devido às restrições seguintes do adaptador de linhas omnibus e ao facto de apenas se verificarem transferências de palavras duplas na linha omnibus local (223), torna-se necessário manipular os bits de contagem de octetos e de palavras. Isso permitirá que os octetos/palavras ímpares sejam transferidos para a PE (85) S/370, e permitirá também um en

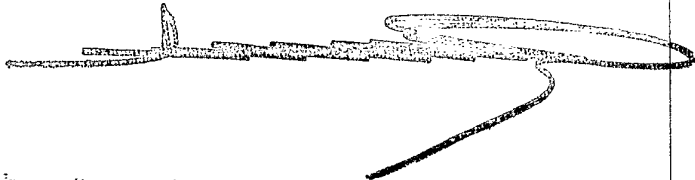


dereço inicial que não é um limite de palavra dupla. A contagem de octetos que é apresentada ao adaptador de linhas omnibus (154) não pode ser superior a 64 octetos. A contagem tem de ser representada em octetos-1. Nenhuma transferência de blocos pode cruzar um limite de 64 octetos. Quando a contagem de octetos for igual a, ou menor que 64 octetos e não houver atravessamento de um limite e o endereço inicial não estiver num limite de palavras duplas, pode ser necessário um ajustamento extra à contagem de palavras duplas.

Quando houver um atravessamento de um limite de 64 octetos, são necessárias pelo menos duas operações de transferência de comandos/dados da linha omnibus do adaptador, independentemente do valor da contagem. O processador S/88 fará o cálculo prévio da contagem de palavras duplas e dos bits R, S e I, com base num exame dos factores atrás descritos, e na contagem das transferências totais de octetos. Os bits R e S não serão apresentados ao adaptador (154) até à última operação de transferência de comandos/dados.

Quando o PE (62) S/88 coloca a contagem na linha omnibus local (223) (fig. 45F), o DMAC (209) captura os bits 31-16 e a BCU captura os bits 26-6. A BCU (156) armazena os bits 26-14 no registador (220) ou (222). Os bits 26-16 representam o campo de contagem de palavras duplas. O contador (220) ou (222) é decrementado num limite de palavra dupla (bit 2). O processador S/88 PE (62) coloca um BSM Read/Write Select Up Command na linha omnibus de endereços local (247) e o endereço inicial BSM na linha omnibus de dados local (223).

O DMAC (209) é um dispositivo de 16 bits que está ligado a uma linha omnibus de 32 bits. Está programado para transferir palavras (2 octetos) durante operações DMA em todos os canais e cada registador de endereços de memória interno MAR é incrementado de uma palavra (2 octetos) por transferência. Porém, é necessário um incremento de palavra dupla (4 octetos), visto que cada transferência é efectivamente de 32 bits. Para o conseguir, o processador S/88 PE (62) inicializa sempre o MAR para metade do endereço inicial desejado (na memória (210)). A BCU (156) compensa então isso duplicando o en



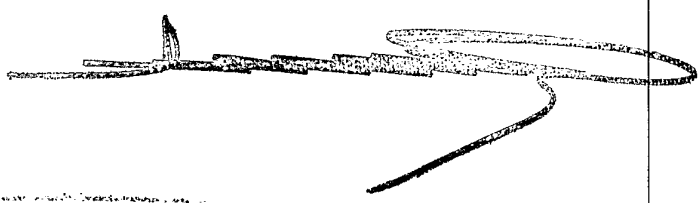
dereço do MAR antes de o apresentar à linha omnibus local (223), donde resulta a sequência de endereços correcta, tal como é vista na memória (210).

A BCU (156) efectua o seguinte:

1. O contador de limites (221) ou (224) é carregado a partir de bits 2-5, invertidos, da linha omnibus de dados local (223), ao mesmo tempo que o registador de endereços BSM (228) ou (231) é carregado.
2. Decremento do contador de palavras completas (220) ou (222) num limite de palavras duplas (bit 2); e
3. Incremento do registador de endereços BSM (228) ou (231) num limite de palavras duplas (bit 2).

Quando ficam mais de 64 octetos ou se verifica um atravessamento de um limite durante uma transferência de blocos de dados, a BCU (156) carrega a contagem de octetos de comando de leitura/escrita BSM na linha omnibus de comandos/estado (249) ou (252), a partir do contador de limites (221) ou (224) e do registador de endereços BSM (231) ou (228), bits 1,0 (invertidos). Efectua-se depois a operação de Leitura/Escrita. A BCU (156) decrementará os registadores de contagem de limites (221) ou (224) e o registador de contagem de palavras completas (220) ou (222) num limite de palavras duplas; além disso, incrementará o registador de endereços BSM (231) ou (228) num limite de palavras duplas. A BCU (156) parará quando, no registador de endereços BSM (231) ou (228), bits 5-2 = 0000, um limite de 64 octetos. Neste instante, os bits do contador de limites deve ser = 1111.

Quando restarem 64 octetos, ou menos, e não houver atravessamento de um limite durante uma transferência de blocos de dados, a BCU (156) carregará a contagem de octetos de comando de Leitura/Escrita BSM na linha omnibus do adaptador, na linha omnibus de comandos/estados (249) ou (252), a partir dos bits 5-2 do contador (220) ou (222) e dos bits R e S. A BCU (156) efectua então uma operação de Leitura/Escrita durante a qual decrementa o registador (220) ou (222) num limite de palavras duplas, incrementa o registador de endereços BSM (231) ou (228) num limite de palavras duplas e para quando todos os



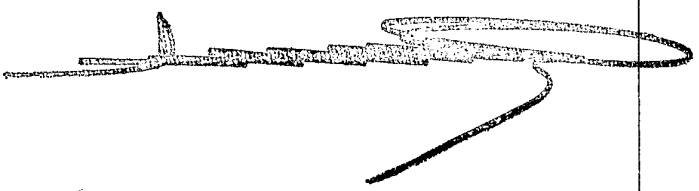
bits 12-2 do registrador (220) ou (222) forem "1". Um atravessamento de um limite é detectado comparando os bits 2-5 do registrador de contagem (220) ou (222) com o seu registrador de limites (221) ou (224). Se o valor do registrador de contagem (220) ou (222) for maior do que o do registrador de limite (221) ou (224), então foi detectado um atravessamento de um limite.

g) Sequências de transmissão com passagem de testemunho
BCU 156/Adaptador (154)

O diagrama de distribuição de tempos da fig. 5 mostra as sequências de transmissão com passagem de testemunho entre a BCU (156) e o adaptador (154) para os comandos de Leitura da Caixa de Correio e dos comandos de leitura da memória, incluindo a transferência de duas palavras de 32 bits para uma memória tampão da fila de espera de trabalho na memória local (210).

Quando um comando de leitura da caixa de correio ou de leitura da memória é produzido na linha omnibus (290) (fig. 19A), dois sinais Gate Left e Gate Right introduzem sequencialmente as partes da esquerda e da direita do comando e do endereço em registradores (214) e (219) (fig. 19B) para o adaptador (154) buscar os dados apropriados da memória S/370 (162). Activa-se o comando Tag Up na linha (262a), a que se seguem sinais periódicos de leitura de dados. Activa-se o comando Tag Down na linha (262b), até os dados buscados serem armazenados na memória tampão (259). Quando se activam os sinais periódicos Clock Left e Clock Right seguintes, introduzem-se as partes da esquerda e da direita da primeira palavra buscada na memória tampão (226), através da linha omnibus (250).

Activa-se Bus Request na linha (263a ou b) para o canal 0 ou 1 do DMAC. O DMAC arbitra o controlo da linha omnibus local através da linha (269). Quando este pedido for aceite pela lógica (216), activa-se Bus Grant na linha (268). O DMAC (209) activa o sinal de confirmação Acknowledge na linha (264a) ou (264b), o que faz com que a BCU encaminhe os dados na memória tampão (226) para a linha omnibus de dados local (223),



enquanto que o DMAC (209) coloca o endereço de memória local seleccionado na linha omnibus de endereços local (247). Então o DMAC (209) envia DTC pela linha (267) para fazer com que a lógica (253) active o sinal de selecção da memória, Store Select, na linha (210a), sendo os dados na linha omnibus (223) colocados na memória tampão apropriada na memória local (210).

Os sinais periódicos sucessivos Tag Up, Clock Left e Clock Right, DMAC Request encaminham as palavras de dados sucessivas para a memória tampão (226), sendo estas palavras transferidas para a memória tampão apropriada, na memória (210), enquanto o DMAC (209) obtém acesso às linhas omnibus locais (247) e (223), através da lógica de arbitragem (216), e emite sinais Acknowledge e DTC.

A fig. 26 mostra analogamente as sequências de transmissão com passagem de testemunho para comandos de selecção da fila de espera, Quene Select Up, e de escrita na memória, Storage Write. Quando for emitido qualquer destes comandos pela linha omnibus (290), os sinais Gate Left e Gate Right transferem o comando e o endereço (previamente armazenados nos registadores (225) e (227)) para o adaptador (154). Na linha (262a) são activados um comando Tag Up seguido de sinais de dados periódicos. Na linha 263 c ou d é activado DMAC Request. O DMAC (209) arbitra a linha omnibus local (247,223) através da linha (269) e da lógica (216). Quando o pedido for aceite através da linha (268), o DMAC (209) activa a confirmação Acknowledge na linha (264c ou d), a que se segue DTC na linha (267) para transferir a primeira palavra de dados da memória (210) para o registador (227). Os sinais periódicos Gate Left e Gate Right seguintes transferem a primeira palavra de dados do registador (227) para a memória tampão (260) do adaptador (154).

Os sinais seguintes DMAC Request na linha (263c ou d) e DMAC Acknowledge e DTC transferem as palavras de dados seguintes para o registador (227) enquanto o DMAC (209) arbitra o controlo das linhas omnibus locais (247,223). Os sinais periódicos Gate Left e Gate Right seguintes transferem cada palavra de dados do registador (227) para a memória tampão (260).

ELEMENTO PROCESSADOR S/370 PE (85)

Cada elemento de processamento, tal como PE (85), da forma de realização preferida contém as facilidades básicas para o processamento de instruções S/370 e contém as facilidades seguintes:

Fluxo de dados básico de 32 bits.

Unidade aritmética/lógica (ALU) (306) de 32 bits.

Unidade de deslocamento (307) de 32 bits

Memória local de dados de 48 registadores (de 32 bits cada um) (303) com possibilidade de endereçamento de 3 portas;

Memória tampão de instruções S/370 (309) de 8 octetos; e

Facilidades de distribuidor de tempos (distribuidor de tempos CPU, comparador, etc.) (315).

O fluxograma de dados simplificado de uma forma de realização preferida do PE (85) está representado na fig. 27, devendo entender-se que existem muitas realizações práticas do processador S/370 bem conhecidas na técnica. A forma preferida de cada elemento de processador (85) da realização preferida é um processador capaz de executar as instruções da arquitectura do sistema/370. O processador busca instruções e dados a partir de uma área de memória real (162) da memória (16), através da linha omnibus (170) do processador. Esta linha omnibus (170) bidireccional é a ligação universal entre o elemento processador PE (85) e as outras unidades do conjunto (150) de micropastilhas S/370. O PE (85) actua como director, mas tem a prioridade mais baixa no sistema. As instruções são executadas por meios físicos e por microinstruções que o processador executa quando no modo micro.

O PE (85) tem três grupos principais de funções:

- O "grupo linhas omnibus" que consiste nos registadores de emissão e de recepção (300,301), e os registadores de endereços (302) para armazenar operandos e instruções.
- O "grupo aritmética/lógica" que consiste na memória local de



dados (DLS) (303), os registadores de operandos A e B (304, 305), a unidade aritmética e lógica ALU (306) e a unidade de deslocamento (307).

- O "grupo decodificador de operações" constituído pelo registador de endereços da memória de controlo (CSAR) (308), a memória tampão de instruções S/370 (I-Buffer) (309), os registadores OP (310) e os contadores de ciclos (311) com controlo de transferência de programas e de excepções.
- O "grupo de distribuidores de tempos", com uma unidade (315) pequena, relativamente independente, constituído por um distribuidor de tempos de intervalos (315), o relógio das horas do dia, o comparador de relógios e o distribuidor de tempos da CPU.

A descrição breve seguinte esboçará a utilização destas unidades lógicas.

A memória tampão de instruções (309) põs as instruções S/370 à disposição do decodificador o mais rapidamente possível. A primeira meia palavra contendo o código operativo é fornecida através do registador de operação (310) ao decodificador (312) para iniciar a fase I S/370. A segunda e a terceira meias palavras (se as houver) são fornecidas à ALU para cálculo do endereço. A memória tampão I (309) é um registador de palavras duplas que é carregado por operações tais como IOL, LOAD PSW ou PSW permutadas através de uma operação forçada no registador (313), antes de iniciar a sequência da instrução S/370. A memória tampão I (309) é de novo preenchida palavra-a-palavra enquanto as instruções são fornecidas para o registador de operação (310) (e para a ALU (306), para cálculo de endereços) e é completamente preenchida durante cada ramificação com êxito. O decodificador de operações (OPDEC) (312) selecciona a operação a efectuar. O decodificador é alimentado a partir dos registadores de operações de microcódigos de operações (310). Os bits de modo decidem qual (ou nenhum, no caso de uma operação forçada) toma o controlo para a decodificação.

Os conteúdos da memória tampão (309) são fornecidos para o registador de operação (310) e, em paralelo, para o registador CSAR (308) para endereçar uma tabela de códi-



gos de operações na memória de controlo (171). Cada uma das entradas nesta tabela serve para dois fins: indica se existe uma rotina em microcódigo e endereça a primeira instrução dessa rotina. As rotinas de microcódigos existem para a execução de instruções mais complexas, tais como instruções com campo de comprimento variável e todas as outras que não são executadas directamente por meios físicos. Códigos especiais de funções nas instruções micro activam os meios físicos de suporte de modo que é possível controlar o fluxo de dados de 32 bits usando instruções micro de 16 bits.

Todo o processamento ocorre numa canalização de informações de três etapas:

- A primeira etapa lê a instrução para o registador OP (310).
- A segunda etapa lê os endereços e/ou os dados para os registadores A/B (304) e (305) e para o registador de emissão (300) da linha omnibus (300). O registador OP (310) é libertado para uma outra primeira etapa, fazendo passar o seu conteúdo para o descodificador OP (312), que controla a terceira etapa.
- A terceira etapa realiza a operação ALU, deslocamento ou linha omnibus, conforme o que for requerido em cada caso.

Efectuam-se nesta etapa também operações de escrita DLS.

O processamento efectivo é adicionalmente melhorado realizando o descodificador em vários grupos (não representado), um dedicado especificadamente para a ALU, um outro para o grupo de linhas omnibus e assim sucessivamente. Multiplexadores seleccionáveis por octetos (não representados) na entrada dos registadores A/B e na saída da ALU ainda melhoram a execução das operações. Assim, há instruções RR S/370, cada uma das quais ocupa uma das etapas da canalização de informações durante apenas um ciclo.

Os registadores (313) de operação força da (FOP) são usados para controlo interno. Estes registadores obtêm entradas de dispositivos de rejeição e de condições excepcionais e forçam um outro modo para o descodificador (312). São operações típicas a carga de memórias tampão de instruções I, transição para o nível dos dispositivos de rejeição e o início de rotinas de excepção.

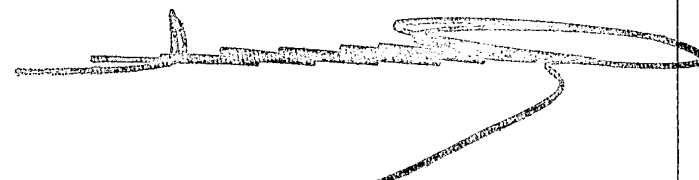


Todos os registradores de operação (310) têm um contador de ciclos (311) próprio. Os contadores de ciclos de microcódigo são compartilhados por algumas operações forçadas (FOP). As operações aritméticas e a maior parte das outras micro-instruções apenas requerem um ciclo. A maior parte das micro-instruções que executam operações das linhas omnibus do processador exigem dois ciclos.

A memória de dados local DLS (303) contém 48 registradores de palavras completas (4 octetos), que são acessíveis através de três portas, sendo uma delas uma porta de entrada e duas portas de saída. Qualquer registrador pode ser endereçado através do registrador (314) para a entrada (IN), podendo o mesmo registrador ou dois registradores diferentes ser simultaneamente endereçados para a saída. Este endereçamento triplo permite que a busca de operandos se sobreponha ao processamento. Devido a uma lógica de comparação e encaminhamento de dados (não representando) um registrador precisamente endereçado para uma operação de escrita pode também ser usado como entrada no mesmo ciclo. Isto facilita as operações de canalização das informações.

A ALU (306) é de preferência uma unidade lógica de palavras completas, capaz de executar operações E, OU, OU EXCLUSIVO e ADD, na forma verdadeira e na forma invertida, sobre dois operandos de palavras completas. A adição decimal é também admitida. A predição e a geração de paridades, bem como a propagação rápida do transporte estão também incluídas. O registrador de guarda (320) permite as operações de divisão. A lógica de estados (ALU STAT) (321) gera e armazena várias condições para decisões em ramificações do programa, avaliação do sinal, etc..

O registrador de endereços do controlo da memória (CSAR) (308) endereça micro-instruções e tabelas para a memória de controlo (171). A entrada para CSAR (308) é um endereço actualizado proveniente do modificador MOD associado (322) ou um endereço de destino de uma ramificação com resultado afirmativo, ou um endereço forçado para a consulta de uma tabela. A consulta de uma tabela é obrigatória no início de cada

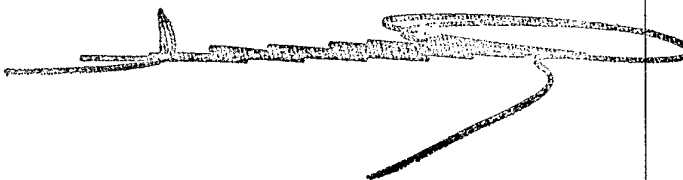


instrução S/370 e para algumas operações forçadas (FOP). O CSAR (308) obtém o padrão do código OP como um endereço para aceder à tabela de Códigos OP (fig. 29). A saída desta tabela de códigos OP define a forma de execução que pode ser a descodificação directa do registador de operações (310). Se for necessária a execução indirecta, a saída da tabela de códigos OP é reenviada para o CSAR para endereçar a micro-rotina apropriada.

O registador de endereços da memória (302) está dimensionado para endereços de 24 bits. Um modificador (323) associado MOD actualiza o endereço de acordo com a dimensão do bloco de dados buscado. As instruções são buscadas em avanço, por incrementos de uma palavra (4 octetos) enquanto a memória tampão de instruções (309) está a ser esvaziada. A entrada para o registador de endereços de memória (302) vem do registador (324) de endereços de operandos das instruções. Além disso, é posto em paralelo com o registador de endereços das instruções (324) por razões de aumento da velocidade.

O fluxo de dados da CPU permite o processamento em sobreposição de até três instruções S/370 de cada vez. As instruções S/370 são executadas, ou por meios físicos, ou interpretadas por micro-instruções. O tempo de ciclo básico da forma de realização preferida é 80 ns. O processamento das instruções faz-se em uma ou mais etapas de 80 ns. Um dispositivo de multiplicação com elevada velocidade PS (151) acelera as operações de multiplicação binárias em vírgula flutuante. As micro-instruções provenientes da memória central (171) são usadas apenas para a execução das instruções demasiado complexas e portanto muito dispendiosas quando feitas inteiramente por meios físicos. As micro-instruções, se forem necessárias, são fornecidas com a velocidade de 60 ns por instrução. O conjunto de micro-instruções está optimizado para as instruções S/370 de interpretação. As micro-instruções têm o formato de meias palavras e podem endereçar dois operandos.

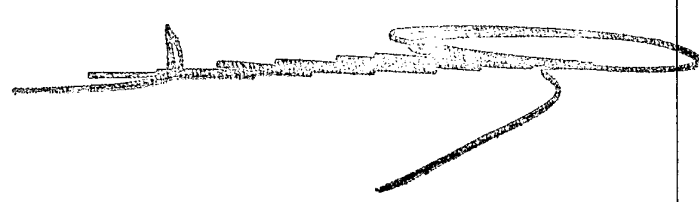
O microcódigo não contido na memória de comando (171) é retido na área objecto interna IOA (187), que é uma área reservada na memória S/370 (162) (ver fig. 28,29). Este microcódigo inclui o código sensível de menor eficácia para



excepções, instruções S/370 executadas pouco frequentemente, etc.. Estas micro-rotinas são buscadas com base em pedidos para a memória tampão de 64B (186) na parte de RAM da memória de comando (171). Sempre que o PE (85) encontrar um endereço maior que o realizado na memória de comando (171), ele indica uma operação de busca de um bloco de 64 B para o controlador da memória "cache" (153) e a interface do controlador da memória (155). As unidades (153,155) buscam o bloco de 64 B a partir da área IOA (187) e emite-o para o PE (85), que o armazena na memória tampão (186). As micro-instruções são buscadas pelo PE (85) a partir da memória tampão (186) para execução. Todo o microcódigo é carregado na memória no instante de carga do microcódigo inicial (IML). O sistema proporciona um apoio IML para facilitar a carga do microcódigo do S/88 na memória.

As instruções S/370 e os dados do utilizador são buscados a partir de uma memória "cache" (340) de elevada velocidade de 8KB (fig. 31). Os dados são lidos/escritos da/na memória "cache" (340) à base de palavras completas. O tempo necessário para ler/escrever uma palavra completa de/para a memória "cache" é de 120 ns. A memória "cache" (340) é automaticamente reenchida com blocos de 64 octetos, a partir da memória (162), à medida que surge a necessidade. O PE (85) comunica com a memória "cache" (340) por via dos comandos das linhas omnibus do processador. Os endereços virtuais proporcionados pelo PE (85) são usados para consultar os endereços das páginas pré-traduzidas correspondentes na tabela separada de consulta da lista (DLAT) (341).

A memória local de dados (303) no PE (85) inclui 16 registadores gerais, 4 registadores de vírgula flutuante e 24 registadores de trabalho. Todos os registadores podem ser endereçados individualmente através de três portas susceptíveis de ser endereçadas separadamente. Assim, a memória (303) pode fornecer dois operandos em paralelo para a ALU (306) e simultaneamente aceitar uma palavra completa da ALU (306) ou da memória "cache" (340) dentro do mesmo ciclo de 80 ns. Como não há disposição em série, como nas memórias de dados locais convencionais, as operações aritméticas e lógicas podem ser executadas de maneira sobreposta com preparação para as instruções



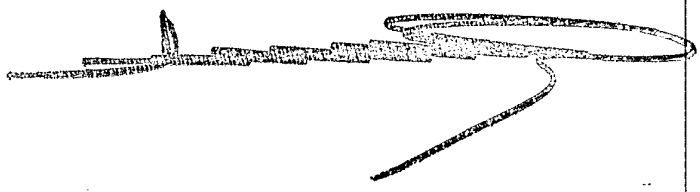
seguintes.

A CPU mantém uma memória de instruções (I-Buffer) de 8 octetos (309) para instruções S/370. Esta memória tampão é inicializada por uma instrução de ramificação S/370 com resultado positivo. O PE (85) busca uma palavra dupla de dados a partir da corrente de instruções S/370 a partir da memória "cache" (340) e carrega a mesma na memória de instruções (309). Quando se carrega a primeira palavra completa na memória de instruções (309), o PE (85) inicia de novo a execução de uma instrução. Os dados da memória tampão de instruções são buscados a partir da memória "cache" (340) simultaneamente com a execução das instruções S/370. Como o primeiro ciclo em cada execução de uma instrução é um ciclo "não "cache"", a CPU utiliza este ciclo para pré-buscar uma palavra completa a partir da memória "cache" para a introduzir na memória tampão de instruções (309).

Dispõe-se de um segundo ciclo "não-cache" com instruções S/370 que necessitam de indexação durante o cálculo do endereço efectivo, ou que são executadas por micro-rotinas. Nestes casos, a busca de instruções S/370 pode ser completamente sobreposta com a execução de instruções S/370.

Na forma de realização preferida, o conjunto de micropastilhas S/370 comunica através de um mecanismo de interrupção que exige que a micropastilha receba uma interrupção para a confirmar repondo o retentor de interrupção da micropastilha de emissão.

Sempre que o sistema (por exemplo através da BCU (156)) ajusta (activa) um ou mais bits num registador de estados (STR) (descrito mais adiante) do adaptador (154), o sistema tem também de activar uma linha de controlo NATTNREQ. Isso provoca uma excepção no elemento processador (85) quando tiver sido executada a instrução S/370 corrente, forçando assim o elemento processador (85) a "olhar" para o registador de estados. Um processador de excepções detectará então o conteúdo do registador STR, interrogará o "tipo ou tipos de interrupção" e despachará a ou as micro-rotinas do sistema apropriadas. Sempre que o elemento processador (85) activa um bit no registador STR,



o sistema tem de reagir ao mesmo em conformidade. Basicamente há dois tipos de pedidos de interrupção:

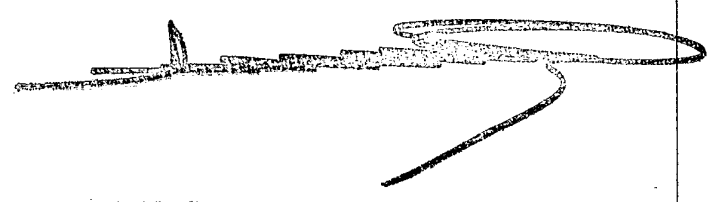
1. Pedidos do sistema (SYSREQ) são pedidos (através da BCU (156)) para o elemento processador (85) S/370. O sistema ajusta o ou os tipos de interrupção no registador STR para especificar o seu pedido. Isso provoca uma excepção no elemento processador (85) que transfere o controlo para o dispositivo que trata as excepções. Este dispositivo que trata as excepções despacha a micro-rotina apropriada, que envia um comando PROC-Bus para o adaptador (154), para repor o tipo de interrupção apropriado no registador STR, executa a função definida pelo tipo de interrupção e inicia a execução da instrução S/370 seguinte.
2. Pedidos de transferência podem ser invocados ou pelo sistema ou pelo PE (85) implicando uma transferência adicional de dados na interface do sistema. Para isso, prevêem-se dois retentores de interrupção no registador STR: um é o pedido de comunicação do processador de comunicação (PCR) e outro é o pedido de comunicação do sistema (SCR). O PCR é ajustado pelo PE (85) e repostado pelo sistema; o SCR é ajustado pelo sistema e repostado pelo PE (85).

Para as operações de transferência rápida de dados, previu-se a existência de dois registadores adicionais: o registador BR (115) (fig. 13), que é ajustado pelo PE (85) e lido pelo sistema, e o registador BS (116), que é ajustado pelo sistema e lido pelo PE (85).

O que se segue é um exemplo de um PE (85) para pedido de transferência do sistema. O PE (85) coloca dados para serem transmitidos para o sistema no registador (115) e liga o retentor PCR1. O sistema lê os dados provenientes do registador (115) e repõe o retentor PCR.

O processador pode explorar o retentor PCR para verificar se ele foi ou não repostado. O PE (85) pode transferir mais dados para o sistema repetindo a sequência anterior.

O sistema pode transferir dados para o PE (85) de uma maneira análoga, como segue. O sistema coloca dados para ser transmitidos para o PE (85) no registador (116) e



liga o retentor SCR. O PE (85) é interrompido, explora o STR, encontra o retentor SCR ligado, lê os dados do registador (116) e repõe o retentor (SCR). O sistema pode interrogar o retentor SCR para verificar se foi ou não repostado.

3. O sistema pode transferir outros dados para o PE (85) repetindo a sequência anterior.

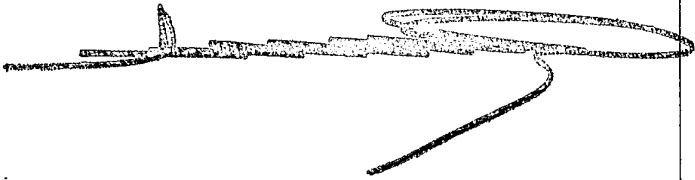
Podem também permutar-se dados através da área IOA da memória (187). Há comandos PROCBUS para o PE (85) e para o adaptador (154) que permitem quer armazenar quer buscar dados em/a partir da área IOA (187).

O PE (85) tem um conjunto de memórias tampão atribuídas na área IOA (187), nas quais coloca dados para serem buscados pelo sistema. Correspondentemente, o sistema tem um outro conjunto de memórias tampão atribuídas na área IOA (187), nas quais coloca dados para serem buscados pelo PE (85). Os tipos de interrupção IOASYS/IOAPU podem ser usados em pedidos SYSREQ para indicar-se mutuamente que se colocaram dados nas memórias IOA.

Certas condições de verificação da máquina e de interrupção externa são activadas pelo sistema de utilização. O sistema comunica uma condição de interrupção ao PE enviando um pedido de comunicação SYSREQ ou XFERREQ. O PE (85) executa as funções seguintes:

- a. Explora o registador STR e interroga o seu conteúdo.
- b. Chama a micro-rotina proporcionada pelo sistema. O dispositivo de tratamento dos pedidos de interrupção do sistema efectua o processamento de interrupção específico. Num instante apropriado, a micro-rotina envia um comando PROCBUS para o adaptador (154) para repor o SYSREQ ou XFERREQ correspondente. Finalmente, devolve o controlo para o microcódigo S/370.
- c. O PE (85) permuta PSW para a classe de interrupção S/370 apropriada e executa a função NSI.

Os pedidos de interrupção I/O são gerados pelo sistema ajustando o bit I/O no registador STR. Quando, de cada vez, se completa a instrução S/370 corrente, é pedido o dispositivo de tratamento e excepções. Nesta rotina, o PE (85) lê o registador STR para reconhecer o pedido de interrupção I/O.



O PE (85) repõe o bit STR e ajusta o retentor do pedido de interrupção interno ao PE (85). Este retentor é mascarado com a máscara I/O da PSW corrente. Se a máscara for 1 e não estiver pendente nenhum pedido de interrupção de prioridade mais elevada, o dispositivo de tratamento de exceções passa o controle para um dispositivo de tratamento de pedidos de interrupção I/O proporcionados pelo sistema que processa o pedido de interrupção I/O.

Linha omnibus do processador (170) (fig. 11 e 30) e comandos da linha omnibus do processador

A linha omnibus do processador (170) é a ligação comum entre todos os componentes do conjunto de micro pastilhas S/370. Logicamente, todas as linhas listadas adiante pertencem a esta linha omnibus:

1. As linhas (0-31 + 4 paridade) da linha omnibus do processador são geralmente usadas para transferir um comando juntamente com um endereço num ciclo e depois para transferir os dados associados no ciclo seguinte. A permissão para utilizar a linha omnibus é dada por um árbitro de preferência situado no adaptador das linhas omnibus (154). O PE (85) tem a prioridade mais baixa. Quando for dada a permissão através da linha BUS GRANT PE85, o PE (85) coloca quatro itens nas linhas apropriadas da linha omnibus no ciclo seguinte. Para uma operação de acesso à memória, o comando colocado nas linhas 0-7 da linha omnibus do processador, o endereço é posto nas linhas 8-31 da linha omnibus do processador, uma chave de acesso é posta na linha omnibus de estados da chave (KEY STATUS BUS) e simultaneamente é excitado um sinal "N-Command-Valid".
2. A linha omnibus KEY/STATUS (0-4 + paridade) é usada para dois fins: para emitir uma chave de acesso para a memória e para obter em retorno uma informação de estado. São transferidos quatro bits da chave de acesso PSW S/370 mais um quinto bit que representa o resultado de uma operação lógica E do bit do modo de controle PSW (BC ou EC) e do bit de tradu



ção dinâmica do endereço.

O estado devolvido deve ser zero para uma operação boa. Um estado não-zero provoca na maioria dos casos uma rejeição para PE (85). Não é esperado qualquer estado para comandos do tipo "mensagem" que ajustam retentores de controlo na unidade da linha omnibus endereçada.

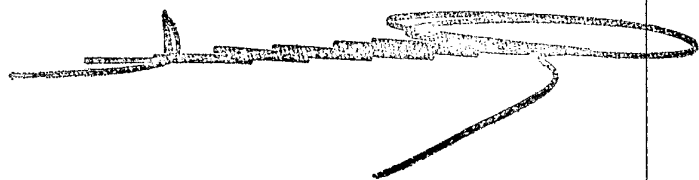
3. A linha N BUS BUSY proporciona uma indicação de ocupação sempre que uma operação não pode ser completada no mesmo ciclo em que foi iniciada. A linha N-BUS BUSY é activada pelo PE (85) simultaneamente com a linha N-CMD-VALID para todos os comandos que requerem mais de um ciclo para se completarem.

É da responsabilidade da unidade omnibus endereçada levar a linha N-BUS-BUSY para o nível activo se a execução do comando leva dois ou mais ciclos. A linha N-BUS-BUSY é também levada para o nível activo quando a unidade de linha omnibus endereçada não pode aceitar o comando seguinte durante um par de ciclos. Há uma excepção à regra: PE (85) activará N-BUS-BUSY durante três ciclos se ele enviar comandos de operação da memória para a memória principal do agregado BSM (162). Em geral, N-BUS-BUSY estará no nível activo pelo menos um ciclo a menos que a duração da execução de uma instrução.

4. O sinal de ocupação da unidade de gestão da memória MMU BUSY origina-se no controlador da memória "cache" (153). É usado para indicar ao PE (85) a chegada de estados e dados para todas as operações de acesso à memória que tomem mais de um ciclo para ser executadas.

As operações de busca principalmente fornecem dados no ciclo seguinte (após terem sido iniciadas) ou posteriormente. Se se fornecerem dados e estados no ciclo seguinte, o sinal MMU-BUSY mantém-se inactivo no nível baixo (0). Se não puderem fornecer-se dados e estados no ciclo seguinte, MMU-BUSY é elevado para o nível 1 e retorna ao 0 no ciclo em que são efectivamente colocados dados e estados na linha omnibus.

Durante as operações de memória, o PE (85) espera estados na linha omnibus KEY STATUS BUS no ciclo



- seguinte (após ter sido iniciada a operação de memória). Se não puderem fornecer-se estados no ciclo seguinte, MMU-BUSY mantém-se inactivo (0); além disso é elevada para o nível 1 e volta ao 0 no ciclo em que efectiva é fornecido um estado.
5. O indicador de falha da memória "cache" na linha MISS IND é usado pelo controlador da memória "cache" (153) para indicar uma falha de DLAT, uma falha da chave ou uma violação de endereçamento para o PE (85). A indicação é uma duplicação de informação que está disponível também no estado. A linha está válida no mesmo ciclo em que se apresentou o estado na KEY STATUS BUS, mas a linha de indicação de falha é activada poucos nanossegundos mais cedo. A indicação de falha força uma rejeição através do PE (85) no ciclo seguinte.
 6. O sinal na linha BUS GRANT PE85 dá a permissão para a utilização da linha omnibus para o PE (85). O sinal tem origem no árbitro. PE (85) subsequentemente coloca o comando e o endereço para operação desejada na linha omnibus no ciclo que se segue aquele em que o sinal de permissão se tornou activo e N-BUS-BUSY não está activo.
 7. Utilização: o sinal de pedido de atenção na linha N-ATTN-REQ origina em alguma outra unidade de linhas omnibus (tal como o adaptador de linhas omnibus (154)) o pedido do PE (85) para efectuar a operação "sense". O PE (85) acata o pedido logo que se complete a operação corrente em curso (por exemplo a execução de uma instrução).
 8. O sinal de comando válido na linha N-CMD-VALID é usado pelo PE (85) para indicar que o padrão de bits nas linhas 0-31 da linha omnibus do processador e as linhas 0-4 (incluindo todas as linhas de paridade) de KEY STATUS BUS é válido. Esta linha pode ser activada (nível baixo) no ciclo que se segue àquele em que BUS-GRANT-PE85 se activou e N-BUS-BUSY se desactivou.
 9. A linha ADDR-DECREMENT é usada pelo PE (85) para operações de acesso à memória que procede do endereço inicial para posições no sentido descendente (tais como as necessárias para a transferência de dados de processamento de dados de cima para baixo). O sinal pode ser activado no mesmo ciclo em que se activou N-CMD-VALID.

10. O sinal de anulação do comando na linha CMD-CANCEL é usado pelo PE (85) para anular um acesso à memória de busca já iniciada. Isso pode ocorrer no ciclo depois de se activar N-CMD-VALID quando o PE (85) detectar condições que inibem o uso imediato dos dados pedidos.

Na forma de realização preferida, há quatro grupos de comandos PROCBUS de tipos geralmente bem conhecidos:

CPU-Storage; I/O-Storage; MMU Operation; Message Exchange e Floating Point

A unidade de linhas omnibus (PE (85), adaptador (154) ou controlador da memória "cache" (153)) que pede o controlo da linha omnibus (171) coloca o comando na linha omnibus. Para os comandos CPU-Storage e I/O-Storage, a unidade de linhas omnibus também coloca o bit de tradução da chave de acesso e do endereço dinâmico na linha omnibus KEY STATUS BUS. Depois de completado o comando, devolve-se o estado na mesma linha omnibus para a unidade de linha omnibus que fez o pedido.

O adaptador (154) envia comandos CPU-Storage e comandos I/O-Storage enquanto que o PE (85) apenas pode enviar comandos CPU-Storage. Estes grupos de comandos são como segue:

OPeração	Comando da memória CPU	Comando da memória I/O
1. Referência da memória principal S/370		
a) FETCH		
Cache Hit	Busca da memória "cache"	Busca da memória "cache"
Cache Miss	Recarregar a linha da memória "cache" a partir da memória (incluindo CAST OUT) e busca a partir da memória "cache"	
b) STORE		
Cache Hit	Armazenar na memória "cache"	Armazenar na memória "cache"

Cache Miss

Recarregar a linha da memória "cache" (incl. EAST OUT) e armazenar na memória "cache"

2. Referência da área objecto interna (IOA)

Certos comandos da memória CPU permitem o acesso à verificação dos endereços de acesso à memória IOA

Os comandos I/O-Storage são executados no controlador da memória "cache" (153) sem verificação do endereço da memória principal S/370. Esta verificação é executada em STC1 (155). Os comandos CPU-Storage são dirigidos para o controlador (153) para execução e têm um campo de comando, de um octeto, e um campo de endereço real ou virtual, de três octetos.

Os bits do campo de comando são os seguintes:

Bit do comando	Significado
0 - 1 = 10	Comando da memória CPU
2 = 1	Operação de busca
2 = 0	Operação de armazenamento
3 = 1	Curtocircuito da memória "cache", nenhuma verificação de endereço
3 = 0	Acesso à memória "cache" com verificação de endereço - comparar endereço S/370 - verificar ACB
4 = 1	Nenhum acesso DLAT; isto é, - nenhuma verificação de protecção controlada pela chave - nenhum tratamento de bits de referência e de permuta Acesso DLAT; isto é, - verificação da protecção controlada pela chave - tratamento de bits de referência e permuta

5 - 7 = nnn

Contagem do comprimento em octetos:

000 = 1 octeto

001 = 2 octetos

010 = 3 octetos

011 = 4 octetos

100 = 8 octetos

101 = 64 octetos

110 = 64 octetos FETCH! retardamento de
BSM

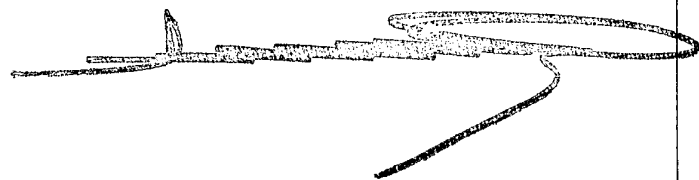
111 = 64 octetos FETCH! retardamento do
adaptador

Exemplos de comandos de memória CPU são:

1. Fetch (10111nnn)/store (10011nnn) Real N Byte, para buscar ou memorizar até 64 octetos de/para a memória (162) com um endereço real.
2. Fetch (101010nn)/store(100010nn) Cache Real N Byte para leitura/escrita até 4 octetos de/para a memória "cache" com um endereço real.
3. Fetch (101011nn)/store(100011nn) Cache Real N Byte para leitura/escrita até 4 octetos de/para IOA com um endereço real (100000nn).
4. Fetch (101000nn)/store(100000nn) Cache Virtual N Byte para leitura/escrita até 4 octetos de/para a memória "cache" com um endereço virtual.

Os comandos I/O-storage (memorização I/O) são iniciados pelo adaptador (154) e dirigidos para o controlador da memória "cache" (153). Eles transferem correntes de dados com o comprimento de 1 - 64 octetos, por ordem ascendente dos endereços. O formato de comandos de 32 bits inclui um endereço real do octeto nos três octetos de ordem inferior e o octeto de ordem superior inclui um bit "0" da ordem mais elevada, o bit seguinte de ordem mais elevada define uma operação de busca ou de armazenamento e os restantes seis bits definem o comprimento da transferência de dados (1 - 64 octetos). As correntes de dados são transferidas nos limites de palavras, excepto para a primeira e a última transferências, que podem exigir o alinhamento de posição na linha omnibus.

Os comandos MMU são usados para comandar



o controlador da memória "cache" e os seus registadores incluem o DLAT, ACB, lista de endereços e outros similares.

Os comandos de mensagens são usados para transferir mensagens entre as unidades de linhas omnibus ligadas à linha omnibus (151).

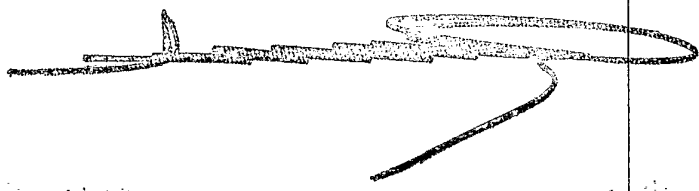
Unidade de gestão da memória S/370 (81)

1. Controlador da memória "cache" (153) (fig. 31)

O controlador da memória "cache", fig. 31, inclui a memória "cache" (340) e a lógica de endereçamento e comparação (347,348), um alinhador de busca (343), bem como a tabela separada de consulta da lista (DLAT) (341) para a tradução rápida dos endereços. O controlador (153) aceita endereços virtuais e comandos de memória da linha omnibus do processador (170) e transfere comandos de busca ou de memorização para a interface (155) (fig. 11) de controlo da memória, através do multiplexador (349) e da linha omnibus STC (157), quando não puder satisfazer o pedido através da memória "cache" (340).

A tabela DLAT (341) proporciona a tradução rápida de endereços virtuais de páginas em endereços reais de páginas. As suas 2 x 32 entradas contêm 64 endereços de páginas pré-traduzidos. A DLAT (341) é acedida utilizando um esquema de endereçamento associado ajustado por duas vias. A dimensão das páginas virtuais é de preferência 4 KB. No caso de falta da DLAT, o PE (85) é interrompido e a tradução do endereço virtual é feita por microprograma usando tabelas de segmentos e páginas (não representadas) na memória principal S/370 (162), de uma maneira bem conhecida. A DLAT (341) é depois actualizada para reflectir o novo endereço das páginas virtuais e reais da informação buscada a partir da memória e colocada na memória "cache". Uma cópia da chave da memória é buscada a partir da memória S/370 e incluída na entrada da DLAT.

A memória "cache" de 8 KB (340), com a sua lista (342), proporciona uma memória tampão de elevada velocidade para melhorar de maneira significativa a eficácia do processador. As matrizes de dados e da lista estão divididas em 4

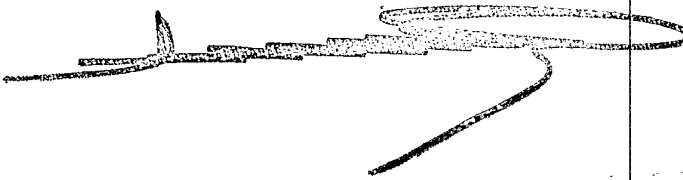


compartimentos. Cada compartimento na memória "cache" está organizado em 256 x 8 B (octetos). Para a busca de dados na memória "cache" (340), o desvio em octetos no endereço virtual é usado para endereçar simultaneamente a tabela DLAT (341), a lista de endereços da memória "cache" (342) e a memória "cache" (340). A verificação de protecção, controlada pela chave, é feita pelo circuito de comparação (345) utilizando a chave da memória na entrada DLAT seleccionada. São retidos 4 x 8 B de dados na saída (340a) da memória "cache" (340). Se os dados pedidos estiverem na memória "cache" (340), utiliza-se um último sinal de selecção para encaminhar os octetos apropriados para o alinhador de busca (343).

Para as operações de armazenamento efectua-se um armazenamento parcial numa base de octetos.

No caso de uma falta da memória "cache" o controlador da memória "cache" (153) estabelece automaticamente um comando BSM para a busca da linha de memória "cache" de 64 B no modo descontínuo por grupos. Se a linha da memória "cache" a substituir por uma nova linha da memória "cache" foi alterada desde que foi carregada, inicia-se uma operação de expulsão para a memória (162) antes de se carregar a nova linha de memória "cache". Dados de entrada/saída (I/O) nunca provocam operações de expulsão de linhas da memória "cache" e de carga de linhas da memória "cache". Os dados I/O a buscar na memória (162) serão consultados, quer na memória principal (162), quer na memória "cache" (340) acedendo às duas facilidades. Se se verificar uma resposta positiva na memória "cache", é cancelada a operação da memória, sendo os dados fornecidos pela memória "cache". Se os dados I/O não estiverem na memória "cache", eles serão buscados directamente na memória, mas não se substituirá qualquer linha na memória "cache". Os dados I/O a armazenar na memória serão armazenados na memória "cache" (340) se a linha endereçada estiver já na memória "cache"; caso contrário, serão armazenados directamente na memória (162).

A memória de chave de 4 KB (344) retém as chaves da memória para a memória 16 MB. A memória de chaves é uma matriz organizada de 4K x 8. Cada octeto contém uma chave de memória. Cada entrada da tabela DLAT contém uma cópia da cha



ve de memória associada com o seu endereço do bloco de 4 KB. Isso reduz de maneira significativa o número de acessos à memória de chaves enquanto acede repetitivamente a uma página. As alterações das atribuições das chaves da memória afectam, quer a memória de chaves, quer quaisquer cópias na memória "cache".

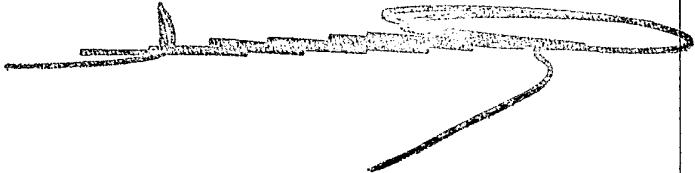
Os comandos, dados e endereços recebidos pelo controlador de memória "cache" (153) da linha omnibus do processador (170) através do circuito receptor (355) são armazenados nos registadores de comandos, de dados e de endereços (350), (351) e (352). O registador de endereços (347) armazena a gama de endereços válidos para o elemento de processamento PE (85) S/370 associado. A lógica de comparação CMP (348) verifica a validade do endereço recebido. A função de comparação de endereços S/370 proporcionada pelo registador de endereços (347) e a sua lógica de comparação associada (348) trata endereços provenientes, quer do PE (85), quer do adaptador de linhas omnibus (154).

A função de comparação do registador de limites de comparação de endereços (ACB) (353) assegura que as referências da memória principal S/370 destinadas à área do utilizador não endereçam a área IOA. O registador ACB (353) armazena a linha divisória (limite) entre a área IOA reservada e a área não reservada da memória S/370 (162). Cada acesso à memória S/370 conduz à comparação pela lógica (354) do endereço recebido com o valor ACB.

2. STCI (155) (fig. 32A,B)

a) Introdução

A interface de controlo da memória (STCI) (155) liga o conjunto de micropastilhas S/370 à memória duplexada tolerante das avarias (16,18), através da lógica das linhas omnibus (178) e da linha omnibus do sistema (30) (fig. 1). Ela apoia todos os comandos do processador S/370 e de armazenamento/busca I/O que definem transferências de dados a partir de 1 - 64 octetos por comando. Todos os ECC, refrescamentos, inicialização e configuração da memória, recuperações de informa-



ções, etc., são tratados pelo processador S/88 (62) e pelas memórias (16,18). Nas fig. 32A,B está representado um fluxo de dados pormenorizados da STCI (155).

A STCI (155), a STCI emparelhada (155a) (não representada numa unidade de gestão da memória (83)) e o seu par de STCI correspondente (não representado) e a unidade parceira (23) (fig. 8), em conjunto, arbitram o controlo da estrutura de linhas omnibus (30) através de uma arbitragem tal como a lógica (408) (fig. 32B) em cada STCI. Não só as STCI (155) arbitram contra os controladores I/O e outras CPU (25), (27) e (29), (31) do módulo (9), como se vê na fig. 7, mas também as STCI têm de arbitrar contra o seu processador S/88 (62) associado (e os processadores emparelhados e emparceirados com esse processador nas CPU (21,23) da fig. 8) que podem estar a pedir o controlo da linha omnibus para funções I/O S/370 ou funções S/88 convencionais.

Porém, a lógica de arbitragem é de resto genericamente análoga à descrita na patente de Reid, baseada primariamente em posições de ranhuras no painel traseiro do processador e cartões I/O, lógica que será agora descrita. Durante uma fase de arbitragem, qualquer unidade do módulo do processador (9) susceptível de ser um módulo director das linhas omnibus e que esteja pronto para iniciar um ciclo de linha omnibus arbitra o uso da estrutura de linhas omnibus. A unidade faz essa arbitragem declarando um sinal de Pedido de Ciclo de Linha Omnibus e verificando simultaneamente, por meio de uma rede de arbitragem, se há unidades de prioridade mais elevada que estejam também a declarar um sinal de pedido de ciclo de linhas omnibus. A unidade, ou par de unidades emparceiradas, que tem êxito na obtenção do acesso à estrutura de linhas omnibus durante a fase de arbitragem é chamada directora da linha omnibus e inicia um ciclo de transferência durante a fase de relógio seguinte. Nunca nenhuma das unidades de memória (16,18) é uma unidade directora e não arbitra.

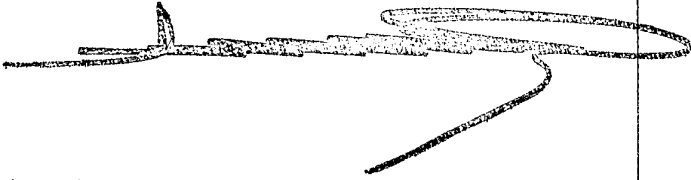
Durante a fase de definição de um ciclo, a unidade que é determinada para ser a directora das linhas omnibus para o ciclo define o tipo de ciclo, produzindo um conjun



to de sinais de definição de ciclo ou de função. A unidade directora da linha omnibus também declara os sinais de endereço e coloca na linha de paridade do endereço uma paridade par para os sinais de endereço e de função. Todas as unidades do módulo processador, independentemente do seu estado interno de operação, recebe sempre os sinais nos condutores da linha omnibus que transportam os sinais de função e de endereço, embora as unidades de controlo dos periféricos possam operar sem receber sinais da paridade. O ciclo que é definido é abortado se nesse instante for declarado um sinal Bus Wait.

Durante a fase de resposta, qualquer unidade do sistema endereçada que esteja ocupada pode declarar o sinal Bus Busy (linha omnibus ocupada) para abortar o ciclo. Uma unidade de memória, por exemplo, pode declarar um sinal Bus Busy se for endereçada quando estiver ocupada ou durante um ciclo de refrescamento. Um sinal Bus Error (erro na linha omnibus) declarado durante a fase de resposta fará abortar o ciclo, pois o erro pode ter sido no endereço dado durante a fase de definição do ciclo. Os dados são transferidos quer na linha omnibus A quer na linha omnibus B durante a fase de transferência de dados, quer o ciclo de leitura quer para o ciclo de escrita. Isso permite ao sistema canalizar uma mistura de ciclos de leitura e de ciclos de escrita na estrutura da linha omnibus sem recorrer à re-arbitragem para a utilização das linhas de dados e sem ter de colocar indicadores nos dados relativos à unidade de origem ou à unidade de destino.

As transferências de palavras completas são acompanhadas pela declaração quer de sinais UDS, quer LDS (sinais de impulsos de sincronização de dados superiores e inferiores). As transferências de meias-palavras ou de octetos são definidas como transferências acompanhadas pela declaração de apenas um destes sinais de impulsos de sincronização. As transferências de escrita podem ser abortadas cedo no ciclo pela unidade directora da linha omnibus, simplesmente não declarando nenhum dos sinais de impulsos de sincronização. As unidades dirigidas, que estão a ser lidas, têm de declarar os sinais de impulsos de sincronização com os dados. Os sinais de impulsos de sincronização são incluídos calculando a paridade dos dados na



linha omnibus.

Erros detectados durante a fase de transferência de dados farão com que a unidade que detecta o erro declare um ou ambos os sinais de Bus Error na fase de distribui-ção de tempos seguinte, que é a primeira fase após os dados. As unidades de controlo de periféricos aguardam para ver se ocorre um erro antes de utilizar os dados. A unidade central de processamento (21) e a unidade da memória principal (16) do sistema, no entanto, usam os dados logo que são recebidos e, na eventualidade de um erro, com efeito recuam e aguardam pelos dados correctos. A declaração de um sinal Bus Error durante uma fase pós-dados faz com que se repita a fase de transferência durante na próxima fase, a sexta, do ciclo de transferência. Isso faz abortar o ciclo, se existir algum, que de outro modo transmitiria dados pela estrutura da linha omnibus durante esta segunda fase pós dados, isto é, a sexta.

O modo "backplane" normal de funcionamento do sistema ilustrado é quando todas as unidades estão no modo Obey Both (Obedecer a Ambos), no qual, quer a linha omnibus A, quer a B aparecem sem erros. Em resposta a um erro na linha omnibus A, por exemplo, todas as unidades comutam sincronizadamente para o modo Obey B (Obedecer a B). O módulo (9) regressa ao modo de funcionamento Obey Both por meio de meios de programação de supervisão executados numa unidade central de processamento S/88.

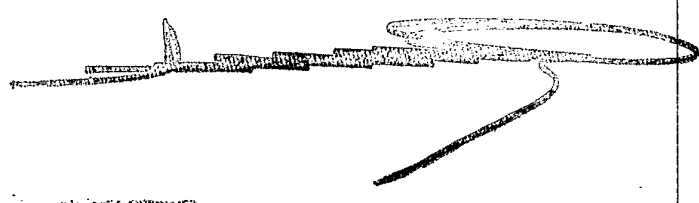
Em ambos os modos de funcionamento Obey A e Obey B, ambas as linhas omnibus A e B são accionadas pelas unidades do sistema e todas as unidades efectuam ainda uma verificação dos erros plana. A única diferença de funcionamento no modo Obey Both é que as unidades simplesmente registam outros erros na linha omnibus que não está a ser obedecida, sem exigir que se repitam os dados, sem fazer abortar quaisquer ciclos. Mas um sinal Bus Error na linha omnibus obedecida é tratado como anteriormente e faz com que todas as unidades comutem para obedecer à outra linha omnibus.

b) Fases da linha omnibus do sistema

A fig. 33 ilustra a operação anterior com quatro ciclos de transferência multifásicos canalizados na estrutura de linhas omnibus (30) para o módulo (9). As curvas (56a) e (56b) mostram o relógio principal e os sinais de sincronização directores que o relógio (38) aplica à linha omnibus X (46), para 21 fases de distribuição de tempos sucessivas, numeradas de (1) a (21), como se indica na parte superior do desenho. Os sinais de arbitragem na estrutura de linhas omnibus, representados com as curvas (58a), mudam no início de cada fase da distribuição de tempos para iniciar, em cada uma das 21 fases ilustradas, a arbitragem de um novo ciclo, como se anota com a legenda de numeração dos ciclos ($\#1$), ($\#2$), ($\#3$), ..., ($\#21$). A fig. 33 representa sinais de definição de ciclos com a forma de onda (58b). Os sinais de definição de ciclos, para cada ciclo, ocorrem uma fase de relógio mais tarde que os sinais de arbitragem para esse ciclo, como se indica com os números dos ciclos na curva (58b). O desenho representa ainda os sinais BUSY, WAIT, DATA, A BUS ERROR e B BUS ERROR. A fiada inferior do desenho indica o modo "backplane" e representa as transições.

Ainda com referência à fig. 33, durante a fase de distribuição de tempos número 1, o módulo (9) produz os sinais de arbitragem de ciclos para o ciclo $\#1$. O sistema está a operar no modo Obey Both, como foi designado. A unidade directora da linha omnibus determinada durante o ciclo de arbitragem da fase 1 define o ciclo e efectuar durante a fase de distribuição de tempos (2), como está designado com a referência $\#1$ na curva do sinal de definição de ciclos (58b). Também na fase de distribuição de tempos (2) é efectuada a arbitragem para um segundo ciclo, ciclo $\#2$.

Durante a fase de distribuição de tempos (3) não há sinal de resposta na estrutura de linhas omnibus para o ciclo $\#1$, o que indica que este ciclo está pronto para prosseguir com uma transferência de dados como se verifica durante a fase (4) da distribuição de tempos e está designado por 1 na curva de dados (58e). Também durante a fase 3 da distribuição de tempos é efectuada a arbitragem do ciclo para o ciclo



2 e a arbitragem para um outro ciclo ## 3.

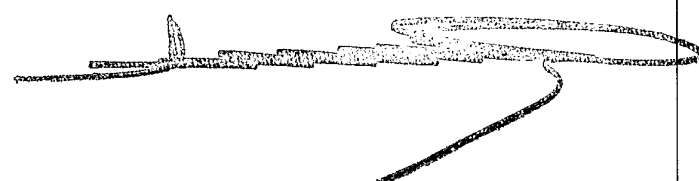
Na fase 4 da distribuição de tempos, transferem-se os dados do ciclo ## 1 e efectua-se a definição para o ciclo ## 3. Também é declarado um Bus A Error durante esta fase da distribuição de tempos como se indica com a curva (58f). O sinal de erro faz abortar o ciclo ## 2 e comuta todas as unidades no módulo para o modo Obey B. O sinal Bus A Error da fase 4 da distribuição de tempos indica que na fase anterior 3 da distribuição de tempos pelo menos uma unidade do sistema detectou um erro relativamente aos sinais provenientes da linha omnibus (A) (42). O erro verificou-se quando não havia nenhum dado na estrutura de linhas omnibus, como se indica pela ausência de dados na curva (58b) durante a fase 3 da distribuição de tempos, não havendo portanto necessidade de repetir uma transferência de dados.

Durante a fase 5 da distribuição de tempos, com o sistema operando no modo Obey B, é arbitrado um quinto ciclo, define-se a função para o ciclo ## 4 e não está presente qualquer resposta na estrutura da linha omnibus para o ciclo ## 3. Por conseguinte, esse ciclo procede à transferência de dados durante a fase 6 da distribuição de tempos. Também na fase 6 da distribuição de tempos é declarado um sinal Bus Wait, como aparece na curva (58d); isso está em ligação com o ciclo ## 4. O efeito é estender esse ciclo para uma outra fase da distribuição de tempos e fazer abortar o ciclo ## 5.

Arbitra-se um novo ciclo ## 7, na fase 7 da distribuição de tempos, e a operação de definição prossegue para o ciclo ## 6. Na fase 8 da distribuição de tempos, aplicam-se os dados para o ciclo ## 4 na estrutura da linha omnibus para transferência. Também na fase 8 da distribuição de tempos é declarado um sinal BUSY. Este sinal faz parte da resposta ao ciclo ## 6 e faz abortar o ciclo.

As operações de arbitragem e definição na fase 9 da distribuição de tempos seguem o mesmo padrão com declaração de outro sinal Bus A Error. O sistema já está a operar no modo Obey B e por conseguinte a resposta a este sinal é simplesmente registar o erro.

O sinal BUS WAIT declarado na fase 10



da distribuição de tempos e que continua para a fase 11 estende o ciclo $\neq 8$ por mais duas fases da distribuição de tempos, de modo que os dados para esse ciclo são transferidos durante a fase 13 da distribuição de tempos, como está indicado. O sinal BUS WAIT declarado durante estas fases também faz abortar os ciclos $\neq 9$ e $\neq 10$, como está representado. Qualquer sinal BUSY declarado durante as fases 10, 11 ou 12, com vista à extensão do ciclo $\neq 8$ pelo sinal WAIT, faria abortar o ciclo $\neq 8$. Notar que a transferência dos dados para o ciclo $\neq 7$ ocorre na fase 10 da distribuição de tempos independentemente dos sinais nos condutores WAIT e BUSY durante esta fase da distribuição de tempos.

Outros sinais BUS A ERROR que ocorram durante as fases 11, 12 e 14 da distribuição de tempos mais uma vez não têm qualquer efeito no sistema senão o de serem registrados, porque o sistema já está em operação no modo Obey B. O sinal WAIT declarado durante a fase 14 da distribuição de tempos faz abortar o ciclo $\neq 13$. Também estende o ciclo $\neq 12$, que é no entanto abortado pelo sinal BUSY declarado durante a fase 14 da distribuição de tempos. Além disso, a transferência de dados para o ciclo $\neq 14$ verifica-se na fase 17 da distribuição de tempos.

Na fase 19 da distribuição de tempos, imediatamente a seguir ao ciclo $\neq 15$ da transferência de dados da fase 19 da distribuição de tempos, é declarado um sinal BUS B ERROR. Este sinal de erro faz abortar o ciclo $\neq 17$, que está na fase de resposta, e inicia uma repetição da transferência de dados para o ciclo $\neq 15$. A transferência repetida ocorre durante o ciclo $\neq 20$. Além disso, este sinal de erro comuta o módulo para o modo Obey A.

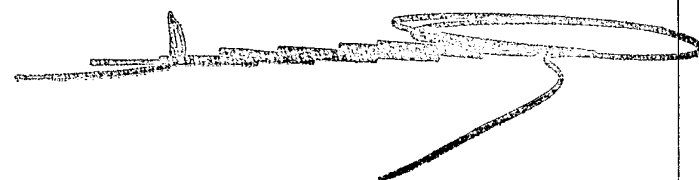
Notar que o sinal BUS WAIT é activado apenas pelas unidades directoras que tenham sido endereçadas por uma unidade directora da linha omnibus e não estão prontas para efectuar uma transferência de dados. Como a STCI (155) nunca é uma unidade dirigida e apenas endereça a memória, e não dispositivos I/O, esta linha não é utilizada pela STCN (155).

A lógica (178) da linha omnibus do sistema proporciona o elo de ligação da STCI (155) para os cartões



de memória S/88 (16,18) e inclui a lógica de arbitragem (408) (fig. 32B). Os mesmos ciclos de transferência básicos atrás definidos para a linha omnibus (30) são usados pela lógica (178):

1. Fase de arbitragem - Esta fase prossegue em todos os ciclos enquanto os controladores das linhas omnibus disputarem a posse da linha omnibus. Tipicamente a prioridade da arbitragem baseia-se no painel traseiro SLOT ID dos dispositivos de arbitragem. Para a forma preferida da concepção da interface STCI a prioridade das arbitragens baseia-se na SLOT IT para CPU simples, enquanto que se utilizam as linhas (409) do indicador FIFO Almost Full/Almost Empty (AFE) e do indicador Half-Full (HF) em cada CPU PE (85) e a sua unidade em parêntese) para atribuir paridades baseadas no pedido real de tarefas em realizações com CPU múltiplas.
2. Fase de definição de ciclos - Esta fase segue-se a uma aceitação pela linha omnibus do ciclo anterior. Inclui um código de função de 4 bits em BUS FN CODE A e B da linha omnibus (30) para especificar transferências de 16, 32 ou 64 bits Leitura/Escrita (R/W) juntamente com o endereço físico inicial de 27 bits para a memória (16). A memória (16) é de 256 MB, para a forma de realização preferida. Todos os acessos da memória são nos limites de 16, 32 ou 64 bits, de modo que não é usado o bit 0 de endereços. De preferência o acesso de octetos e de palavras é indicado pelos sinais UDS, LDS representados na fig. 14, em ligação com a definição de BUS FN CODE.
3. Fase de resposta do ciclo - Esta fase pode incluir uma condição de BUS ERROR ou BUS BUSY na linha omnibus (30) a partir da memória, o que forçará a interface STCI a re-arbitrar e reenviar a fase de definição de ciclo anterior.
4. Fase de dados - Uma vez aceite o pedido de memória (depois da fase de resposta ao ciclo) verificar-se-á a fase de dados no ciclo que se segue à fase de resposta do ciclo (2 ciclos depois da fase de definição do ciclo). Podem transferir-se 16, 32 ou 64 bits dentro de uma fase de 125 ns, em leitura ou escrita.
5. Fase pós-dados - Necessária para verificar se há erros na linha omnibus, o que forçará a repetição dos dados (ou a



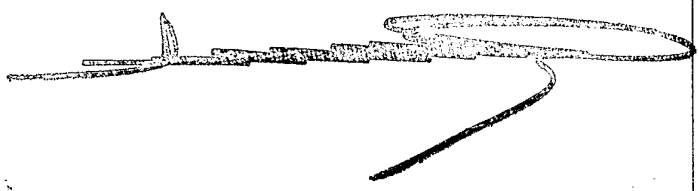
partir da interface STCI (155) ou da memória (16)) na linha omnibus do sistema (30) depois de os dados serem enviados inicialmente. Como as duas linhas omnibus A e B transmitem dados idênticos, podem ocorrer erros quer da linha omnibus A quer da B, durante as fases pós-dados.

Uma diferença importante entre o processador S/88 (62) arbitrar a linha omnibus (30) e a interface STCI (155) arbitrar a linha omnibus (30) pode agora ser descrita. Tipicamente, um processador S/88 (62) será operado em apenas uma das cinco fases em qualquer instante. Porém, devido à capacidade de canalização de informações de busca e de memória na interface STCI (155) (descrita mais adiante), a interface STCI pode operar em todas as fases ao mesmo tempo. Por exemplo, durante uma operação de leitura de 64 octetos, a interface STCI (155) pode ser operada em todas as cinco fases ao mesmo, se não houver quaisquer erros e se à interface STCI for concedida um controle de arbitragem da linha omnibus (30) em cada um dos cinco ciclos consecutivos. Isso melhora a eficácia do sistema, especialmente numa versão uni-processador de um módulo (9).

c) Características da interface STCI

Descrevem-se a seguir algumas das características da interface STCI:

1. FIFO (400) - Quatro memórias RAM rápidas (64 x 9 bits) de atendimento por ordem de chegada ("First-IN-First-OUT) formam uma memória tampão para permitir reter até quatro comandos de memória de 64 octetos antes de a unidade (155) ficar no estado de ocupada. Transporta também paridade de entrada para saídas de todos os dados. O relógio S/370 (152) comanda por impulsos de relógio a entrada de comandos e dados na FIFO (400) e o relógio S/38 comanda por impulsos de relógio a saída de comandos e dados da FIFO (400). Uma forma de realização preferida da memória FIFO (400) é a CY7C409, descrita mais completamente, a começar na página 5-34 no Product Information Manual publicado em 15 de Janeiro de 1988 por Cypress Semiconductor Corp.



Além dos sinais de sincronização por passagem de testemunho normais na indústria, proporcionam-se também indicadores AFE ("Almost Full/Almost Empty" - Quase cheio-Quase vazio) e HF ("Half Full"-Semi-cheio). AFE está no nível elevado quando FIFO está quase cheia ou quase vazia. Caso contrário AFE tem o nível baixo. HF tem o nível alto quando FIFO está semi-cheia; caso contrário tem o nível baixo.

A memória aceita palavras de 9 bits em paralelo nas suas entradas sob o controlo da entrada Shift-In (SI) quando o sinal de controlo "Input-Ready" (IR) tiver o nível elevado. A saída dos dados faz-se pela mesma ordem por que foram memorizados sob o controlo da entrada Shift-Out (SO), quando o sinal de controlo Output-Ready tiver o nível elevado. Se a memória FIFO estiver cheia (IR no nível baixo) são ignorados os impulsos na entrada SI; se FIFO estiver vazia (OR no nível baixo) são ignorados os impulsos na entrada SO.

A ampliação em paralelo para palavras mais compridas é realizada por combinação lógica, em circuitos E, das saídas IR e OR (respectivamente) das memórias FIFO individuais. A operação lógica E garante que todas as memórias FIFO estão prontas para aceitar mais dados (IR no nível elevado) ou então prontas para dar saída a dados (OR no nível elevado), compensando assim as variações dos tempos de atraso da propagação entre os dispositivos.

AS operações de leitura e escrita são completamente assíncronas, permitindo que a memória FIFO seja usada como memória tampão entre duas máquinas digitais com frequências de relógio de operação muito diferentes ou com fases de relógio muito diferentes. A memória FIFO (400) inclui um indicador de escrita e um indicador de leitura e a lógica de controlo necessária para gerar sinais de sincronização com permuta de informações conhecidas (SI/IR, SO/OR), bem como indicadores AFE e HF. Com a memória FIFO vazia, a lógica da interface STCI manterá SO no nível elevado, de modo que, quando se escreve uma palavra, ela atingirá rapidamente a saída. O sinal OR tornar-se de nível elevado durante



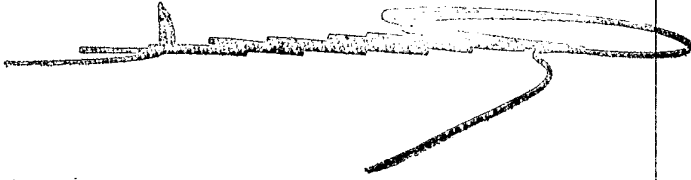
um ciclo interno, voltando depois de novo ao nível baixo. Se se escreverem mais palavras na memória FIFO, elas alinham-se depois da primeira palavra e não aparecerão nas saídas até SO se tornar de nível baixo.

Os dados não se propagam fisicamente através da memória. Em vez de deslocar os dados incrementam-se os indicadores de leitura e escrita. O tempo necessário para incrementar o indicador de escrita e propagar um sinal da entrada SI para a saída SO de uma memória FIFO vazia ("fallthrough time") ou o tempo necessário para incrementar o indicador de leitura e propagar um sinal da entrada SO para a saída IR de uma FIFO cheia ("bubblethrough time") determinam a velocidade a que os dados podem passar através da memória FIFO (400).

Quando se liga a alimentação, a memória FIFO é reposta com um sinal Master Reset. Isso faz com que o dispositivo entre na condição de vazio, o que é indicado pelo facto de o sinal OR ter o nível lógico baixo ao mesmo tempo que o sinal IR tem o nível lógico elevado. Nesta condição, as saídas de dados (DOO-DO8) terão o nível lógico baixo. O indicador AFE terá o nível alto e o indicador HF o nível baixo.

A disponibilidade de uma posição vazia é indicada pelo estado de nível lógico elevado do sinal Input Ready (IR). Quando IR tiver o nível lógico alto, uma transição de nível baixo para nível alto no perno Shift-In (SI) carregará os dados nas entradas para a memória FIFO (400). A saída IR irá então para o nível lógico baixo, indicando que os dados foram amostrados. A transição do nível alto para o nível baixo do sinal SI inicia a transição do nível baixo para alto do sinal IR, bem como a transição do nível baixo para o nível alto do indicador AFE se a memória FIFO (400) estiver quase cheia ou quase vazia.

A disponibilidade de dados nas saídas da memória FIFO (400) é indicada pelo estado de nível elevado do sinal OUTPUT READY (OR). Depois de a memória FIFO ser reposta, todas as saídas de dados (DOO-DOS) estarão no estado de nível baixo. Enquanto a memória FIFO estiver vazia, o

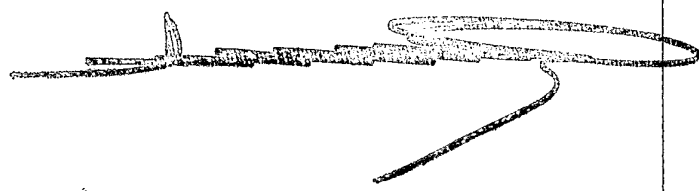


sinal OR terá o nível baixo e todos os impulsos SHIFT OUT (SO) aplicados à mesma serão ignorados. Depois de os dados terem sido deslocados para o interior da memória FIFO o sinal OR irá para o nível elevado.

Dois indicadores, ALMOST FULL/ALMOST EMPTY (AFE) e HALF FULL (HF) descrevem quantas palavras estão armazenadas na memória FIFO. AFE terá o nível elevado quando houver oito ou menos, ou 56 ou mais, palavras armazenadas na memória FIFO. Caso contrário, o indicador AFE tem o nível baixo. HF tem o nível elevado quando houver 32 ou mais palavras armazenadas na memória FIFO, caso contrário, o indicador HF terá o nível baixo. As condições do indicador verificam-se em relação com os flancos descendentes de SI e SO.

2. Lógica SBI - A lógica (178) da interface da linha omnibus S/88 (SBI), que permite que o processador S/370 (85) inicie a leitura/escrita para a memória S/88 (16), inclui a lógica (408) para arbitrar cada um dos ciclos para acesso à linha omnibus (30), para iniciar transferências de 16, 32 ou 64 bits. As linhas da interface da lógica (178) e a lógica de arbitragem (408) são de preferência do tipo descrito na patente de Reid, na medida em que não sejam modificadas como aqui se descreve.
3. Tolerância das avarias - Toda a lógica da interface STCI, incluindo a memória tampão FIFO (400), é duplexada para proporcionar autoverificação no cartão do processador S/370. A única lógica simplexada inclui a lógica do comparador (402 a-g), a lógica de interrupções (403) e a lógica de geração de impulsos de relógio (não representada). Assim, a interface STCI (155) tem uma interface STCI (155a) (não representada) emparelhada, substancialmente idêntica, que faz parte da unidade de gestão da memória (83) da fig. 8.

A lógica do comparador (402 a-g) constitui a lógica de comparação (15) da fig. 8 e a lógica de interrupção faz parte da lógica comum de controlo (75) da fig. 8. Na forma de realização preferida, a verificação de comparação S/370 é efectuada apenas nas interfaces empare-



- lhadas STCI (155) e (155a), para proteger contra a dispersão de dados errados através da estrutura de linhas omnibus (30). Porém, S/370 os erros de verificação da máquina S/370 e de paridade são fornecidos à lógica (403) através da linha omnibus (460). Alguns erros nas linhas omnibus BCU (247, 223) são captados por circuitos de comparação S/88 (12f) (fig. 8).
4. Verificação de endereços - Proporcionam-se dois registadores (404) e (405), com o mapa da memória (MEM Base & MEM Size) para garantir que a dimensão de cada espaço de memória do processador S/370, tal como (162), não é violado enquanto se utiliza um desvio da base (fig. 10) para gerar um endereço físico válido de utilizador S/370 na memória (16) do sistema S/88.
 5. Operação síncrona - Derivam-se impulsos de relógio S/370 (152) do relógio S/88 (38) (fig. 7), entrada de 16 MHz, através da linha omnibus (30) e da lógica de sincronização (158) (fig. 19c), para permitir a sincronização entre os relógios no interior de um período do relógio de entrada do oscilador S/370 a partir do arranque do relógio S/88 (38). Isso permite canalizar leituras consecutivas (por exemplo um comando de leitura de 64 octetos) da memória (162) para o conjunto de micropastilhas S/370 (150), sem qualquer estado de espera entretanto (considerando ciclos consecutivos admitidos para a interface STCI (155) na linha omnibus do sistema (30)).
 6. Interface da linha omnibus STC - Todos os comandos normais de busca/memorização S/370 são executados juntamente com o cancelamento do comando. Os erros de paridade e/ou erros ECC não serão comunicados ao sistema operativo S/370, mas sim são tratados como tentativas repetidas (erros ECC ou erros de paridade das linhas omnibus) ou interrompidas (erros de paridade internos do cartão). Os atravessamentos da linha de fronteira dos 64 octetos terão como consequência uma viragem dos endereços.

Como se representa na fig. 11, a interface STCI (155) é uma interface para o processador S/370 (85), através da unidade de controlo da memória "cache"



(153), que trata a tradução dinâmica dos endereços (virtuais) S/370, utilizando uma memória "cache" de instruções/dados de 8 KB, bem como uma tabela DLAT de 6 entradas (341) (tabela separada de consulta da lista). Assim de todas as transferências reais/virtuais I/O ou de processador resulta um endereço "real" formado na linha omnibus STC (157) pela unidade (153). Tipicamente, quando o adaptador de linhas omnibus (154) ou o processador S/370 (85) conduzem operações de memória "real", a unidade (153) simplesmente actua como um estágio de transição da linha omnibus do processador (170) para a linha omnibus STC (157), excepto para as respostas positivas da memória "cache", que podem ter como resultado o cancelamento de um comando depois de ter sido formado na linha omnibus STC (157).

Apresenta-se agora uma breve descrição das 41 linhas omnibus STC (fig. 32A e (30)). A linha omnibus STC de dados/endereços/comandos (406) tem 32 linhas omnibus de dados bidireccionais, mais paridade ímpar, por octeto. Esta linha omnibus é usada para transportar comandos e endereços num ciclo e até 32 bits de dados em cada ciclo subsequente da operação de memória. A linha STC VALID é activada pela unidade (153) para a interface STCI (155) para sinalizar que um comando/endereço é válido na linha omnibus STC no mesmo ciclo. A linha STC CANCEL é activada pela unidade (153) para a interface STCI (155) para cancelar um comando anteriormente emitido. Pode aparecer até 2 ciclos depois de STC VALID ser emitido. É combinada logicamente por um circuito OU com a entrada COMMAND CANCEL da PE (85). A linha STC BUSY (440) é activada pela interface STCI (155) para a unidade (153), um ciclo depois de ter sido emitido um "STC VALID", para significar que a unidade está ocupada e não pode aceitar um novo comando. É libertada 1 ciclo antes de a unidade (155) estar em condições de receber um novo comando. STC DATA INVALID na linha (433) pode ser emitido pela interface STCI (155) para a unidade (153) no mesmo ciclo em que são devolvidos dados numa busca para invalidar a transferência de dados. A unidade (153) ignora o ciclo de dados se a linha não estiver activada. Esta linha será activada coincidentemente com dados quando tiver ocorrido um erro FAST ECC na linha omnibus (30), quan-



do tiver havido uma comparação com resultado negativo de dados entre a lógica de unidades emparelhadas STCI (155,155a) ou se se tiver detectado uma paridade incorrecta durante o ciclo de leitura da linha omnibus (30).

A linha STC DATA TRANSFER (441) é excitada para a unidade (153) pela interface STCI (155), para sinalizar uma transferência de dados na linha omnibus STC (157) no ciclo seguinte. Para as memorizações, ela dita que a unidade (153) forneça a palavra de 32 bits seguinte no ciclo seguinte. Para as buscas, ela alerta a unidade (153) de que o ciclo seguinte conterá dados válidos, a menos que seja ultrapassada por STC DATA INVALID no ciclo seguinte. A concepção da interface STCI (155) é completamente canalizada para permitir que todos os estados anteriores estejam activos no mesmo instante, no interior de uma CPU S/370. Deste modo, admitindo aceitações contínuas pela linha omnibus e a não existência de quaisquer erros, a interface STCI (155) pode manter dados canalizados em buscas, sem estados de espera, utilizando leituras de 64 bits (por ciclo da linha omnibus do sistema (30) de 125 ns) na linha omnibus STC (157) de 32 bits, 62,5 ns.

A interface (410) do sistema S/88 é usada na interface STCI (155) para permitir o acesso aos registados DIMENSÃO DA MEMÓRIA/BASE DA MEMÓRIA (405) e (404) no interior do espaço de memória de endereços virtuais local BCU. Os erros "BROKEN" (403) e "BUS INTERRUPT REQ"(IRQ) são fundidos com os do cartão (102) do processador S/88 para activar uma interrupção de baixa prioridade na linha omnibus (30) como uma CPU única.

Os erros "Bus IRQ" diferem dos erros "Broken" por esses erros, usualmente devidos a sinais não protegidos provenientes da linha omnibus (30), que são detectados diferentes pelo mesmo cartão ou pelo cartão parceiro, não desligam um cartão da linha omnibus (30), como o fazem os erros "Broken". Estes erros só são activados quando o cartão está no modo "Obey both".

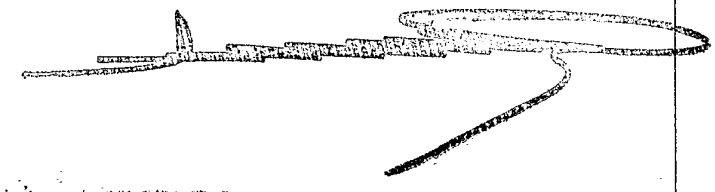
Além disso, são activados a partir da lógica (415) do cartão do processador S/88 os sinais "Obey A", "Obey B" e "Duplexed" pelas linhas (411), (412) e (413), em vez



de os realizar de novo no interior dos processadores S/370. Os sinais ObeyA/ObeyB são usados para controlar os multiplexadores de entrada (71) e (73) (fig. 8) para respectivamente, os multiplexadores de verificação e de entrada de dados do lado da activação, bem como para introduzir condições de erro nas linhas omnibus. O sinal Duplexed na linha (412) é usado para sinalizar quando os cartões estão emparelhados (isto é, é usado na lógica (408) de arbitragem das linhas omnibus para garantir que os dois parceiros são arbitrados conjuntamente quando em ranhuras consecutivas).

Os sinais Obey A e Obey B são invertidos para proporcionar +Obey A, -Obey A, +Obey B e -Obey B. Os sinais +Obey A e -Obey B são aplicados aos registadores (428) e (429), respectivamente. Os registadores (428) e (429) são acoplados respectivamente às linhas omnibus A e B da estrutura de linhas omnibus (30), respectivamente. Os sinais de relógio S/88 (não representados) transmitem cadenciadamente dados das linhas omnibus A e B para os registadores (428) e (429) respectivamente para todos os três modos de relógio A, B e Ambos. Os dados no registador (428) saem para as linhas omnibus (436) e (435) quando a linha omnibus está a funcionar nos modos Obey A ou Obey Both e o registador (429) dá na sua saída sinais para as linhas omnibus (435) e (436) apenas durante o modo Obey B. Analogamente, como se vê na fig. 34, o conteúdo do registador (428a) da interface STCI (155a) é igualmente fornecido na saída durante os modos Obey B e Obey Both. O conteúdo do registador (429a) é fornecido na saída durante o modo Obey A. Combinando por um circuito OU as saídas dos registadores (428) e (429) e (428a) e (429a) realizam-se as funções dos multiplexadores de entrada respectivos.

Os valores Dimensão da memória/Base da memória nos registadores (405) e (404) são colocados num mapa da memória no espaço de endereços virtuais do processador S/88 (62), através do espaço de endereços locais da BCU. Eles têm de ser colocados durante o processo de inicialização do S/88, uma vez definido o espaço (162) da CPU S/370 dado. Podem ser altera dos pelo S/88 desde que não estejam em processamento operações de memorização/busca na interface STCI.



O acesso aos registadores (404) e (405) é obtido pela lógica de descodificação de endereços (216) da fig. 19A, através de um endereço local (007E01FC) e incluem os dados seguintes: bits PA 20-23 e bits PA 20-27, que são iguais respectivamente à dimensão da memória (MEM SIZE) S/370 (162) e ao endereço da base da memória (MEM BASE), sendo:

MEM Size = mega-octetos (1 a 16) da memória principal atribuídos pela memória S/88 (16) à área de memória (162).

MEM Base = mega-octetos de desvio do endereço zero no espaço de endereços físicos na memória (16) atribuídos à área de memória (162).

PA = S/88 endereço virtual traduzido (isto é, endereço físico).

Quando a lógica (216) descodifica o endereço 007E01FC, são colocados os bits da dimensão e do endereço da base nos registadores (405,404) pelo processador (62), através da sua linha omnibus (161D). Durante esta operação, a lógica (216) desacopla o processador (62) do seu equipamento físico associado, de modo que a carga dos registadores (404) e (405) é transparente para o sistema operativo S/88. Além disso, o sistema operativo S/370 desconhece a sua existência ou o seu uso no acesso à memória S/370 (162).

As fig. 32A e B também ilustram as linhas de entrada/saída (I/O) usadas pela interface de controlo da memória (155). Esta inclui além da linha omnibus STC (157) todas as linhas necessárias para a interface para a linha omnibus (30) do sistema S/88, para o processador S/88 (62) e para a lógica (415) no cartão (102) da CPU S/88. Para facilitar a descrição, não se representam nas fig. 32 A e B os emissores-receptores (13) da fig. 8.

d) Operações de memorização de dados

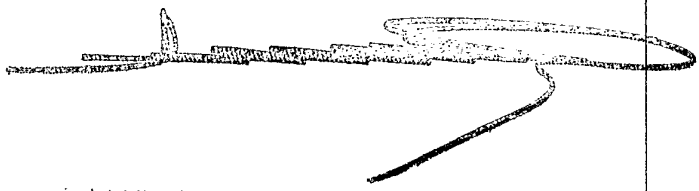
Quando vem um comando de memória da unidade do controlador da memória "cache" (153), a interface STCI (155) introduz, com um impulso de relógio, o comando na linha



omnibus de endereços/dados (406) (que faz parte da linha omnibus STC (157)) bits 0-7 e armazena-os na memória tampão de comando (CMD BUFF) (416) juntamente com o bit STC VALID e na memória tampão (417). O sinal STC BUSY na linha (440) será activado durante o ciclo seguinte pela lógica FIFO/STC.BUS (401), para indicar que a unidade (155) está ocupada. Entretanto o endereço real de 24 bits na linha (406) é também introduzido por um impulso de relógio no registador A/D (417).

Sempre que a memória FIFO (400) não esteja cheia e possa aceitar todo o comprimento da transferência de dados (até 64 octetos) especificado no comando (não existência de transbordo na memória FIFO), será activado o sinal STC DATA TRANSFER pela lógica (401), mantendo-se activo em todos os ciclos até estarem completas todas as transferências de dados da linha omnibus STC para este comando. Nas memórias, o sinal STC DATA TRANSFER não é enviado (e portanto o comando não é deslocado para a memória FIFO) até ser assegurado que não foi emitido nenhum cancelamento (até 2 ciclos depois de STC VALID). Porém, durante este tempo, a lógica (401) desloca o endereço de 24 bits do registador (417) para o registador (442) e os quatro primeiros octetos de dados são transferidos da unidade (153) para o registador (417). Além disso, os indicadores FIFO HF e AFE (409) são comparados com o comprimento descodificado do comprimento da transferência de octetos proveniente da memória tampão de comando CMD BUFF (416). Os indicadores FIFO indicam 1 de 4 valores da profundidade da memória tampão em uso. Se o comprimento da transferência de octetos, mais os 4 octetos dos dados da palavra de comando, excederem a capacidade de 64 palavras da memória FIFO, então são retidas todas as activações de STC DATA TRANSFER até desaparecer esta condição de transbordo. Isso verificar-se-á logo que se tenham deslocado para fora da memória FIFO palavras suficientes para modificar o estado dos indicadores.

Se não se verificar qualquer cancelamento e não existir transbordo na memória FIFO, as descodificações do comando proveniente do bloco (401), concatenadas com o endereço de 24 bits proveniente do registador (442), através do multiplexador (447), são armazenadas na memória FIFO (400) em ci-



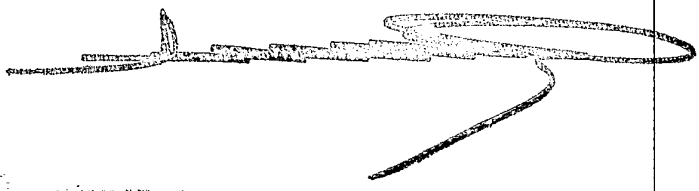
culos consecutivos, através do registador (442), uma vez que o comando de memória inicial seja deslocado para o interior da memória FIFO. A porta (423) é usada para multiplexar os 16 bits inferiores nos 16 bits superiores para transferências de 16 bits na linha omnibus (30).

O bit S é usado para distinguir armazenamentos de buscas e o bit C/A é usado para diferenciar entre palavras de comando e palavras de dados na memória FIFO (400), como se vê na fig. 35. A paridade é mantida através da memória FIFO.

As entradas e as saídas da memória FIFO são comandadas por impulsos de relógio de maneiras diferentes. Os dados são deslocados para o interior da memória FIFO (400) com impulsos de relógio S/370, enquanto que são deslocados para fora da memória com impulsos de relógio S/88. As distribuições de tempos são ajustadas para considerar o caso mais desfavorável dos tempos de passagem pelas memórias FIFO (60 ns), quando a memória FIFO (400) está vazia. As palavras de comandos e de dados da memória FIFO estão representadas na fig. 35, onde:

- S = (1 = Armazenamento, 0 Busca)
- C/A = (1 = Comando/endereço, 0 = dados)
- P01 = Octetos 0,1 Paridade par
- P23 = Octetos 2,3 Paridade par
- LWD = Seleção da palavra de dados inferior (LWD) (palavra de dados inferior multiplexada na palavra superior; P01 = P23 neste caso)
- 64 OVFL = excedida a transferência de 16 palavras devido ao alinhamento irregular do endereço; exige um ciclo adicional de dados de 32 bits.
- 32B16B,8B,4B = contagem ponderada das transferências de octetos
- TRL1,0 = Codificação para octetos válidos na palavra "traseira" (transferência dos últimos dados de 32 bits).

Dispositivos sequenciadores individuais no bloco (401), nos lados de entrada/saída da memória FIFO (400), seguem as transferências para/da memória FIFO. O sequenciador de saída segue o número de transferências de dados na linha om-



nibus (30) pendentes para o comando corrente de busca ou de armazenamento. Uma vez que a palavra de comando atinga a saída da memória FIFO, o bit C/A = 1 é descodificado na lógica (401); e, desde que não haja qualquer comando anterior pendente para se completar, funde-se o endereço real S/370 proveniente da memória FIFO (400) com o registador da base (404), através da lógica (422) e (423), que é então carregado como endereço "físico" inicial na memória de endereços (420), enquanto se carrega a contagem de transferências no sequenciador de saída em (401). Também a lógica de arbitragem (408) é ajustada para começo da arbitragem.

A lógica de controlo de ciclos em (408) seguirá todas as fases activas da interface STCI (155) na linha omnibus (30) para as operações quer de busca, quer de memorização. Juntamente com as linhas de estados da linha omnibus (30) (isto é, BUS BUSY (linha omnibus ocupada; Bus Error, erro na linha omnibus), esta lógica é usada no interior da interface STCI (155) para tratar as operações das fases normais na linha omnibus (30), bem como para o tratamento de condições de erro que conduzem a definição ou fases de dados dos ciclos cancelados.

O endereço físico é formado comparando em primeiro lugar na lógica (442) os quatro bits superiores do endereço real de 24 bits S/370 proveniente da memória FIFO (400) com o valor da dimensão da memória S/370 no registador (405). Se os bits do endereço S/370 não excederem a zona da dimensão atribuída ao processador S/370 (85), os quatro bits superiores são então adicionados pela lógica (423) ao valor de base da memória S/370 no registador (404), e concatenados com os bits inferiores 19-1 na memória tampão (420), para formar um endereço físico de 27 bits que é usado como endereço S/88 inicial na área S/370 (162). Por outro lado é anunciado uma verificação do programa. Qualquer atravessamento dos limites dos endereços de 64 octetos terá como consequência um regresso ao endereço inicial.

O contador U/D de endereços (421) é usado para reter os bits 5-2 do endereço físico de saída. Ele recebe impulsos de relógio, em sincronismo com o sequenciador de saída e, embora normalmente incrementado, ele pode ser decrementado quando responde a condições BUS BUSY e BUS ERROR de um ci-



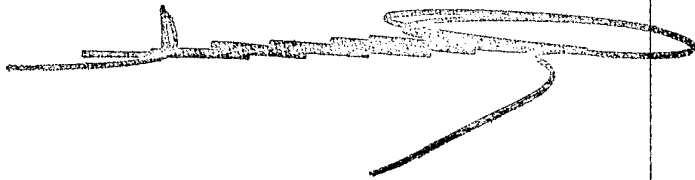
clo da fase de resposta. Uma vez carregado o sequenciador de saída, a lógica associada inicia ciclos de armazenamento, com base em aceitações de arbitragem da linha omnibus através da lógica (408), enquanto que responde às condições BUS ERROR e BUS BUSY. Produz-se um código apropriado de uma função S/88 pela lógica (401) correspondente ao comando de armazenamento S/88; e o código da função é colocado no registador (443) para aplicação nas linhas omnibus A e B da estrutura (30) de linhas omnibus, quando um ciclo de arbitragem pedido for atendido.

O sequenciador de saída é normalmente decrementado de cada vez que é atendido um pedido, sendo o decremento de 1 para transferências de 32 bits e de 2 para transferências de 64 bits para a linha omnibus (30), até se atingir o zero, indicando que não há mais octetos para ser transmitidos para o comando presente.

No caso de uma condição de BUS BUSY ou de BUS ERROR durante uma fase de resposta do ciclo sobreposta com uma fase de definição do ciclo o sequenciador de saída será incrementado de 1 para as transferências de 32 bits e de 2 para as transferências de 64 bits (apenas busca) canceladas.

Simultaneamente, o contador U/D de endereços (421) é decrementado de 1 para transferências de 32 bits e de 2 para transferências de 64 bits (apenas buscas).

O registador de saída de dados (425) é usado para reter temporariamente dados de saída. O registador de saída de dados (426) é necessário no caso de ser necessário reactivar os dados, devido a um Bus Error subsequente (erro na linha omnibus A ou B). Neste caso, os dados subsequentes (para um endereço mais elevado) podem ser aceites e armazenados na memória (16,18) mais cedo que os dados do ciclo anterior, que estão associados com o Bus Error, porque a transferência de dados ter de ser pedida 2 ciclos depois da sua transferência inicial. (Em contraste com os dados de armazenamento, os dados de busca não podem ser recebidos fora da sequência). Entretanto, a lógica da arbitragem da linha omnibus (408) arbitra continuamente ciclos até serem iniciadas as transferências e aceites na linha omnibus (30). A arbitragem e a transferência de dados para a linha omnibus do sistema (30) e para a memória (16,18) são seme-



lhantes às descritas anteriormente na secção b).

Finalmente, note-se que a concepção da memória FIFO permite o armazenamento até 64 palavras (quase 4 grupos de transferências de memória de 64 octetos) antes se atingir o estado de ocupado. Para os armazenamentos, enquanto a memória FIFO não estiver cheia e possa aceitar as palavras de comandos e de dados associados com a memorização, a memória FIFO é carregada continuamente até ser possível. Por conseguinte, desactiva STC Busy depois de se executar cada comando de armazenamento, libertando-se a unidade (153) e permitindo que o processador S/370 (85) continue a execução. Admitindo que há uma elevada percentagem de resultados positivos na memória "cache" na unidade (153), aumenta-se de maneira significativa a eficácia memorizando na memória tampão o equivalente a um armazenamento de quase 64 octetos na memória FIFO ou armazenamentos de 32 octetos 1-4.

Parte-se da hipótese de que a interface STCI (155) é o lado de "excitação" do par (155,155a) de interfaces STCI e que a interface STCI (155a) é o lado de "verificação" de erros. Portanto, apenas a interface STCI (155) activa sinais (controlo, endereço, dados) na estrutura de linha omnibus (30), como se mostra na fig. 32B. Quando os sinais se destinam para as duas linhas omnibus A e B, as linhas de activação da interface STCI (155) estão representadas acopladas às duas linhas omnibus (através dos emissores-receptores (13), não representados na fig. 32 B). Na interface STCI (155a), as linhas correspondentes não estão acopladas à estrutura de linhas omnibus (30); apenas à lógica de comparação (402-g).

A lógica de comparação (402g) compara os bits 27-6 do endereço provenientes da memória tampão (420), os bits de endereço 5-2 provenientes do contador U/D de endereços (421), o bit de endereço modificado 1 e o bit de paridade proveniente da lógica geradora de paridade (PTY Gen) (445), e o código da função proveniente do registador (443), com os bits correspondentes provenientes da interface STCI (155a). No caso de uma comparação com resultado negativo, a lógica (402g) aplica sinais de erro na lógica de interrupções (403) e às linhas BUS ERROR A e B.



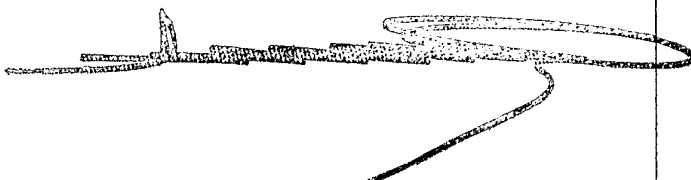
A lógica (402a) compara os bits de saída dos dados que saem do registador (425) com os bits correspondentes provenientes da interface STCI (155a) e aplica os sinais de comparação negativa à lógica (403) e às linhas BUS ERROR A e B. A lógica (402d) compara bits provenientes da lógica (401) da memória FIFO com bits correspondentes da interface STCI (155a). A porta E (446) proporciona um sinal de erro para a lógica (403) se estiver activado o sinal STC VALID enquanto o sinal STC está activado na linha (440).

e) Operações de busca de dados

Um comando de busca segue o mesmo trajecto que os comandos de memorização, através dos registadores (416), (417), (442) e da memória FIFO (400), atrás descrito. Uma diferença é que o sinal STC DATA TRANSFER não é activado na lógica STC BUS (408) até saber-se que os dados foram recebidos no registador (428) ou (429), a partir da memória (162), através da linha omnibus (30). Um comando de busca e um sinal STC VALID são recebidos e armazenados no registador (417). A lógica do STC BUS em (401) emite um sinal STC BUSY durante o ciclo STC BUS seguinte para impedir que o controlador da memória "cache" emita outro comando até ser removido STC BUSY.

Assim, quando for possível ter recebido um comando de busca, o sinal STC BUSY é mantido pela lógica (401) até ter sido completamente executado o comando de busca, visto que o controlador da memória "cache" (153) está à espera dos dados para a busca a receber. (Durante os ciclos de memorização, STC BUSY foi removido, logo que se tenham transferido os dados de armazenamento do controlador (153) para a memória FIFO (400). Durante o ciclo do comando de busca, o sinal STC BUSY tem de ser mantido até que todo e qualquer comando de armazenamento na memória FIFO (400) seja executado, executando-se depois o comando de busca. Só então pode o sinal STC BUSY ser removido para permitir a transferência do comando seguinte para a interface STCI (155).

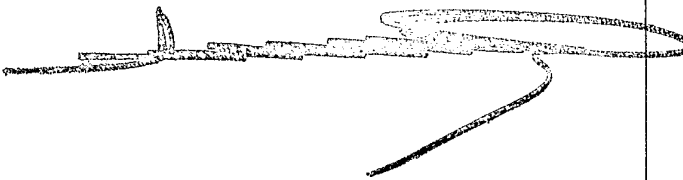
Nos ciclos a seguir ao armazenamento do



comando nos registradores (416) e (417), o comando e o endereço são transferidos para o registrador (442) e depois para a memória FIFO (400).

Quando o comando de busca for recebido no último andar da memória FIFO (400) (quando o sinal OUTPUT-Ready tiver o nível elevado, como atrás se descreveu), o bit C/A e outros bits do comando são decodificados, na lógica (401). Um código de função S/88 correspondente aos bits de comando S/370 decodificados é colocado no registrador (443), para aplicação à estrutura de linhas omnibus (30), quando tiver sido satisfeito um pedido de ciclo de arbitragem.

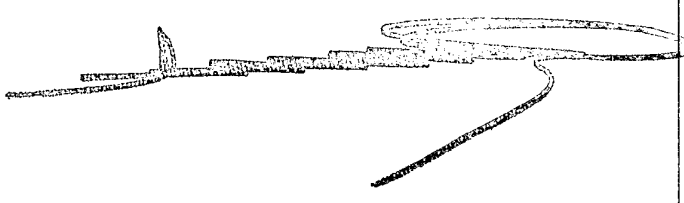
A seguir a essa aceitação e à fase subsequente de definição do ciclo e à fase de resposta do ciclo, a interface STCI (155) introduzirá a fase de dados, admitindo que não foi comunicada nenhuma situação de ocupação da linha omnibus, ou de erro na linha omnibus, durante a fase de resposta do ciclo. Os primeiros 32 bits, juntamente com os bits DP, UDS e LDS, são recebidos nas linhas omnibus A, B da estrutura de linhas omnibus (30) da posição apropriada na área (162) da memória (16) e da sua parceira, e retidos nos registradores (428, 429), respectivamente, iniciando o relógio S/88 a segunda metade do ciclo da linha omnibus (30). Admitindo que está activo o modo Obey Both ou o modo Obey A, os dados serão encaminhados do registrador (428) para a memória tampão (430) no ciclo seguinte do relógio S/88 (início do ciclo da linha omnibus (30) seguinte). Para as transferências de 64 bits, os segundos 32 bits são retidos em registradores (428) e (429), concorrentemente com a transferência dos dados anteriores para a memória tampão (430). Um gerador de paridade (431) adiciona paridade ímpar à palavra de dados armazenada em (430). Estes dados e os bits de paridade, juntamente com os bits UDS, LDS e DP recebidos, são aplicados à lógica (402c), através das linhas omnibus (435) e (436). A lógica (402c) compara estes bits com os bits correspondentes produzidos na interface STCI emparelhados (155a). A memória (430) encaminhará agora a primeira palavra de dados, mais a paridade, para a memória tampão (432) para ser activada durante o ciclo STC BUS seguinte, para a transferência para o controlador da me



mória "cache" (153), através de uma linha omnibus (406) da linha omnibus STC (157). A memória tampão (432) é comandada com impulsos de relógio do relógio S/370, que são sincronizados com impulsos de relógio S/88 de modo tal que o início do ciclo da linha omnibus STC ocorre depois da activação do relógio S/88. Como são definidos períodos de 62,5 ns idênticos para os dois relógios S/88 e S/370, isso permite a canalização das leituras consecutivas da linha omnibus (30) para a linha omnibus STC. Assim, na forma de realização preferida, executam-se dois ciclos da interface STCI (155) por cada ciclo de 125 ns da linha omnibus (30).

Admitindo que há aceitações sucessivas para a interface STCI (155), seguir-se-á uma segunda fase de dados à primeira fase de dados atrás descrita (admitindo que não há erros da linha omnibus, etc.). Admitindo que há transferências de 64 bits, os dados serão então introduzidos por impulsos de relógio nos registadores (428) e (429), simultaneamente com os dados introduzidos por impulsos de relógio da memória tampão (428) (ou (429) para o modo Obey B) para a memória tampão (430). Os dados da memória tampão (430) aplicarão então os 32 bits seguintes à memória tampão (432) para transferência para o controlador da memória "cache" (153), como atrás se descreveu. Vê-se portanto como podem utilizar-se transferências de 64 bits consecutivas para manter um fluxo de dados canalizado na forma de realização preferida.

Se durante esta fase ocorrer um erro FAST ECC, ou uma comparação de dados com resultado negativo, ou um erro de paridade, é emitido o sinal STC DATA INVALID, pela linha (433), pela lógica (402c) simultaneamente com os dados na linha omnibus de endereços/dados STC (406). Além disso, se chegarem dados subsequentes no ciclo depois do ciclo no qual houve dados invalidados, será forçada uma condição de erro na linha omnibus pela lógica STCI SBI, em ambas as linhas omnibus A e B, a seguir a esse ciclo de dados. Isso garante que os dados serão de novo activados 2 ciclos mais tarde (isto é, um ciclo depois de ser anunciado o erro na linha omnibus), mantendo-se assim a integridade e a funcionalidade dos dados na linha omnibus STC, transferindo dados buscados em sequência. A activação de erros



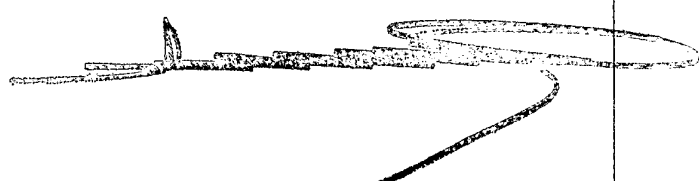
da linha omnibus quer na linha omnibus A quer na B é equivalente à memória (16) comunicar uma condição de erro ECC em relação com um erro "verdadeiro" na linha omnibus, não provocando assim uma alteração na lógica OBEY das linhas omnibus ao longo de todos os controladores nas linhas omnibus do sistema (30).

Analogamente, utiliza-se a mesma lógica (402c) para comparar os dados recebidos e para verificar a paridade através das linhas omnibus (435) e (436), nas operações de memorização, para verificar os resultados da comparação da saída de dados em (402e), efectuando uma comparação de dados "em anel fechado" ("loopback") da linha omnibus do sistema (30) através dos registadores (428) e (429). Isso ajuda a identificar os problemas nos emissores-receptores (13), no cartão (101), mais rapidamente e ajustará o cartão da lógica de interrupções (403) nas memórias, se houver uma comparação com resultado negativo e não for comunicado um erro da linha omnibus no ciclo seguinte da linha omnibus. Além disso, todas as saídas dos comparadores (402 a-g) que produzem uma condição de avaria em comparações com resultado negativo, válidas para operações de busca e de armazenamento, gerarão uma condição de interrupção na lógica (403). O ajustamento inicial de interrupção gerará sinais de erro nas linhas omnibus, quer na linha omnibus A, quer na B, assegurando assim que é repetida uma transferência de dados no ciclo anterior, enquanto que qualquer fase de definição do ciclo no ciclo anterior é abortada.

Em contraste com o armazenamento, para as buscas todos os comandos anteriormente na memória FIFO têm de ser executados, bem como a busca corrente, antes de a unidade poder libertar a linha STC BUSY (440) e aceitar outro comando. O controlador da memória "cache" (153) tem de receber os dados para um comando de busca antes de poder ser emitido um outro comando de memória.

A definição dos tipos de ciclo de leitura/escrita disponíveis está representado nas fig. 36 A-D, nas quais:

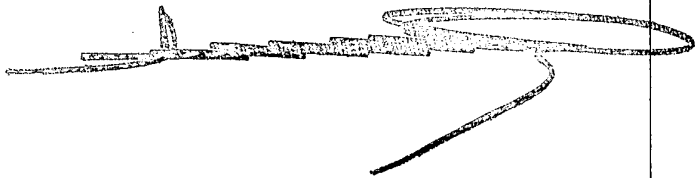
- UU = Octeto superior da palavra superior
- Um = Octeto superior da palavra média
- LM = Octeto inferior de palavra média



LL = Octeto inferior de palavra inferior
MEM 16 = Ciclo de memória de 16 bits
MEM 32 = Ciclo de memória de 32 bits
MEM 64 = Ciclo de memória de 64 bits
LW = Palavra comprida (32 bits)
UDS = Impulsos sincronização de dados superiores
LDS = Impulsos de sincronização de dados inferiores

Não se dispõe de escritas de 64 bits na forma de realização preferida da unidade (155), devido à ênfase posta na minimização dos meios físicos. Uma memória FIFO 64 x 36 é suficiente para suportar a transferência de memória de 32 bits do S/370. Uma limitação da eficácia resultante de usar apenas escritas de 32 bits é que, visto que cada "folha" do cartão de memória S/88 na memória intercalada (16) tem o comprimento de 72 bits (64 bits mais 8 bits ECC), cada folha, uma vez tendo sido acedida para as escritas, manter-se-á ocupada durante e ci cl os adicionais (125 ns). Isso significa que a mesma folha pode ser acedida apenas uma vez cada 5 ciclos (625 ns) em escritas consecutivas. Como todas as escritas de 32 bits S/370 são definidas para endereços sucessivos, isso significa que transferências consecutivas no interior do mesmo limite de 64 bits não po de m ser emitidas mais rapidamente que de 5 em 5 ciclos (625 ns) enquanto que transferências em limites de 64 bits podem ser emi ti da s em ciclos sucessivos de 125 ns (admitindo que a arbitragem foi ganha).

Os ciclos de leitura de 64 bits são admitidos e, neste caso, enquanto as leituras consecutivas não te na m acesso à mesma folha, podem ser executadas em ciclos con se cu t i va s. Caso contrário podem ser executadas de 2 em 2 ciclos (250 ns). Devido ao facto de cada 32 bits serem recebidos da li nh a omnibus (30) em leituras de 64 bits de 62,5 e 62,5 ns (por exemplo, duas vezes todos os 125 ns do ciclo da linha omnibus (30)), os tempos dos ciclos das linhas omnibus STC e (30) estão adaptados de modo tal que os dados podem ser canalizados da li nh a omnibus do sistema (30) para a linha omnibus STC (157) depois de ter sido recebidos. Utilizam-se dois níveis extraordiná ri os de memorização em memórias tampão (memórias tampão (430) e (432)) com registadores (428) e (429), para suportar sincroniza-



ção apropriada de ciclos e permitir a geração de paridade de cada octeto de dados.

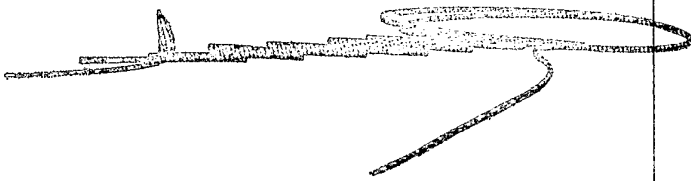
Cada endereço de 27 bits e cada código de função de 4 bits são enviados, em conjunto com um bit de paridade, que os acompanha durante as fases de definição de ciclos da linha omnibus (30). Os dados de 32 bits levam também um bit de paridade, com eles associado, durante as fases de dados da linha omnibus (30). Um ciclo básico de 125 ns na linha omnibus (30) permite transferências normais de 16 e 32 bits, bem como transferências de leitura de 64 bits dentro da janela de 125 ns. Optativamente, pode usar-se equipamento físico adicional para aceitar transferências de escrita de 64 bits na interface STCI (155).

Suporte de I/O S/370 (fig. 37)

A fig. 37 ilustra esquematicamente uma vista geral dos equipamentos físicos e do código de aplicação S/88 que é utilizado para suportar as funções de I/O S/370. Os dispositivos físicos são (601), (602), (615) a (619), (621) e (623) a (625). As rotinas em meios de programação (ou em meios de microcódigos) são (603) a (614), (620), (622) e (626).

As funções destes vários elementos vão agora descrever-se. O bloco (606) é o controlo principal para o código de aplicação S/88 que consiste nos blocos (606) a (614). Este conjunto de blocos, conhecido por EXEC370, efectua todas as funções do código de aplicação S/88 para a emulsão e o suporte dos dispositivos, serviços, configuração e consola do operador, etc. S/370.

O bloco (603) é o microcódigo que é executado no microprocessador S/370. Um protocolo entre o bloco (603) e o bloco (606) permite que eles transmitam pedidos e respostas entre si, relativas à iniciação das operações I/O S/370, à sua terminação e à informação de estados do dispositivo e canais de I/O S/370. Ele permite também que o bloco (606) solicite o bloco (603) para efectuar funções CPU S/370 específicas. O bloco (605) é a memória S/370 e é acessível directamente quer



para o bloco (603) quer para o bloco (606). O bloco (606) proporciona a configuração S/370 apropriada através dos dados contidos no bloco (602), que é um ficheiro de dados S/88.

O bloco (604) é uma tarefa, executada separadamente, que proporciona o painel do operador S/370 através do dispositivo terminal S/88. Esta tarefa pode ser iniciada ou parada em qualquer instante sem cortar o funcionamento lógico do processo S/370. O bloco (607) é uma parte do EXEC370 e proporciona a emulação da interface entre o processo S/370 e o bloco (604).

O bloco (601) é um conjunto de "ficheiros de correcções" ("patch files") que contêm o código objecto S/370 que foi escrito especialmente para fins de detecção e correcção de erros ("debugging") de programas S/370 incluindo a sua BCU (156). Há um painel de detecção e correcção de erros de programas proporcionando no bloco (604), que permite a selecção e a carga no bloco (605) de um destes "ficheiros de correcção".

O bloco (608-1) consiste no código responsável pela emulação do canal S/370. Ele executa a busca de palavras de comandos S/370, o movimento de dados para e do bloco (605), a comunicação de informação de interrupções I/O S/370 ao bloco (603) e a selecção do emulador de código da unidade de controlo apropriado. Pode haver mais de um canal S/370 (por exemplo (608-2)), mas é usado o mesmo código.

O bloco (609-1) é o código emulador da unidade de controlo S/370. O sistema S/370 tem muitos tipos diferentes de unidades de controlo, isto é, controladores DASD, controladores de fita, controladores de comunicações, etc.. A função dos controladores S/370 está distribuída entre o bloco (609-1) e o emulador dos dispositivos especiais, o bloco (610) a (614). A finalidade principal do bloco (609-1) é o das funções de separação de endereços, mas outras funções específicas da unidade de controlo podem residir no bloco (609-1). Há portanto mais de um bloco deste tipo (por exemplo o bloco (609-2)), isto é, o emulador do controlador DASD, o emulador do controlador de comunicações, etc.; mas não há uma correspondência de um para um com as unidades de controlo S/370 suportadas.

O bloco (610) representa o código neces-



sário para emular uma consola S/370. O bloco (611) representa o código necessário para emular um terminal S/370. O bloco (612) representa o código necessário para emular um leitor S/370. Este é um dispositivo de entrada virtual, modelado de acordo com o leitor VM normalizado. Ele proporciona a entrada de ficheiros sequenciais que foram gerados a partir de uma outra fonte, tipicamente uma fita ou uma disquete.

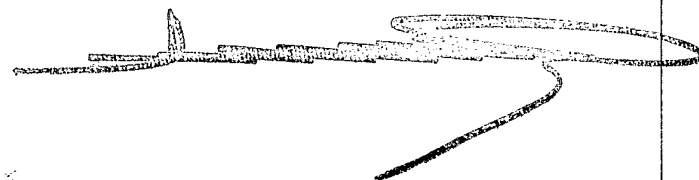
O bloco (613) representa o código necessário para a emulação de um impressor S/370. Um impressor S/88 actual pode ser accionado, ou os dados S/370 podem ser escritos num ficheiro S/88, para impressão da bobina, mais tarde. O bloco (614) representa o código necessário para emular um disco S/370 (DASD): admitem-se os dois formatos: Contagem, Chave e Dados; e Bloco fixo, por dois conjuntos diferentes de código.

O bloco (615) representa um terminal S/88, tipicamente o dispositivo de saída da consola S/88. A consola do sistema S/88 visualiza, quer mensagens de operador S/88, quer mensagens de operador S/370, além de registar as mensagens para um registo em disco, que aparecerá ao S/370 como um terminal 3278 ou 3279.

O bloco (616) representa um terminal S/88. O bloco (617) representa um ficheiro de dados sequencial S/88, num disco S/88. O bloco (618) representa um impressor S/88 ou um ficheiro de dados S/88 sequencial num disco S/88. O bloco (619) representa um ficheiro de dados S/88 num disco S/88. O bloco (620) é o código que lerá uma fita do sistema S/370 montada num dispositivo de fita S/88 e que a apresentará no formato de um bloco (617) quando ele aparecer na fita S/370 original. O bloco (621) representa um dispositivo de accionamento de fita S/88 no qual está montada uma fita escrita em S/370.

O bloco (622) é o código que lerá um ficheiro introduzido no S/88 a partir de um computador pessoal e o apresenta num formato do bloco (617) quando ele apareceu originalmente ao ser gerado num sistema S/370.

O bloco (623) é um computador pessoal (PCTERM) configurado para emitir e receber dados, quer do sistema S/88, quer do S/370. O bloco (624) é um sistema S/370. O bloco (625) representa um impressor com enrolador S/88. O bloco



(626) é o código que elabora o formato de um ficheiro S/88 para um dispositivo DASD S/370 emulado. Esta é uma tarefa executada separadamente no S/88 que elaborará o formato do ficheiro para qualquer dos DASD S/370 suportados desejados.

OPERAÇÕES I/O S/370. VISTA GERAL DOS MEIOS DE MICROPROGRAMAS

Apresenta-se agora uma vista simplificada e generalizada da I/O do sistema S/370. A arquitectura S/370 proporciona vários tipos de instruções I/O, um esquema de códigos de condições (CC) susceptíveis de ser ensaiadas por programas e um mecanismo de interrupções de programas. Conceptualmente, uma instrução I/O é dirigida para um "Canal I/O", que dirige e controla o trabalho da operação I/O em paralelo com outro processamento CPU e comunica estados à CPU quando a instrução I/O é executada (através do código de condições), e/ou quando a operação I/O está completa (através da interrupção do programa).

As instruções S/370, os códigos de condições, as interrupções e os dispositivos I/O (DASD, fita, terminais, etc.) estão estreitamente architectados. Contudo, o canal I/O está architectado de maneira solta para proporcionar uma certa latitude de concepção, existindo muitas formas de realização prática diferentes.

A visão alargada do aperfeiçoamento do sistema S/370 tolerante das avarias é então uma CPU S/370 (conjunto de micropastilhas com meios de microprogramas, específicos para o cliente) e um "psendocanal I/O" que consiste em faixas de tempo de uma CPU S/88 e um sistema operativo (OS), com a adição de meios de microprogramas especiais e meios de programação ao nível da aplicação (EXEC370) que proporcionam quer a emulação dos dispositivos I/O S/370, quer o controlo global do complexo do sistema. A parte S/88 deste complexo proporciona uma CPU, um sistema operativo, dispositivos I/O, blocos de alimentação, linhas omnibus e memórias tolerantes das avarias; a CPU S/370 é feita tolerante das avarias mediante redundância dos meios físicos e a adição de lógica de comparação.

Os meios de microprogramas (isto é, mi-

crocódigos) de cliente incluem-se em dois grupos:

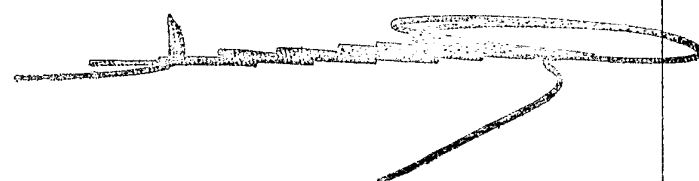
- a. Meios de microprogramas accionadores da BCU S/88 (ETIO) executados no processador S/88 (62) - rotinas de serviço para a inicialização e controlo do equipamento físico BCU/DMAC, serviço de interrupções DMAC e tratamento de estados e erros.
- b. Microcódigo S/370 (processador (85) - instruções I/O, tratamento de interrupções I/O e alguns controlos especiais tais como invocação de reposição, IPL, paragem, etc..

Como auxílio para compreender o contexto das várias operações de microprogramas, considere-se a seguinte sequência simplificada de eventos que ocorrem numa operação I/O típica: uma escrita S/370 de uma mensagem de 80 octetos para um terminal com visualizador 3278 S/370 emulado.

Suponhamos, para este exemplo, que a inicialização já foi feita, que o S/370 e o S/88 estão a funcionar normalmente e não está em curso nenhuma outra operação I/O S/370, sendo feita referência directamente à fig. 43 e às fig. 19 A-C. Cada uma das transferências de dados/comandos entre o PE (62) e os elementos da BCU (156) é efectuada usando o mecanismo "de desacoplamento" atrás descrito com referência à fig. 20. O fluxograma da fig. 43 ilustra esquematicamente esta operação I/O inicial típica.

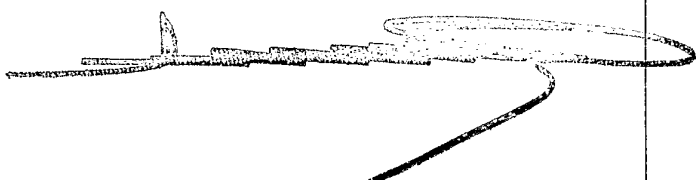
- a. O processador S/370 (85) encontra uma instrução START I/O (SIO) (todas as instruções I/O no conjunto de micropastilhas (150) são microcodificadas na forma de realização preferida).
- b. São invocados os meios de microprogramação para SIO: ele desloca vários parâmetros para a posição de caixa do correio fixa (188) (na área IOA da memória principal S/370), emite um pedido de serviço para a BCU 156 (pedido PU-BCU) e aguarda uma resposta.
- c. Os meios físicos da BCU detectam o pedido e geram um comando para a leitura da caixa do correio, de 16 octetos, a partir da posição fixa IOA S/370 e depois responde ao processador S/370 (85) repondo o pedido através de BCU-PU ACK (que significa "o pedido foi satisfeito").
- d. No processador S/370 (95), são libertados os meios de microprogramas SIO para terminar a instrução SIO e continuar o

- 
- processamento na instrução sequencial seguinte.
- e. Concorrente com o evento "d", como consequência de "c", os meios físicos S/370 estiveram a transferir os 16 octetos de dados da caixa do correio para a memória (259) da interface BCU no adaptador (154) através da linha omnibus (170).
 - f. Enquanto os dados são memorizados na memória tampão (em blocos de 4 octetos) os meios físicos BCU assinalam repetidamente o DMAC (209) (canal 0) para transferir os dados da caixa de correio (em blocos de 4 octetos) para um bloco FILA DE ESPERA DE TRABALHO na memória local (210).
 - g. Quando estiver completa a transferência de 16 octetos, o DMAC (209) apresenta uma interrupção (NOTIFICAR, fig. 43) ao processador S/88 (62) e prepara-se depois para uma operação de caixa de correio futura carregando o item seguinte na lista encadeada. Esta interrupção é uma das 8 interrupções DMAC para o processador (62), isto é, uma interrupção do canal 0 DMAC "normal".
 - h. Quando o S/88 aceitar a interrupção DMAC (sujeita a um possível retardamento, devido a mascaramento), executa-se uma rotina de serviço de meios de microprogramas de cliente (na ETIO); ela verifica o estado do DMAC (209), encontra o bloco da FILA DE ESPERA DE TRABALHO justamente recebido por referência à lista encadeada e introduz esse bloco na lista de espera para passar ao programa de aplicação EXEC370.
 - i. EXEC370 verifica a FILA DE ESPERA DE TRABALHO, retira da fila de espera o bloco da FILA DE ESPERA DE TRABALHO; constrói um pedido de dados no bloco FILA DE ESPERA DE TRABALHO e chama uma rotina dos meios de microprogramas para obter os 80 octetos de dados a enviar para o terminal 3278.
 - j. Os meios de microprogramas preparam e fazem arrancar o DMAC (209) (canal 1), depois enviam um comando para os meios físicos BCU para iniciar a leitura dos 80 octetos a partir de uma posição da memória S/370 específica através do adaptador (154), da linha omnibus (170) e do controlador da memória (155).
 - k. Os meios físicos BCU, o adaptador (154) e o DMAC (209) transferem os 80 octetos para o bloco FILA DE ESPERA DE TRABALHO e o DMAC (209) apresenta uma interrupção ao S/88; isto é se



melhante às operações nas alíneas f. e g. anteriores. Esta interrupção, uma interrupção "normal" do canal 1 do DMAC, é uma das oito interrupções DMAC atrás descritas.

- l. Uma rotina de serviço de interrupção dos meios de microprogramas mais uma vez verifica o estado do DMAC e introduz na fila de espera um indicador do bloco FILA DE ESPERA DE TRABALHO para EXEC370.
- m. EXEC370 faz qualquer conversão de dados necessária, depois escreve os dados para o terminal 3278 emulado usando os serviços do sistema operativo S/88. Passado um certo tempo, recebe a notificação do fim (normal ou de erro) dessa operação. Constroi então, no bloco FILA DE ESPERA DE TRABALHO, uma mensagem de interrupção S/370 apropriada, incluindo o estado, e chama de novo uma rotina dos meios de microprogramas para a escrever na fila de espera de mensagens S/370.
- n. Os meios de microprogramas preparam e fazem o arranque do DMAC (canal 3), depois emitem um comando para os meios físicos BCU para escrever 16 octetos para a fila de espera das mensagens S/370. Isto é análogo a uma leitura da mala de correio no sentido inverso, excepto que neste caso o adaptador (154) gera uma interrupção de excepção ao nível do microcódigo no processador S/370 (85), no fim da operação (também sujeita a um retardamento por mascaramento). O DMAC (209) também interrompe (NOTIFICAR, fig. 43) o processador S/88 (62), tal como em g. e k. anteriormente. Esta interrupção, uma interrupção "normal" do canal 3 do DMAC, é uma das oito interrupções DMAC.
- o. No processador S/370 (85), os meios de microprogramas de cliente tratam a excepção e têm de ensaiar as máscaras do canal para a possibilidade de retardamento; se estiverem mascarados, de modo que não pode apresentar-se uma interrupção no programa em execução, os dados essenciais são deslocados da área (189) da fila de espera de mensagens para uma fila de espera de interrupções pendentes; um outro dispositivo de tratamento dos meios de microprogramas de cliente atenderão a mesma quando o canal for de novo activado para interrupções. Se não estiver mascarado, estes meios de microprogramas comutam o contexto do S/370 para a rotina de



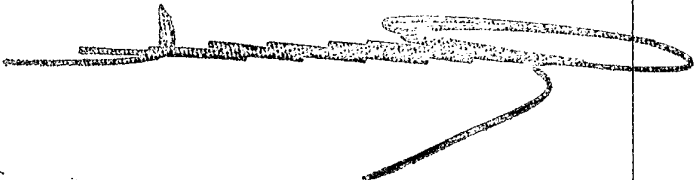
interrupção de programa, imediatamente.

Uma visão ampla do sistema FT aperfeiçoado conduz à conceptualização do papel do S/88 como um processador I/O dirigido anexo - é um processador I/O ou um pseudo-canal para S/370. Porém, actualmente toda a comunicação básica entre os processadores tem de ser iniciada a partir do S/88 (devendo à concepção). Também, o S/88 pode aceder a toda a memória e ao espaço de microcódigos S/370, através de EXEC370, ao passo que o inverso não é verdadeiro - o processador S/370 (85) não pode aceder de modo nenhum à memória S/88, nem mesmo acidentalmente. Assim, a imagem mais real é a do S/370 como dirigido pelo S/88, mas com a imagem interna de um S/370 independente, normal, com I/O S/370. O S/370 não "sabe" que o S/88 está ali.

Mas, como os programas S/370 são executados em sincronismo com o S/88 e não têm de ser impedidos, as instruções I/O S/370 têm de ser capazes de INICIAR uma acção, sendo esta facilidade proporcionada pela linha (256a) de pedido da PU-BCU, que tem um significado singular: S/370 tem uma espera de mensagens de alta prioridade para o S/88 (usualmente uma instrução I/O). A natureza da prioridade deste pedido de serviço é a razão para o esquema de caixa de correio automática e para a programação do canal 0 do DMAC numa lista encadeada.

O DMAC (209) é uma parte integrante da concepção dos meios físicos BCU. Ele é inicializado e controlado basicamente pelos meios de microprogramas S/88, e as transferências de dados são cedenciadas pela lógica BCU, que activa as quatro linhas (263-d) de entrada de pedido REQ, uma para cada canal. Além disso, a lógica BCU externa activa a linha (257a) CHANNEL 0 PCL, quando se completa cada transferência da caixa do correio, fazendo com que o DMAC (209) apresente um pedido de interrupção ao processador S/88 (62).

Há quatro operações básicas de transferência de dados entre S/370 e S/88:



	Dimensão	Adaptador DMAC (154) (209)	Canal	Canal	Tipo de operação DMAC
1. Ler caixa do correio	16 octetos	0	0		Contínua Lista encadeada
2. Ler dados	1-4096 octetos	0	1		Arranque-para- gem pré-esvazi- ável
3. Escrever dados	1-4096 octetos	1	2		Arranque-para- gem pré-esva- ziável
4. Escrever mensagem Q	16 octetos	1	3		Aranque-para- gem

A inicialização e a programação do DMAC (209) é completamente a normalizada e de preferência em conformidade com a arquitectura MC68450. Em resumo:

Todos os 4 canais - dimensão da transferência das palavras (16 bits); a linha REQ controla a transferência; o endereço da memória na memória (210) é contado para cima; o endereço do dispositivo (registador da memória de dados BCU) não é contado
interrupções activadas; roubo de ciclos sem retenção; dispositivo com confirmação/modo de endereçamento endereçado implicitamente/
/simples; porta de dispositivo de 16 bits;
PCL = entrada de estados

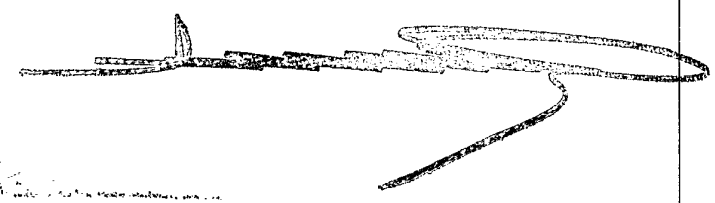
Adicionalmente ao anterior -

CHO Transferência do dispositivo para a memória (memória (210)); encadeamento de agregados ligados; PCL = entrada de estado com interrupção

CH1: transferência para a memória (memória (210)); ausência de encadeamento

CH2 e 3 transferência da memória (memória (210)) para o dispositivo; ausência de encadeamento.

O DMAC "pensa" que o dispositivo tem dados de 16 bits, mas a lógica exterior provoca transferências de 32 bits. O modo de encadeamento de grupos ligados usado em CHO (canal 0 de DMAC (209)) implica que existe uma lista encadeada usada e que ela é ajustada pela rotina de inicialização ETIO.



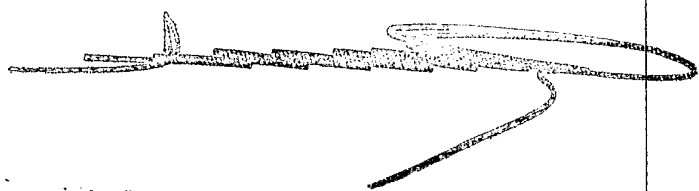
Uma vez verificado o arranque de CHO, ele só pára devido a uma condição de erro, ou por se encontrar a última entrada válida na lista encadeada. No funcionamento normal, verifica-se uma interrupção de S/88 de cada vez que o DMAC (209) completa uma interrupção de caixa do correio e os meios de microprogramas vigiam e preenchem a lista encadeada em tempo real; assim a última entrada válida da lista não é nunca atingida e CHO funciona continuamente (funciona em vazio).

Cada um dos canais DMAC está provido de dois registadores de vectores de interrupção NIV e EIV (fig. 18), um para fim de operação normal e outro para fim forçado por um erro detectado. O presente aperfeiçoamento utiliza todos os oito vectores, com oito rotinas de interrupção ETIO separadas na memória de microcódigos (174). Adicionalmente, a interrupção normal do canal 0 tem dois significados possíveis: um, "caixa do correio recebida", interrupção provocada por PCL e o menos comum "canal parado devido ao fim de uma lista encadeada". O dispositivo de tratamento das interrupções diferencia estes dois significados por ensaio de um bit de estado DMAC.

Os meios de microprogramas S/88 proporcionam quatro entradas de serviços para o programa de aplicação EXEC370: inicialização e arranque das três transferências de dados básicas atrás descritas - leitura de dados, escrita de dados e escrita de mensagens na fila de espera.

A entrada ETIO-INICIALIZE é usualmente chamada logo depois da ligação da energia, mas pode também ser usada para reinicializar para tentativas de recuperação de erros. Ela repõe os meios físicos BCU e o DMAC (209), depois programa os registadores DMAC, em todos os quatro canais, com valores de configuração e de controlo. Forma também a lista encadeada e faz arrancar o canal 0, fazendo com que o DMAC (209) se autocarregue com o primeiro conjunto de parâmetros da lista encadeada e depois espere por uma transição de pedido proveniente dos meios físicos BEU na linha (263a).

As outras três entradas de serviços são chamadas para fazer arrancar os canais 1 (ler dados), 2 (escrever dados) e 3 (escrever mensagens na fila de espera) do DMAC. O programa de chamada (EXEC370) proporciona um indicador para



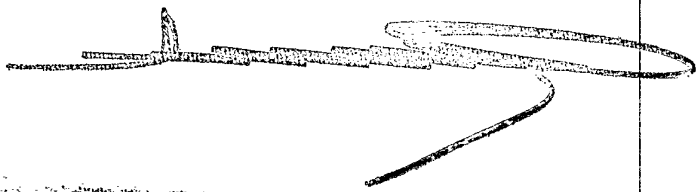
um bloco FILA DE ESPERA DE TRABALHO que foi pré-ajustado com en dereços de dados, valores de contagem, etc.. Estas rotinas ou fazem arrancar imediatamente o DMAC (209) e os meios físicos BCU, ou introduzir na fila de espera a operação, se o canal DMAC pedido estiver ocupado (Uma fila de espera "de trabalho pendente" separada, representada na fig. 41E, é mantida para ca da um destes três canais). Uma vez o serviço pedido iniciado ou introduzido na fila de espera, o controlo volta ao programa de chamada e o processador das interrupções continua o funcionamento até se completar.

Uma terceira área, pequena mas crucialmente importante, dos meios de microprogramas de cliente S/88 é a modificação do sistema operativo (OS) S/88 para intersectar e orientar as oito interrupções DMAC para os meios de tratamento de cliente, mas transparentes ao sistema operativo S/88. Isso implica modificações na tabela de vectores architectada normalizada MC68020 no OS para o nível 6 (que é normalmente autoorientada no cado das falhas de energia) e a colocação dos meios de tratamento das interrupções de cliente no OS. É essa uma realização preferida; mas, como adiante se verá, na secção relativa às rotinas de inicialização para interrupções, pode proporcionar-se uma lógica na BCU (156) para colocar um vector na linha omnibus local (223), eliminando a necessidade da modificação de vectores.

Todos os meios de microprogramas S/88 pa ra a força de realização preferida estão escritos em linguagem "assembler MC67020", podendo assim ser de maneira apropriada de signados por microcódigo. São considerados meios de microprogramas, devido à natureza das suas funções.

Há quatro categorias de meios de microprogramas de cliente necessários para o processador S/370 (85):

1. Instruções I/O de microcódigo, que vão para o pseudocanal S/88,
2. Tratamento de mensagens assíncronas que vêm do S/88, incluindo interrupções I/O,
3. Manutenção de dados e estados de configuração de todos os serviços I/O S/370 (emulados), e
4. Realização prática de um subconjunto de operações manuais



de utilizador.

Todos estes meios especiais de microprogramas são escritos em microcódigo S/370 e utilizam, sempre que possível, sub-rotinas funcionais pré-existent.

Há dez instruções do tipo I/O no S/370, que são discutidas com mais pormenor com referência à descrição das fig. 44 A-I:

CLRCH	- "Clear channel" (limpar o canal) (op. de canal único)
CLRIO	- "Clear I/O" (limpar entrada/saída)
HDV	- "Halt device" (reter o dispositivo)
HIO	- "Halt I/O" (reter entrada/saída)
RIO	- "Resume I/O" (retomar entrada/saída)
SIO	- "Start I/O" (arrancar entrada/saída)
SIOF	- "Start I/O Fast" (arrancar I/O rapidamente)
STIDC	- "Store channel ID (armazenar canal ID (op. de canal único)
TCH	- "Test channel" (ensaiar canal (op. de canal único)
TIO	- "Test I/O" (ensaiar entrada/saída)

Cada uma destas instruções é realizada em microcódigo, de modo a passar toda a informação essencial para EXEC370 no S/88 através do mecanismo de caixa de correio, mantendo no entanto a concordância com a arquitectura S/370.

Várias condições diferentes nos meios físicos no adaptador (154) têm como uma consequência a activação do pedido de "Adapter Attention" que, por sua vez, é uma das várias causas possíveis de uma "Forced Exception" num nível de microcódigo no processador S/370 (85). O tratamento desta excepção pelo microcódigo efectua-se entre instruções S/370 (imediatamente, se o PE (85) estiver no estado de espera). A causa mais frequente e comum de "Adapter Attention" é a recepção pelo PE (85) de uma mensagem proveniente do pseudocanal I/O S/88 para a área fixa "Message-Q" (mensagens na fila de espera) da secção IOA da memória principal S/370.

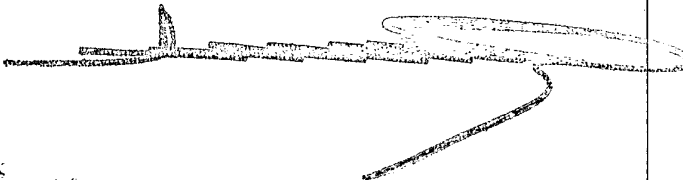
O meio de tratamento das excepções de microcódigo S/370 existente é modificado para o caso "Adapter Attention". O código ensaia o adaptador (154) para determinar a causa do pedido e trata particularmente apenas o tratamento de

"Q-not-empty" (fila de espera não vazia) (o que significa mensagem recebida); qualquer outra causa regressa ao código não modificado para o tratamento.

As categorias definidas de mensagens recebidas são:

0000 NOP	Nenhuma operação
0001 RESET	Invocar a rotina existente de reposição de programa S/370
0002 CLEAR RESET:	Invocar a rotina existente Clear Reset S/370
0003 HALT:	Reter a execução do programa S/370, passar para o modo ISTEP
0004 STEP:	Fase de instrução; executar uma instrução e depois parar
0005 RUN:	Repor o modo ISTEP; reatar a execução do programa
0006 LPSW:	Executar a função S/370 "Load PSW", usando uma PSW proporcionada no interior da mensagem. Abandonar o estado HALTED (de paragem)
0007 SMSG:	Mensagem de estados - actualizar os bits de estado na Tabela de Estados dos Dispositivos local (IOA) para um ou mais dispositivos de I/O configurados.
0008 IMSG:	Mensagem de interrupção - introduzir na fila de espera ou apresentar imediatamente uma interrupção I/O S/370, conforme o estado da máscara do canal.

Os tipos de mensagens 0001-0006 anteriores são operações manuais S/370, para controlo do estado, resultados de entrada do utilizador na Consola do Sistema S/370 (emulada). Podem também ser forçadas directamente por EXEC370, conforme for necessário, para a recuperação de erros ou sincronização. O tipo de mensagem 0007 é usado para informar o S/370 de alterações assíncronas de estado nos dispositivos de I/O, tais como perda de energia, alterações de ON/OFF-LINE, erros detectados em dispositivos, etc.. Podem também ser expandidos, para comunicação para fins gerais do S/88 para o S/370. O tipo de



mensagem 0008 é o veículo para comunicar estados de operação de fim de I/O para o S/370 - quer de condições normais, quer de condições de fim de avaria. Terá sempre como consequência uma eventual modificação de Program Interrupt e Device Status Table no S/370.

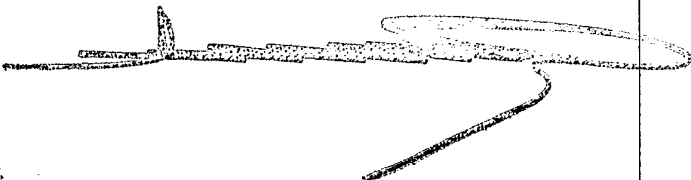
Vão agora discutir-se alguns pormenores das funções ETIO e EXEC370, interfaces, protocolos e fluxos de instruções.

Concepção do microcódigo do sistema

1. Introdução

A fig. 38 ilustra a concepção do microcódigo para uma forma de realização preferida do presente aperfeiçoamento. O código que é executado na unidade de processamento S/370 (cada elemento de processamento tal como o (85)) é mantido na memória de controlo (171) e interpreta instruções S/370, quando elas são executadas pelo PE (85). As instruções microcodificadas para Start I/O, tratamento de interrupções, funções de operador, verificação da máquina e carga de microprogramas/carga de programas inicial (IML/DPL) são concebidas especificamente para estabelecer a interface com o microcódigo S/88 como se mostra na figura. A interface inclui as facilidades de meios físicos comuns da lógica da interface (81), incluindo a memória local (210), a memória "cache" S/370 e o espaço de memória real S/370 (162) com capacidade de interrupções para o microcódigo dos dois processadores (85) e (62). No código S/88, o activador do microcódigo S/370 inclui o conversor de palavras de comando (CCW), o dispositivo de tratamento de interrupções, o dispositivo de tratamento de erros, IML/IPL e código de sincronização, que interage com a interface de aplicação S/88 (EXEC370) e o sistema operativo S/88.

O processador (62) tolerante das avarias executa tudo referente a I/O, diagnóstico, isolamento da avaria, IPL/IML e sincronização para o sistema. Este sistema não é visto com um sistema coprocessador, porque os programas S/370 são os únicos programas executados do ponto de vista dos utilizadores.



O gestor do sistema pode controlar os atributos do sistema através do sistema S/88, que funciona com tolerância das avarias. A função primária do sistema operativo S/88 e a aplicação EXEC370 é a conversão I/O com uma aparência de canal 370 múltiplo. Todas as funções de erros e recuperação do sistema e de atribuição dinâmica de recursos são tratadas pelo sistema operativo S/88. As funções de verificação da máquina e de operador anteriormente tratadas pelo sistema operativo S/370 são agora passadas para o sistema operativo S/88 de modo que as funções podem ser tratadas de uma maneira tolerante das avarias.

A fig. 39 ilustra a execução de um comando I/O S/370, neste exemplo um comando START I/O. As acções tomadas pela instrução S/370, o microcódigo S/370, os meios físicos de acoplamento (PE (85) para PE (62)), o microcódigo de acoplamento ETIO (executado no PE (62)) e o programa S/88 EXEC 370 estão representados resumidamente, sendo o passo final a execução do SIO S/370, no processador S/88 PE (62).

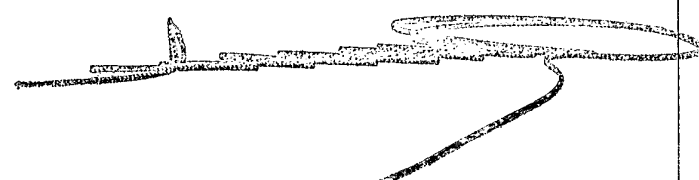
A fig. 40 é uma vista geral simplificada que ilustra resumidamente alguns dos componentes e funções do sistema aperfeiçoado relativamente ao EXEC370 e ao dispositivo de activação do microcódigo usado durante a execução de SIO, juntamente com o fluxo de comandos, o fluxo de dados e a comparimentação de sinais e meios físicos/códigos.

2. INTERFACE DE PROGRAMA ETIO/EXEC 370 - fig. 41A-H, 42

Utilizam-se os termos seguintes nesta secção:

EXEC370 - Todo meio de programação que se executa no PE (62) pertencente à emulação e suporte de dispositivos exteriores, serviços, configuração, consola de operador etc. S/370 e memorizado na memória de microcódigos (174), O código EXEC370 usado menos frequentemente pode ser armazenado na memória "cache" (173).

• MICROCÓDIGO S/370 - O microcódigo que é
• executado no processador S/370 (85) que suporta operações do
• processador S/370 e armazenado na memória (171).



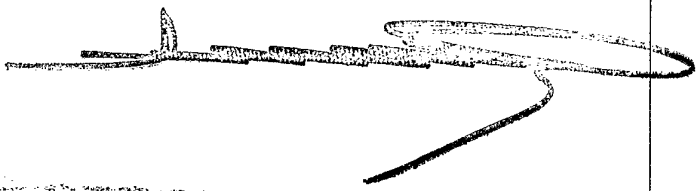
ETIO - Interface do microcódigo entre EXEC370 e os meios físicos da BCU e mantido na memória (174).

O microcódigo S/370 PE (85) e EXEC370 comunicam um com o outro através de um "protocolo", fig. 41A. O microcódigo PE (85) envia mensagens para EXEC370 pedindo a execução de funções como I/O, e EXEC370 envia mensagens que indicam que terminaram as funções I/O, mensagens respeitantes a alterações de estado de dispositivos e de canais e mensagens que pedem o microcódigo PE (85) para efectuar funções CPU S/370 específicas. Estas mensagens (descritas em pormenor mais adiante) são transmitidas entre o microcódigo PE (85) e EXEC370 através de meios físicos que incluem o controlador da memória "cache" (153), o adaptador (154), a BCU (156) e o seu DMAC (209), etc.. Este serviço de transmissão de mensagens é posto pela ETIO à disposição de EXEC370.

A interface entre ETIO e EXEC370 e o protocolo entre o microcódigo PE (85) e EXEC370 vai agora ser descrito.

A interface (fig. 41B) entre EXEC370 e ETIO, os meios de programação de suporte externo S/370, executados pelo S/88, e do activador do microcódigo BCU (ETIO), executados no PE (62) consiste num conjunto de filas de espera e memórias tampão residentes na memória (210), um evento ID, uma variável EXBUSY e uma sequência de chamada de sub-rotinas. A interface CHAMADA de sub-rotinas indica operações de transferência de dados entre S/88 e S/370 e inicializa a DMAC (209) e a BCU (156) no instante da reinicialização. A interface de filas de espera é usada para manter a pista de itens de trabalho até eles poderem ser processados e a interface de eventos ID (uma interrupção para S/88) notifica EXEC370 quando se tiver adicionado trabalho às filas de espera.

Na memória (210) há dezasseis blocos de 4KB (500), fig. 41C. Catorze (500-0 a 500-13) são usados como blocos de memórias tampão de 4KB. Os dois restantes são divididos em trinta e dois blocos de 256 octetos (501-0 a 501-31). Quatro blocos (501-0 a 501-3) são usados para comunicação por meios físicos, um (501-4) para filas de espera e outras variáveis comuns a EXEC370 e ETIO. As restantes vinte e sete são usa



das como memórias de filas de espera de trabalho (WQB) (501-5) a (501-31). No espaço de endereços equivalente aos blocos (501-0) e (501-1) são atribuídos a comandos da BCU (156) (executados pelo PE (62)) 256 octetos e aos endereços dos registradores DMAC são atribuídos 256 octetos para acesso pelo PE (62), como se descreveu com referência às operações da BCU (156). Cada uma das vinte e sete Memórias Tampão das Filas de Espera de Trabalho retêm dados pertencentes a uma tarefa ou pedido de serviço específicos. Vinte e seis WQB são usadas para serviço de pedidos iniciados por microcódigos PE (85). A WQE restante (EXWQE) (501-31) é reservada para servir pedidos originados pelo S/88 e enviados para o microcódigo PE (85); nunca aparecerá na fila de espera livre FREE Q (fig. 41E). Cada bloco WQB, fig. 41D, contém um bloco de caixa de correio (505) de 16 octetos, um bloco de parâmetros (506) de 16 octetos e uma área de trabalho específica de dispositivos (507). O bloco de caixa de correio (505) contém dados passados entre EXEC370 e o microcódigo PE (85). O seu conteúdo é transparente através da interface ETIO. O bloco de parâmetros (506) contém parâmetros passados entre ETIO e EMEC370, usualmente respeitantes à transferência de dados entre a memória local (210) e a memória principal (162). A área de trabalho (507) é propriedade do EXEC370. Contém informação acerca do progresso da operação de pedido, do estado corrente dos dispositivos S/370, de dados do utilizador possíveis, do tipo dos dispositivo S/88, indicadores para outros blocos de controlo EXEC370, informações sobre ocorrência de erros, etc..

O bloco da caixa de correio (505) inclui quatro campos, contendo informação de I/O S/370 passada entre o microcódigo PE (85) e EXEC370:

- OP - Este campo contém um pedido de EXEC370 ou do microcódigo PE (85)
- CUA - Endereço de 16 bits da unidade de canal
- CAW - Palavra de endereço do canal S/370 de 32 bits da posição hexadecimal 48 na memória S/370 (162) quando se emitiu a instrução I/O em questão.
- CCW - Palavra de comando de canal S/370 endereçada pela CAW anterior. Quando EXEC370 devolve uma indicação de interrupção, este campo contém a CSW, a

palavra do estado do canal.

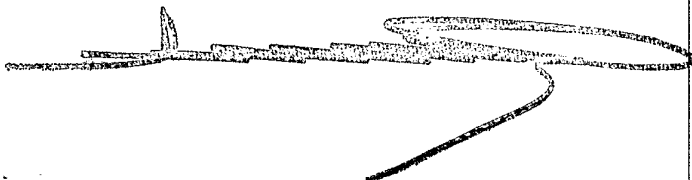
O bloco de parâmetros (506) contém seis parâmetros usados quando é pedida uma transferência de dados entre a memória (210) e a memória principal (162) por EXEC370.

1. req - campo de pedido ETIO:
 - 0 não operação
 - 1 escrever o conteúdo do bloco da caixa de correio na fila de espera de mensagens PE (85) (189) na memória (162) e depois enviar um pedido BCU para PU pela linha (256a).
 - 2 Ler dados a partir da memória S/370
 - 3 Escrever dados para a memória S/370
2. req - resultados do pedido feito pelo campo "req"
Este campo é garantido por EXEC370 para ser inicialmente zero. Se não for zero na reposição, ETIO está a indicar um erro de qualquer tipo.
3. CONTAGEM - o número de octetos a transferir.
4. ENDEREÇO S/370 (S/370 ADDR) - situação na memória S/370 onde começa a área de dados. Este não é necessariamente um valor do campo do endereço da CCW
5. Chave - Este campo de 16 bits conterá o seguinte padrão de bits:
ppkkkk10 00000000
sendo pp (paridade) = 00 e KKKK = a chave de protecção da própria memória S/370

Endereço da memória tampão (BUFF ADDR)

- A posição na memória (210) onde começa a área de dados. Pode estar no interior de uma memória de 4K ou de um bloco WQB. EXEC 370 assegurará a relação seguinte: (S/370 ADDR módulo 4) = (BUFF ADDR módulo 4).

EXEC370 utiliza filas de espera para manter os blocos WQB. A área de comunicação das filas de espera (501-4) tem 256 octetos de comprimento e reside no desvio 400 (hex) na memória (210). A fig. 41E mostra as filas de espera de finidas entre ETIO e EXEC370 para reter as entradas dos indicadores para os blocos WQB:



Fila de espera livre (FREEQ) (510) retêm os indicadores para os blocos WQB não usados correntemente.

Fila de espera de trabalho WORKQ (511) retêm indicadores para os blocos WQB que esperam para ser servidos por EXEC370.

S/3701Q (512) retêm indicadores para os blocos WQB que esperam transferências de mensagens de EXEC370 para PE (85).

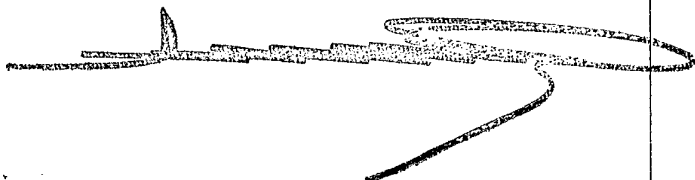
S/3702Q (513) retêm indicadores para blocos WQB que esperam por transferências de dados do controlador da memória "cache" (153) para S/88.

S/3703Q (514) retêm indicadores para blocos WQB que esperam por transferências de S/88 para o controlador da memória "cache" (153).

S88Q (515) retêm indicadores para blocos WQB depois de completado o serviço ETIO.

A fig. 41E representa o trajecto dos blocos WQB através das filas de espera. Todas as filas de espera são inicializadas por EXEC370 durante a reinicialização do S/88. Os blocos WQB vazios são mantidos no campo FREEQ. ETIO retira-os do FREEQ quando necessário para preencher as listas encadeadas (516). O DMAC (209), através da lista encadeada (516), coloca entradas de caixa de correio S/370 provenientes da área de caixa de correio (188) da memória (162) para as áreas de blocos de caixa de correio de blocos WQB vazios. Os blocos WQB na lista encadeada que tenham sido preenchidos são deslocados para a fila de espera de trabalho (511) por ETIO. Quando ETIO coloca um (ou mais) blocos WQB na fila de espera de trabalho (511) e EXEC370 está não ocupado, ETIO notifica o EX370 event ID. EXEC 370 retira o bloco WQB da lista de espera de trabalho antes de atender o pedido.

Durante o processamento do pedido, podem ter de transmitir-se dados entre o controlador da memória "cache" (153) e a memória tampão (WQB ou memória de blocos), ou pode ter de enviar-se uma mensagem para o microcódigo PE (85). ETIO proporciona este serviço para EXEC370. EXEC370 chama ETIO, que inicia a operação apropriada da BCU (156) ou, se os equipamentos físicos estiverem ocupados, coloca o bloco WQB na fila



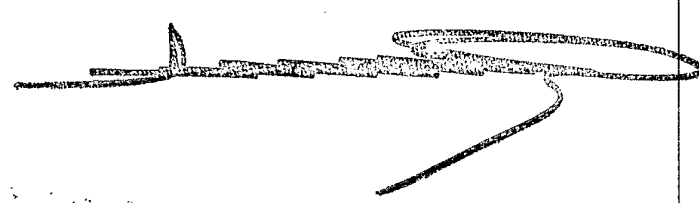
de espera E/30 Q apropriada. Cada um dos três serviços (enviar mensagens para S/370, transferir dados para S/370 e transferir dados de S/370) tem as suas filas de espera próprias (512), (513) e (514). Adicionam-se blocos WQB a uma das filas de espera S/370 pelo código ETIO enquanto no encadeamento do EXEC370. Quando se tiver completado o serviço I/O, a rotina de interrupção ETIO coloca o bloco WQB na fila de espera S/88 (515) e, se EXEC370 estiver não ocupado, notifica o EX370 event ID.

A fig. 42 ilustra o movimento dos blocos WQB através das filas de espera juntamente com as interfaces entre EXEC 370, ETIO, os meios físicos de interface (89) e o microcódigo S/370. Quando estiver inteiramente completado o pedido de trabalho original, isto é, completa a transferência de dados, é enviada uma interrupção IO (se existir alguma) para o PE (85) e EXEC370 reenvia o bloco WQB para a fila de espera livre FREEQ. EXEC370 obtém então a sua tarefa seguinte, verificando primeiro a fila de espera S/88 (515) e a fila de espera de trabalho (WORK Q) (511). Se ambas estiverem vazias, EXEC370 põe uma variável EXBUSY em zero e espera que seja notificado o evento EX370. EXEC370 ajusta EXBUSY para 1 quando for notificado, antes de começar o processamento.

Todas as filas de espera, o evento ID do EX370 e a variável EXBUSY residem na área COMM (501-4)" da fila de espera da memória (200), como se mostra na fig. 41F. Todas as filas de espera são de natureza circular, como se mostra na fig. 41G, com indicadores com dois tipos de índices: um índice FILL (Preencher) (517) e um índice EMPTY (esvaziar) (518). O índice FILL (517) aponta para a entrada seguinte da fila de espera a preencher e o índice EMPTY (518) aponta para a entrada seguinte a esvaziar. Se os índices EMPTY e FILL forem iguais, a fila de espera está vazia. As seis filas de espera nunca terão transbordo, visto que cada uma tem 32 entradas e há apenas 27 blocos WQB.

Cada fila de espera contém também:

qid	identifica esta fila de espera
QSIZE	número de entradas nesta fila de espera (n)
Q(i)	entradas e endereço que apontam para os blocos WQB na fila de espera



A área de comunicação dos meios físicos contém 1 024 octetos. A área de comunicação BCU utiliza 512 octetos do espaço de endereços. As listas encadeadas (516) tomam 480 octetos. 32 octetos estão reservados para comunicação de outros meios físicos. A lista de elos de ligação (516) (fig. 41H) é usada pelo DMAC (209) para levar para a caixa de correio itens de blocos provenientes da área da caixa de correio (188) da memória (162). Os blocos WQB provenientes da fila de espera FREE Q (510) são usados para preencher entradas na lista de elos de ligação (516). Cada entrada da lista de elos de ligação contém dez octetos e identifica o endereço do bloco WQB na memória (210) no qual devem colocar-se os dados, o número de octetos dos dados a transferir (16) e o endereço da entrada do elo de ligação seguinte. O DMAC (209) (canal 0) interrompe S/88 quando atinge uma entrada da lista de elos de ligação com um endereço zero para o elo de ligação seguinte. A posição corrente do DMAC (209) (canal 0) na lista está em qualquer instante disponível para os meios de programação.

Além dos seus pontos de entrada de interrupção, ETIO tem dois pontos de entrada susceptíveis de ser chamados, externos:

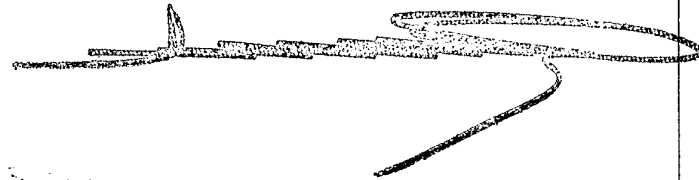
ETIO INIT

ETIO(WBN)

EXEC370 chama ETIO INIT uma vez em cada reinicialização S/88, enquanto EXEC370 está a inicializar-se. As filas de espera já foram inicializadas e os campos EVENT ID estarão válidos. O microcódigo PE (85) ainda não estará a funcionar, mas pode estar no processo IML (carga inicial dos microprogramas).

EXEC370 chama ETIO (WBN) sempre que deseja transferir dados ou mensagens de/para S/370.

O parâmetro WBN é um número interior de dois octetos (Work Quene Buffer Number - Número da memória da fila de espera de trabalho) que identifica o bloco WQB que contém o pedido de serviço. WBN é o valor de um índice, que vai de 0 a 27. O pedido de serviço é identificado pelo campo REQ no bloco de parâmetros. Os valores do campo REQ são : 1 = escrever os conteúdos deste bloco de caixa do correio na fila de espera de mensagens S/370 (189) na memória (162) e depois emitir um pe



dido da BCU para a PU ; 2 = ler dados da memória S/370 (162) para a área especificada da memória (210) e 3 = escrever dados para a memória S/370 provenientes da área especificada da memória (210). A subrotina ETIO coloca este bloco WQB em fila de espera nas filas de espera S/3701Q, S/3702Q ou S/3703Q, se a função de I/O pedida não puder ser iniciada imediatamente. A rotina de interrupção ETIO retirará o bloco WQB seguinte da fila de espera S/370 Q apropriada quando terminar a operação anterior.

Se o campo REQ tiver um 1, o microcódigo PE (85) não deve ser notificado (por exemplo por uma interrupção) até que a entrada do bloco de caixa do correio esteja na área de fila de espera de mensagens S/370 (189) da memória (162).

Se a fila de espera de mensagens S/370 (189) estiver cheia, um erro no campo RET do bloco Parm (parâmetro) identificará o problema para EXEC370. Se for necessário, EXEC370 pode suportar uma fila de espera de apoio.

3. PROTOCOLO DO MICROCÓDIGO S/370, EXEC370

A comunicação entre EXEC370 e o microcódigo S/370 requer uma tabela de estados dos dispositivos (DST - Device Status Table) com uma entrada para cada dispositivo I/O na memória S/370 (162). EXEC370 e o microcódigo S/370 comunicam entre si através de mensagens de 16 octetos (ver Bloco da caixa de correio (505), fig. 41D) que são enviadas para um lado e para o outro. Há uma lista de espera que retém as mensagens por ordem FIFO (primeira entrada-primeira atendida) para o receptor em cada extremidade. Há também um mecanismo de notificação (linhas PU para BCU e BCU para PU). No bloco de caixa de correio (505) o campo "OP" (código de operações S/370 de 16 bits) contém um pedido ou uma resposta vindos de EXEC370 ou do microcódigo S/370. O endereço CUA (Channel Unit Address - Endereço da unidade do canal) é o endereço do operando de uma instrução I/O S/370. CAW é o conteúdo de 32 bits da posição HEX (48) na memória S/370 (162), quando a instrução de I/O foi enviada, e inclui a chave de memória. A CCW de oito octetos é endereçada pela CAW



referida. Quando EXEC370 devolve uma indicação de interrupção, este campo contém a CSW. O PE (85) armazena a CSW na posição 40 Hex S/370 quando ele provoca uma interrupção de I/O. O campo CUA ficará sem modificação.

A mensagem OPERATION é enviada para EXEC 370 pelo microcódigo S/370 sempre que for encontrada uma instrução S/370 que deva ser total ou parcialmente tratada pelo EXEC 370. A mensagem OPERATION contém a informação atrás descrita relativa ao bloco de caixa de correio (505) da fig. 41D.

As mensagens EXEC370 enviadas para o microcódigo S/370 incluem:

1. A mensagem RESET (OP = 1) pede que o microcódigo S/370 processe um Reset S/370.
2. A mensagem CLEAR RESET (OP = 2) pede um Reset and Clear Storage S/370 (repôr e limpar a memória S/370).
3. A mensagem HALT pede que o microcódigo S/370 se abstenha de buscar instruções S/370 e espere por instruções ulteriores. A mensagem HALT inclui um campo OP = 3.
4. A mensagem STEP (OP = 4) pede que o microcódigo S/370 busque e execute uma instrução S/370 e entre depois no modo HALT.
5. A mensagem RUN (OP = 5) pede que o microcódigo S/370 entre no seu modo normal de buscar e executar instruções S/370.
6. A mensagem LPSW (OP = 6) pede que o microcódigo S/370 efetue uma instrução LPSM (Load Program Status Word - Carregar a palavra do estado do programa), utilizando o endereço especificado no campo ADDRESS da mensagem LPSW. Pode ser usada para retirar o microcódigo S/370 da condição HALT.
7. Mensagem MSG (OP = 7) indica alterações de estado para um ou mais dispositivos de I/O configurados no S/370.
8. Mensagem IOINTR (OP = 8) indica que se completou uma operação de I/O. Se o canal não tiver a máscara OFF, o microcódigo S/370 iniciará uma operação de interrupção de I/O. Se o canal tiver a máscara OFF, o microcódigo S/370 preservará a CSW na tabela de estados dos dispositivos e ajustará o estado do dispositivo para OL (CSW registada em memória). A mensagem IOINTR também inclui o campo seguinte CUA e NC (colocar em DST CUA).

Duas mensagens, FETCH e STORE, do S/88

para o controlador de memória "cache" (153) são funções lógicas, em vez de mensagens. É necessário permitir um valor par ou ímpar de campos CNT e ADDRESS.

Os seus campos são:

BUF - 2 octetos endereço da memória tampão na memória (210)

CNT - 2 octetos contagem de octetos

ADDR - 4 octetos endereço W/KEY na memória S/370.

O microcódigo S/370 mantém uma tabela contendo informação acerca do estado de cada dispositivo S/370 endereçável. As peças principais de informação são:

Condição do dispositivo - permite o ajustamento imediato do CR (registador de condições S/370), depois de um TIO, SIO, etc.

Dispositivo a seguir - a condição seguinte a usar quando toma uma interrupção I/O

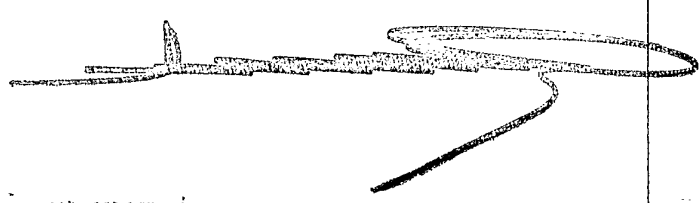
Device CSW - mantida para interrupções I/O 370 com máscara.

São possíveis quatro condições diferentes dos dispositivos no DST (CUA), para um dispositivo 370:

00	Dispositivo pronto
01	Dispositivo não pronto, CSW armazenada
10	Dispositivo ocupado
11	Dispositivo não operacional

Quando se completa uma operação I/O num dispositivo S/370, é enviada uma palavra de estado do canal (CSW) pelo canal para a CPU. Se o canal tiver a máscara OFF a CPU não aceita a CSW.

Na presente aplicação, se o canal tiver uma máscara, o microcódigo S/370 preserva a CSW e ajusta a condição DST(CUA) para 01. Uma TIO ou SIO subsequentes terão como resultado o armazenamento da CSW preservada, sendo colocado o código de condição 01 (CSW memorizada) no CR. Quando o microcódigo S/370 é inicializado, ele parte da hipótese de que todos os dispositivos estão não operacionais. S/88 emitirá uma mensagem ONLINE para cada dispositivo que deve ser suportado. O dispositivo é identificado pelo seu CUA (endereço da unidade de



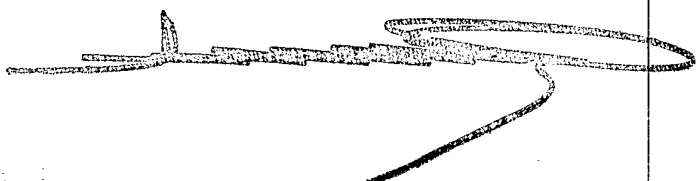
controle).

4. Fluxos de instruções entre o microcódigo S/370 e EXEC370

Enquanto o PE 85 executa sequências de instruções de um programa, S/370 encontrará de tempos a tempos uma instrução de I/O que, na presente aplicação, será executada pelo processador S/88 (62) e os meios físicos, de programação e de microprogramas com ele relacionados. As fig. 44 A-L (e a fig. 43 atrás referida) ilustram a sequência de microcódigos utilizada para a execução destas instruções de I/O S/370. A BCU (156) (e o adaptador (154)) é o mecanismo de acoplamento de meios físicos primários para efectuar a execução final da instrução I/O S/370 pelos meios físicos S/88. No interior da BCU (156), o controlador DMAC (209) é o "polícia sinaleiro" principal para orientar o fluxo de operações e dados. O canal 0 do DMAC (209) recebe comandos I/O do S/370, o canal 1 trata o fluxo de dados do S/370, o canal 2 trata o fluxo de dados para o S/370 e o canal 3 emite mensagens de interrupção (e outras) para o S/370. A memória local (210) na BCU (156) constitui a área de comunicação entre o S/370 e o S/88.

A linha omnibus local (223/247) acopla o processador S/88 (62) ao DMAC (209) e à memória local (210). A linha omnibus local (223/247) acopla o DMAC (209) e a memória (210) ao S/370, através de meios físicos de aceleração na BCU (156) e no adaptador (154).

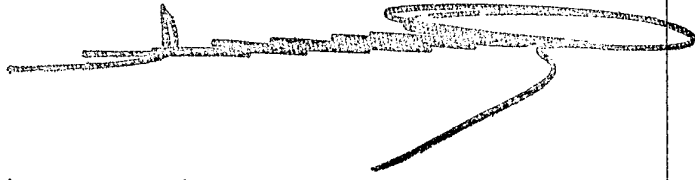
As instruções I/O S/370 são despachadas para as rotinas do microcódigo S/370, para tratamento no interior do S/370, e um programa de aplicação S/88 EXEC 370 (juntamente com o seu microcódigo ETIO S/88 relacionado) efectua a execução final da I/O. O adaptador (154) e a BCU (156) constituem a ligação de meios físicos entre o código S/370 e S/88. A rotina de arranque do microcódigo I/O tem uma tabela DST, que segue o estado de cada dispositivo, por exemplo; está actualmente disponível, ele já emitiu uma SIO, está ele ocupado?, recebeu ele em resposta uma interrupção?. Esta informação está contida no código de condição CC.



Esta secção descreve o fluxo de instruções para várias operações I/O S/370. Alguns processos e termos específicos usados nesta secção estão definidos no fim da secção. As operações são as seguintes:

1. CLEAR CHANNEL (fig. 44A) - Esta instrução provoca uma I/O System Reset a executar no canal endereçado, com uma reposição do sistema sinalizada para todos os dispositivos no canal endereçado. O microcódigo S/370 não sabe quais os dispositivos que estão actualmente no canal, de modo que ajusta CC = 3 para todas as entradas DST nesse canal. Subsequentemente, EXEC 370 enviará uma mensagem ou mensagens SMSG para redefinir a configuração nesse canal.

O canal a limpar é endereçado pelos bits 16 a 23 do endereço da instrução. Quando o microcódigo S/370 receber o controlo do despacho, ele começa por verificar o endereço do canal (CA). O endereço do canal será válido ou inválido. Se o endereço do canal for inválido, o registador de condições (CR) é ajustado para 3 e o S/370 passa à instrução sequencial seguinte. Um canal que seja suportado pelo microcódigo S/370 é considerado ter um endereço de canal válido. Para o endereço de canal válido, o microcódigo S/370 envia uma mensagem de limpeza do canal (CLRCH) para EXEC370. Depois passará por todas as entradas da tabela de estados dos dispositivos (DST) para este canal. Todos os campos do código de condição são ajustados para 3, significando que não estão disponíveis e quaisquer entradas da tabela de interrupções pendentes (PIT) encontradas são libertadas para uma lista de PIT livres. O microcódigo S/370 ajusta então o registador de condição para 0 e passa à instrução sequencial seguinte. Entretanto EXEC370, quando receber a mensagem de limpeza do canal, efectua uma reposição do sistema I/O para todos os dispositivos no canal endereçado. Certifica-se depois de quais os dispositivos que estarão em linha e envia uma mensagem de estado para o microcódigo S/370, para redefinir a configuração nesse canal. Quando o microcódigo S/370 receber a mensagem de estado, ele modifica o código de condição na tabela de estados dos dispositivos, para cada dispositivo endereçado para ele na mensagem de estados.

- 
2. CLEAR I/O (fig. 44B) - Esta instrução suspende a execução da instrução S/370 em processamento no PE (85) até ser devolvido para EXEC370 IMSG para o CUA endereçado.

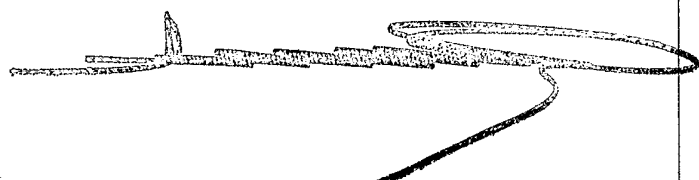
Quando o microcódigo S/370 recebe o controle do despacho, ele obtém o endereço CUA da unidade de controle a partir do endereço da extremidade superior da instrução. Utilizando o endereço da unidade de controle, ele encontra a entrada correcta da tabela DST dos estados dos dispositivos, para este dispositivo. Verifica o valor do código de condição (CC). Há três opções: 1) CC igual a zero ou 3, 2) CC igual a 2, ou CC igual a 1 e a condição NC seguinte igual a 2 e 3) CC igual a 2, ou CC igual a 1.

Para a primeira opção, CC igual a zero ou 3, o microcódigo S/370 apenas ajusta o registador de condições para o valor de CC e passa à instrução seguinte na sequência.

Se CC for igual a 1, há uma interrupção pendente na tabela de interrupções pendentes (PIT). Neste caso, o microcódigo S/370 vai para a entrada da tabela de interrupções pendentes e verifica o valor de NC.

Para o caso de ser CC igual a 2 ou CC igual a 1 e NC igual a 2, S/370 envia uma mensagem CLEAR I/O para EXEC370 (CLRIO). Espera pela confirmação e limpa quaisquer entradas de interrupção pendente associadas com o dispositivo. Espera então pela mensagem de interrupção a ser devolvida por EXEC370. Entretanto quando EXEC370 receber a mensagem de limpeza I/O (CLRIO), efectua a sua reposição selectiva do dispositivo endereçado, forma uma palavra de estado de controle para o dispositivo e devolve uma mensagem de interrupção para o microcódigo S/370. Quando o microcódigo (370) recebe a mensagem de interrupção, gera a entrada PIT e preenche o NC e a CSW a partir da mensagem. A entrada de interrupção pendente é depois ligada à entrada da DST.

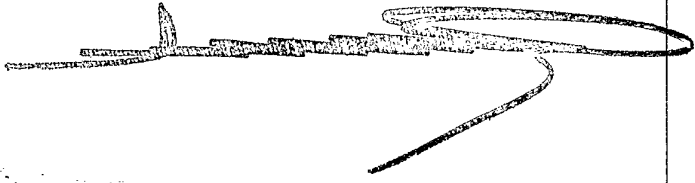
Nesta altura, chegamos à terceira opção, CC igual a 2 ou CC igual a 1. Chegamos a este ponto por duas vias. A primeira via é a de o dispositivo estar ocupado ou o dispositivo ter enviado uma interrupção pendente, mas man



tendo-se ocupado. Este é o caso de ser enviada a reposição selectiva. A segunda via é quando o dispositivo tem uma interrupção pendente mas já não está ocupado. Para ambas estas vias, CC será igual a 2 ou 1. É esta a terceira opção. O microcódigo solta a interrupção, coloca a CSW na memória S/370, ajusta o registador de condição para 1 e passa à operação sequencial seguinte.

3. HALT DEVICE (fig. 44C) - Quando o microcódigo S/370 receber o controlo do despacho para uma instrução HALT DEVICE, ele verifica o código de condição para a entrada na tabela de estados dos dispositivos endereçada. Há três opções, um código de condição igual a 0 ou 2, o código de condição igual a 1 ou o código de condição igual a 3. Para a primeira opção, código de condição igual a 0 ou 2, o microcódigo S/370 envia uma mensagem de retenção do dispositivo (HDV) para EXEC370. Ajusta a 0 os 16 bits de estado na CSW S/370, ajusta o registador de condição para 1 e passa à instrução sequencial seguinte. Entretanto, quando EXEC370 recebe a mensagem HDV, executa a função apropriada no dispositivo endereçado e devolve uma mensagem de interrupção normal. Quando CC = 1, o microcódigo S/370 liberta a interrupção da tabela PIT, coloca uma CSW no sítio próprio, na memória S/370, ajusta o registador de condição para 1 e passa à instrução sequencial seguinte. Para a terceira opção, CC igual a 3, o microcódigo S/370 apenas ajusta o registador de condição igual a 3 e passa à instrução sequencial seguinte.
4. HALT I/O (fig. 44C) - Neste nível de descrição, a função para HALT I/O é idêntica à função para HALT DEVICE.
5. RESUME I/O (fig. 44D) - Num sistema S/370, esta instrução RIO apenas faz uma verificação para ver se o canal está operacional antes de aceitar a instrução. O microcódigo S/370 tem de verificar o CC para o CUA específico, como com outras instruções I/O. A CAW não é referenciada, e a CAW não é buscada para esta instrução.

Quando o microcódigo S/370 recebe o controlo do despacho S/370 para uma instrução de reassumir I/O (RIO), ele verifica o código de condição (CC) para a entrada de estado do dispositivo endereçada. Há duas opções. CC



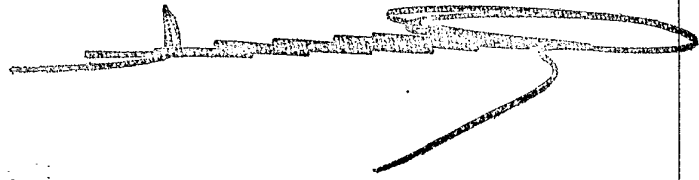
é igual a 0, 1 ou 2 e CC é igual a 3. Para CC igual a 0, 1 ou 2, o microcódigo S/370 emite a mensagem RESUME I/O para EXEC370, ajusta o código de condição para 2 e ajusta o registrador de condição para 0 e passa à instrução sequencial seguinte. Entretanto, quando EXEC370 recebe a mensagem RIO, ele examinará o endereço da unidade de controlo e continua a operação I/O antes suspensa. Para a segunda opção, CC é igual a 3 - o microcódigo S/370 simplesmente ajusta o registrador de condição para 3 e passa à instrução sequencial seguinte.

6. START I/O (fig. 44E) - Quando o microcódigo S/370 recebe o controlo do despacho para uma instrução de iniciar I/O (SIO), ele usa o endereço da unidade de controlo para encontrar a entrada na tabela de estados dos dispositivos. Verifica então o código de condição (CC), havendo quatro opções. CC igual a 0, CC igual a 1, CC igual a 2 e CC igual a 3. Para CC igual a 0, o dispositivo está pronto e o microcódigo S/370 envia uma mensagem de início de I/O (SIO MSG) para EXEC370, ajusta CC para o valor 2, que significa ocupado, ajusta o registrador de condição para 0, significando a aceitação e passa à instrução sequencial seguinte. Entretanto, quando EXEC370 recebe a mensagem SIO, utiliza o endereço da unidade de controlo para encontrar o dispositivo específico e inicia uma operação I/O normal nesse dispositivo. Para a segunda opção, CC igual a 1, o microcódigo S/370 liberta a interrupção, coloca a CSW na memória S/370, ajusta o bit de ocupação na CSW para "ligado", ajusta o registrador de condição para 1 e passa à instrução sequencial seguinte. Para a terceira opção, CC igual a 2, o microcódigo S/370 ajusta a CSW e posição 40X da memória S/370 para "tudo zeros", coloca o bit de ocupação da CSW em ligação, ajusta o registrador de condição para 1 e passa à instrução sequencial seguinte. Para a quarta opção, CC igual a 3, o microcódigo S/370 simplesmente ajusta o registrador de condição (CR) para 3 (significando dispositivo não operacional) e passa à instrução sequencial seguinte.
7. START I/O FAST RELEASE (fig. 44F) - Quando o microcódigo S/370 recebe o controlo do despacho para uma instrução de

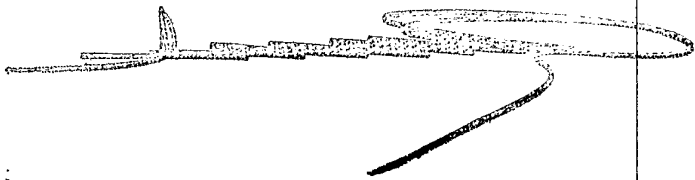


início de I/O rápido (SIOF), ele verifica o código de condição (CC) para a entrada DST endereçada. Há duas opções, CC igual a 0, 1 ou 2 e CC igual a 3. Para a primeira opção, CC igual a 0, 1 ou 2, o microcódigo S/370 envia uma mensagem SIOF para EXEC370, ajusta CC igual a 2, o registador de condição para 0 e passa à instrução sequencial seguinte. Entretanto, quando EXEC370 recebe uma mensagem SIOF, se estiver apto, inicia a operação I/O; caso contrário devolve uma mensagem de interrupção com uma CSW contendo um código de condição de deferimento que actua como uma interrupção normal quando é recebida pelo microcódigo S/370. Para a segunda opção, código de condição igual a 3, o microcódigo S/370 simplesmente ajusta o registador de condição para 3 e passa à instrução sequencial seguinte.

8. TEST I/O (fig. 44G) - Quando o microcódigo S/370 recebe o controlo do despacho para uma instrução de ensaio I/O (TIO), ele verifica o código de condição. Há três opções, CC igual a 0 ou 3, CC igual a 1 ou CC igual a 2. Para CC igual a 0 ou 3, o microcódigo ajusta o registador de condição para o valor CC e passa à instrução sequencial seguinte. Para a segunda opção, CC igual a 1, o microcódigo liberta a interrupção e coloca a CSW na memória S/370, ajusta o registador de condição para 1, significando CSW armazenada, e passa à instrução sequencial seguinte. Para a terceira opção, CC igual a 2, o microcódigo põe zeros na área CSW (40X) na memória S/370, ajusta para "ligação" o bit de ocupado na CSW, ajusta o registador de condição para 1 e passa à instrução sequencial seguinte.
9. STORE CHANNEL ID (fig. 44H) - Quando o microcódigo S/370 recebe o controlo do despacho para uma instrução de Store Channel ID, ele verifica o endereço do canal. Há duas opções endereço do canal válido e endereço do canal inválido. Para a opção canal inválido, o microcódigo ajusta o registador de condição CR igual a 3 e passa à instrução sequencial seguinte. Para a opção endereço do canal válido, o microcódigo ajusta a posição de memória S/370 A8 hexadecimal para 20000000 hexadecimal. Ajusta depois o registador de condição para 0 e passa à instrução sequencial seguinte.

- 
10. TEST CHANNEL (fig. 40I) - Quando o microcódigo S/370 recebe controlo do despacho para uma instrução de ensaio do canal (TCH) ele verifica o endereço do canal. Notar para este fluxograma que há duas opções principais e três opções pouco importantes. Para a primeira opção principal, endereço do canal inválido, o microcódigo ajusta o registador de condição para 3 e passa à instrução sequencial seguinte. Para a segunda opção, endereço do canal válido, o microcódigo verifica ainda todas as entradas da tabela DST para este canal. A primeira das opções menos importantes ocorre se o microcódigo descobrir uma entrada DST para um dispositivo específico com CC igual a 1, significando que este dispositivo tem uma interrupção pendente. Para este caso, o microcódigo ajusta o registador de condição igual a 1 e passa à instrução sequencial seguinte. Se, quando o microcódigo chega ao fundo da lista de entradas DST para este canal, não tiver encontrado uma entrada para a qual CC é igual a 1, ele então verifica se há pelo menos uma com CC igual a 2. Se houver, esta é uma segunda das opções menos importantes e, para esse caso, o microcódigo ajusta o registador de condição para o valor 2 e passa à instrução sequencial seguinte. Caso contrário ocorre a terceira opção pouco importante e o microcódigo ajusta o registador de condição para o valor 0 e passa à instrução sequencial seguinte.
11. Interrupções primárias e secundárias (fig. 44J, 44K) - Os termos interrupções primárias e secundárias são termos do S/370. Uma interrupção primária contém pelo menos o bit de estado Channel End (CE) na palavra CSW resultante de uma operação I/O. Uma interrupção secundária é ou uma segunda interrupção, contendo o Device END (DE) para a operação I/O, ou é uma interrupção assíncrona, iniciada pelo serviço que pediu o dispositivo.

Ao nível desta descrição, não há diferença entre interrupções primárias e secundárias; portanto, descreve-se apenas a interrupção primária. A diferença entre as interrupções I/O mascaradas e I/O activadas das fig. 44J e 44K é se I/O está mascarado. Isto é, se o processador



S/370 aceitará uma interrupção proveniente do canal ou não. Se uma interrupção não for aceite pelo processador S/370, o canal empilha a interrupção e é designada por interrupção pendente até ao momento em que o processador S/370 esteja permissivo. Quando se verificar uma condição de interrupção enquanto EXEC370 estiver a emular uma operação de um dispositivo específico, ele forma uma palavra CSW e armazena-a numa mensagem, que depois envia para o microcódigo S/370. Quando o microcódigo recebe esta mensagem de interrupção ele verifica a máscara S/370 para verificar se I/O está mascarado ou permissivo. Se I/O estiver mascarado (fig. 44J) ele empilha a interrupção. A descrição do processo de empilhamento de interrupções é apresentado mais adiante. Se o microcódigo S/370 verificar a máscara e I/O estiver mascarado (fig. 44K) ajusta-se o campo do código de condição na entrada DST para o dispositivo que faz a interrupção igual à condição seguinte (NC) na mensagem de interrupção, a CSW da mensagem é colocada na memória S/370 e o microcódigo provoca a execução de uma interrupção I/O.

12. Acontecimentos de Mascaramento I/O S/370 (fig. 44L) - Se I/O estiver mascarado quando EXEC370 envia uma mensagem de interrupção para o microcódigo S/370, a interrupção é colocada na pilha numa entrada da tabela de interrupções pendentes (PIT). Num instante ulterior, ocorrerá algum acontecimento S/370, de que resultará a activação das interrupções I/O, ou qualquer interrupção para a qual a máscara activa I/O. Em algum instante em que a máscara do sistema PSW for alterada de modo que active um I/O anteriormente mascarado, o microcódigo S/370 tem de verificar a existência de quaisquer interrupções pendentes para esses canais. Se não for encontrada nenhuma, o microcódigo simplesmente sai para a instrução sequencial seguinte. Se no entanto for encontrada, o microcódigo liberta a interrupção da tabela, coloca CSW na memória S/370 e efectua uma interrupção S/370.

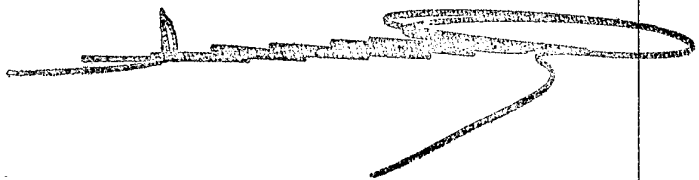
No que segue, estão contidas descrições dos processos que foram referenciados imediatamente atrás:

1. Interrupção empilhada - O termo interrupção empilhada é usado em ligação com mensagens de interrupção que são recebi-



das pelo microcódigo S/370 quando I/O S/370 for mascarado. As interrupções são empilhadas na área de estados dos dispositivos, no que é denominado tabela de interrupções pendentes, ou PIT. As entradas da tabela PIT são encadeadas por ordem FIFO (atendimento por ordem de chegada) para a entrada DST que representa o dispositivo S/370 que provoca a interrupção. Empilhar uma interrupção implica tomar uma entrada da PIT da lista livre, encadeá-la para o fim da lista da PIT para essa entrada DST, colocar a CSW no campo de estados da entrada PIT e o valor NC no campo NC da entrada PIT, e colocar o campo CCW da DST com o valor "1". A colocação de CC com o valor "1" indica que existe uma interrupção pendente para este dispositivo.

2. Libertar a interrupção - Libertar uma interrupção implica retirar da cadeia a entrada PIT no topo da lista DST/PIT, colocar o código de condição DST com o valor encontrado no campo NC da entrada PIT, preservar o campo do estado da entrada PIT que contém uma CSW S/370 e devolver a entrada PIT para a lista livre.
3. Enviar mensagem para EXEC370. Pode fazer-se referência à fig. 43 para esta descrição, a título de exemplo. No instante da opção CC igual a 0, o microcódigo S/370 decidiu que necessita de enviar uma mensagem para EXEC370. Especificamente, a mensagem é uma mensagem START I/O. (SIO MSG). Para esta mensagem, ou qualquer outro tipo de mensagem que o microcódigo S/370 envie, o procedimento é o mesmo. O microcódigo S/370 preenche o campo de dados na entrada da caixa de correio na memória (162) com o conteúdo da mensagem. Envia depois um pedido PU para a BCU, que é recebido pela lógica BCU (253). O microcódigo S/370 então espera a devolução de uma confirmação. Entretanto, a lógica BCU quando recebe uma indicação ZU para BCU inicia um acesso à memória e uma operação do DMA para transferir os dados da caixa de correio para a memória (210) da BCU. Quando estiver completo o DMA, devolve um sinal de confirmação (ACK) para o microcódigo S/370, que prossegue então com a sua instrução sequencial do programa seguinte. Ao mesmo tempo, a lógica DMAC interrompe o sistema 88. A rotina dos meios de programação rece-



be o controlo, verifica a validade da operação e depois emite uma notificação para EXEC370 que retira então a mensagem da fila de espera de trabalho.

4. Enviar mensagem para o microcódigo S/370 - Há vários tipos diferentes de mensagens que o EXEC370 envia para o microcódigo S/370. S/170 I/O Masking Events (fig. 44L) é um exemplo de uma tal mensagem de interrupção. EXEC370 chama o microcódigo ETIO, que forma a interface com a lógica BCU. ETIO inicia uma operação DMA, que transfere a mensagem da memória BCU (210) para a memória S/370. Quando DMA estiver completo, é enviada uma mensagem DCU para PU, para o microcódigo S/370 e é enviada uma interrupção para o sistema 88, que faz com que a rotina da interface ETIO envie uma notificação para EXEC370.

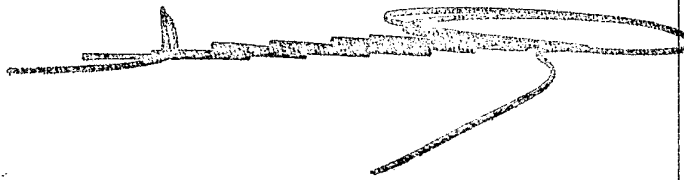
FUNCIONAMENTO DA UNIDADE DE CONTROLO DA LINHA OMNIBUS (BCU) (156)

1. INTRODUÇÃO

Vão resumir-se alguns dos componentes do sistema e as suas funções atrás descritos. A BCU (156) desempenha uma função de interface I/O entre o conjunto de micropastilhas S/370 e o subsistema I/O, que é constituído pelo PE S/88 (62) e o seu sistema associado e componentes I/O no módulo (10). O conjunto (150) de micropastilhas S/370 e o subsistema I/O comunicam através do adaptador de linhas omnibus (154). A área (162) da memória S/370 no interior da memória principal S/99 (16) é aqui por vezes referida como módulo básico da memória BSM (162). Há dois conjuntos de linhas de interface das linhas omnibus do adaptador (249,250) (canal 0) e (251,252) (canal 1) que fazem o acoplamento da BCU (156) e do adaptador de linhas omnibus (154).

A BCU (156) inclui uma memória local de 64 KB (210), um controlador de acesso directo à memória DMAC) (209), uma linha omnibus de endereços local, de 32 bits (247), uma linha omnibus de dados local de 32 bits (233) e uma lógica de interface (205).

Como atrás se descreveu com mais porme-



nor, o DMAC (209) inclui quatro canais de transferência de dados:

Canal 0 - Os comandos da caixa de correio são transferidos do PE (85) para a BCU (156). São lidas mensagens da área de memória S/370 (162) para a memória local (210). Dados de escrita PE S/370 (85).

Canal 1 - Os dados são lidos da área de memória S/370 (62) para transferir para a memória local (210).

Canal 2 - Dados de leitura PE S/370 (85). Os dados são transferidos da memória local (210) para a área de memória S/370 (162).

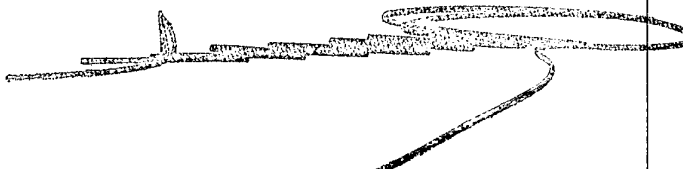
Canal 3 - Transferências de mensagens de prioridade elevada da BCU (156) para o PE S/370 (85). As mensagens são transferidas da memória local (210) para a área de memória S/370 (162).

O controlador DMAC (209) transfere palavras duplas (32 bits) entre o adaptador de linhas omnibus (154) e a memória local (210). Ele também interrompe o subsistema I/O S/88 PE (62) quando estiverem completas as transferências de dados I/O. A memória local (210) inclui memórias tampão de I/O e de dados de mensagens WQB e dados da lista de elos de ligação, para cargas automáticas da caixa de correio através do DMAC (209).

A lógica BCU (205) inclui uma unidade de arbitragem de linhas omnibus locais (216) na qual o PE S/88 (62) e o DMAC (209) disputam o acesso às linhas omnibus locais, isto é, a linha omnibus de dados (223) e a linha omnibus de endereços (247). A linha "Bus Request" (pedido de linha omnibus) PE (62) (190) está activa sempre que são detectados os endereços seguintes (ver fig. 41C) pela unidade (216) de descodificação de endereços e arbitragem:

Qualquer endereço da memória local; qualquer comando dirigido à BCU, incluindo Programmed BCU Reset, BSM Write Select Up, BSM Read Select Up e Read BCU status; ciclo de confirmação de Local bus interrupt e qualquer comando de leitura ou escrita de registador dirigido ao DMAC.

A linha DMAC Bus Request (269) está activa quando pretende obter o controlo da linha omnibus local



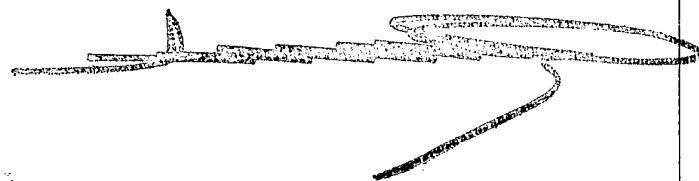
(223,247) para uma sequência DMAC (leitura ou escrita da memória local (210)) ou uma sequência de carga da lista de elos de ligação (leitura da memória local). A linha (288) de aceitação pela linha omnibus é activada quando for dado o controlo da linha omnibus local ao DMAC (209) pela lógica (216); a linha (191) é activada se for dado o controlo ao PE (62).

A lógica BCU (205) controla a distribuição de tempos das transferências do DMAC (209) entre o adaptador de linhas omnibus (154) e o subsistema I/O e converte transferências I/O até 4 KB em transferências de blocos de 64 octetos para o adaptador de linhas omnibus (154), pelos canais 0 e 1.

A lógica BCU (205) detecta um atravessamento de uma fronteira de 64 octetos para qualquer transferência de blocos. Se isso ocorrer, o bloco será partido em duas transferências separadas. A BCU (156) calculará o número de palavras até à fronteira dos 64 octetos para a primeira transferência. Isso será apresentado, juntamente com o endereço inicial, ao adaptador de linhas omnibus (154). As palavras restantes, juntamente com um novo endereço, serão apresentadas ao adaptador das linhas omnibus (154) através de um comando subsequente (BSM read/BSM write). A lógica (BCU) (205) também proporciona um pré-esvaziamento (pre-empt) de transferências de dados I/O (numa fronteira de 64 octetos) quando se verificar um pedido de mensagem de alta prioridade ou um pedido de leitura da caixa de correio. Um pedido de mensagem de alta prioridade e um pedido de caixa do correio podem ser tratados concorrentemente na BCU (156). Uma operação "BSM Read" e "BSM Write" podem ser tratadas concorrentemente na BCU (156).

A BCU (156) realiza as seguintes operações I/O:

Operação Mailbox Read : iniciada pelo S/370 I/O INSTRUCTION MICROCODE através da linha (256a) "PU to BCU REQ". A caixa de correio (188) está localizada no módulo BSM S/370 (162). Ela é utilizada para armazenar comandos I/O que serão executados pelo subsistema I/O (Start I/O, etc.). Podem também conter informação de estados, e outra, que o subsistema I/O recebe do PE (85). Um comando "PU to BCU Select UP" é inicia



do pela BCU (156) quando a linha (210) "PU do PCU Select" for activada no canal 0 da linha omnibus do adaptador. Operações de escrita I/O S/370 (canal 0 da linha omnibus do adaptador) serão pré-esvaziadas numa fronteira de 64 octetos se for activado "PU do BCU Request" pelo PE S/370 (85).

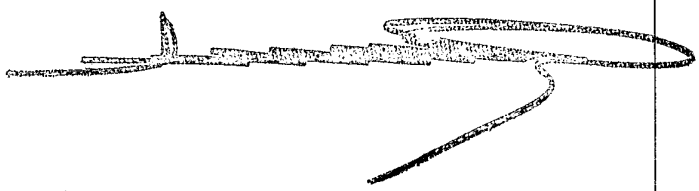
Operações de leitura e escrita I/O S/370: proporcionam transferências de dados (máximo blocos de 4 KB) entre a memória S/370 (162) e os dispositivos I/O nos canais 0 e 1 na linha omnibus do adaptador. Todas as transferências de dados são iniciados pelo subsistema I/O (PE S/88 (62)) através do comando da linha omnibus do adaptador "BSM SELECT UP".

Transferências de mensagens de elevada prioridade: interrupções, mensagens de estados, erros, etc., mensagens de elevada prioridade são passadas do subsistema I/O para o S/370. Todas as transferências são iniciadas a partir da BCU (156), através do comando "Q SELECT UP". As operações de leitura I/O S/370 (canal 1 da linha omnibus do adaptador) serão pré-esvaziadas num limite de 64 octetos se se verificar um pedido de mensagem de elevada prioridade.

2. FLUXO DA SEQUÊNCIA START I/O S/370. DESCRIÇÃO GERAL E PORMENORIZADA

A instrução "START I/O (SIO), a "Palavra de Endereço do canal" (CAW) e a primeira "Palavra de Controlo do Canal" (CCW) estão armazenadas em posições pré-determinadas da "caixa de correio" na memória S/370 (162). Esta informação é passada para a memória local (210) através da lógica de interface BCU (205) e do adaptador de linhas omnibus (154).

Os registadores do canal 0 do DMAC representados na fig. 18 são usados para as operações de leitura da caixa de correio. Eles serão programados pelo PE S/88 (62) para operar num "Modo de Encadeamento de Matrizes Encadeadas". O PE (62) inicializa este modo estabelecendo uma série de "listas encadeadas" (tabelas) na memória local (210), fig. 41H. Depois ajustará o primeiro "endereço superior da lista encadeada", no "Registador de endereços de base" (BAR) do canal 0 do DMAC



(de 32 bits). O endereço aponta para a primeira posição na memória (210) dos dados da lista encadeada.

A "PCL" (linha de controlo de periféricos) do DMAC (257a) será programada pelo PE (62) para fazer com que o DMAC (209) active a sua linha (258) de saída de interrupção IRQ sempre que seja activada a linha PCL (257a). A linha "PCL" (257a) será activada a seguir a ter-se completado a transferência de dados da caixa de correio da memória principal (162) para a memória local (210) através da memória tampão do adaptador (259). A interrupção informará o processador PE S/88 (62) de que precisamente se completou uma carga da caixa de correio.

Os dados da lista de elos de ligação (fig. 41H) consistem no seguinte: o endereço de memória inicial de um bloco de dados; a contagem de transferência de memória e um endereço do elo de ligação para a entrada seguinte da tabela. O endereço do último elo de ligação na tabela será zero.

O processador S/88 (162) ajusta o endereço superior da lista encadeada no registador de endereços de base do canal 0 do DMAC.

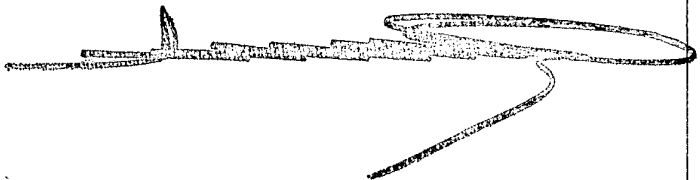
O processador S/88 PE (62) activará o DMAC (209) escrevendo um "1" no bit 7 (bit "Start") do seu registador de controlo de canal (CCR) do seu canal 0. O DMAC (209) lerá então a primeira lista encadeada para o seu registador do canal 0 da seguinte maneira:

Endereço inicial do bloco de dados WQB da memória (210) para o registador de endereço de memória (MAR) (32 bits);

Contagem de transferências (octetos de dados da caixa de correio) para o registador de contagem de transferências da memória (MTC); e

Endereço do elo de ligação para o registador de endereço de blocos (BAR) seguinte.

Mais especificamente, durante a execução da instrução, o PE S/370 (85) descodifica a instrução "START I/O", coloca o comando "START I/O", a Palavra de Endereço do Canal e a primeira palavra de comando do canal em posições sucessivas da "mala de correio", contidas na memória S/370 (162). O endereço inicial da caixa de correio (base + comprimento da



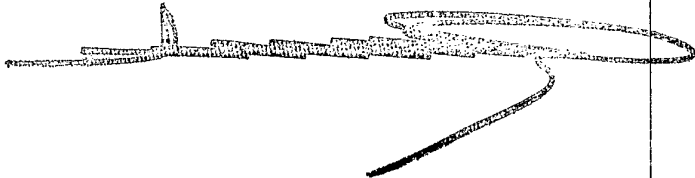
fila de espera) é armazenado no registador de base do adaptador das linhas omnibus (154), no instante da inicialização.

O PE S/370 (85) envia um controlo operativo "LD OSCW" através da linha omnibus do processador, com o bit 11 activo. Isso ajusta para o estado ligado o bit "PU to BCU REQUEST" na palavra de controlo do adaptador das linhas omnibus (154). O bit 11 da palavra OSCW provoca um pedido "PU to BCU Request" na linha omnibus do adaptador (canal 0). Se ocorrer um "PU to BCU REQ" durante uma transferência de dados I/O, então a BCU (156) pré-esvaziará a transferência I/O num limite de 64 octetos para permitir que se verifique uma carga da caixa de correio.

A BCU (156) gera então na linha omnibus (290) um comando "Read Mailbox Select Up", com o formato representado na fig. 45A, onde os bits 0 e 1 são bits de comando e os bits 2-7 são a contagem de octetos, e armazena este no registador de comandos, canal 0, (214). Os bits do endereço da caixa de correio são armazenados no registador (219) através da linha omnibus (290) num formato representado na fig. 45B, onde o bit 7 identifica a área IOA na memória (162), os bits 24-26, o número do canal da BCU e os bits 27-31, o desvio da mala de correio.

Depois a BCU (156) activa a linha omnibus COMMAND/STATUS (249) e a linha omnibus ADDR/DATA (250), preenchendo os registadores (214) e (219), activa um comando "TAG UP" na linha (262a) e espera pelos dados do adaptador da linha omnibus. Ela faz isso por exploração da linha "TAG DOWN" (262b). "TAG DOWN" está activo enquanto os dados não estiverem prontos. Logo que "TAG DOWN" tenha sido desactivado pelo adaptador de linhas omnibus (154) (dados prontos), retêm-se os quatro primeiros "octetos" de dados da caixa de correio na memória tampão de leitura, do canal 0, (226) através de dois subciclos do canal 0.

A lógica da BCU (253) activa depois a linha "REQUEST" (263a) no canal 0 do DMAC (209). O DMAC (209) activa depois "BUS REQUEST" (BR) para a linha (269) para o circuito (216) de arbitragem da LINHA OMNIBUS LOCAL. Se a linha omnibus local não estiver a ser usada pelo processador S/88 (62), é concedido o acesso através da linha de concessão de acesso (BG) (268) ao DMAC (209). O DMAC (209) transfere então o endere



go inicial da caixa de correio local WQB (na memória (210) do MAR para a linha omnibus de endereços (247), e activa "ACKO" (confirmação do canal 0 do DMAC), linha (264a). O sinal "ACKO" inicia a transferência dos dados da memória tampão (226), através da linha omnibus de dados (223) para a parte da Caixa de Correio Local da WQB na memória (210). A linha "DTACK" (265) é activada para informar o DMAC (209) de que está completa a operação.

Os sinais de relógio da BCU (fig. 25) continuam a transferir dados da caixa de correio da memória tampão (259) para o registador (226). A BCU (156) executa duas sequências (de 16 bits cada) da linha omnibus do adaptador ("TAG UP"/"TAG DOWN") para cada sequência (32 bits) da memória local (210)/DMAC(209).

Quando o ciclo DMAC estiver completo (DTACK activo), o DMAC (209) activa a linha (267) "Data Transfer Complete" (DTC) para a lógica BCU (253), que envia outro "REQUEST" para o DMAC (209) pela linha (263a) para ler os segundos quatro octetos do registador (226) para a caixa de correio WQB. Os ciclos DMAC repetem-se, até terem sido transferidos todos os dados da caixa de correio (16 octetos) (4 ciclos da linha omnibus) local. A linha "PCL" (257a) será então activada pela lógica BCU (253) para o DMAC (209). Isso faz com que a linha "IRQ" (258) seja activada a partir do DMAC (209) para o codificador de prioridades do processador S/88/lógica de interrupções (212). O PE (62) tratará então o pedido de caixa de correio.

Quando o DMAC (209) completar a sua carga do registador do canal 0 a partir da lista encadeada, ele espera então um sinal na linha "REQ" (263a) do canal 0, a partir da lógica BCU (253), para iniciar a carga seguinte da caixa de correio. Uma vez que tenha arrancado, o canal 0 do DMAC mantém-se activo indefinidamente, controlando o processador S/88 (62) a lista encadeada circular e suspendendo a BCU (156) as transferências de dados, mantendo a linha "REQ" (263a) inactiva. Se o canal 0 se interromper, devido a uma condição de "fim de lista", o processador S/88 receberá uma interrupção de terminação e fará de novo arrancar o canal 0, quando for apropriado.

3. FLUXO DA SEQUÊNCIA DE TRANSFERÊNCIA DE DADOS I/O S/370.
DESCRIÇÃO GERAL

Todas as transferências de leitura e escrita I/O originam a partir do processador S/88 (62), através da linha omnibus do adaptador, comandos "BSM READ SELECT UP" e "BSM WRITE SELECT UP" arquitetados. O comando CCW S/370 e o endereço inicial (na memória S/370 (162)) são derivados da palavra CCW para um "START I/O". Os dados são transmitidos pelo processador S/88 (62) entre este dispositivo I/O e uma memória tampão local, na memória local (210).

A memória local (210) inclui uma fila de espera de blocos de memória para operações de escrita I/O, que é gerida pelo processador S/88 (62). Quando a fila de espera incluir pelo menos uma entrada, ela está pronta para desencadear uma operação de escrita I/O. O endereço inicial para um dos blocos selecionado é armazenado nos registradores do canal 1 DMA, no controlador DMAC (209), pelo processador S/88 (62), antes da iniciação de uma operação de escrita. Os registradores do canal 1 DMAC são reservados para operações de escrita de I/O S/370 (memória S/370 (162) para I/O) através da memória local (210). A memória tampão de dados do adaptador (259) (64 octetos) é reservada para leitura da caixa de correio e operações de escrita de I/O S/370 (transferências de dados da memória S/370 (162) para a memória local (210)). Esta memória tampão está associada ao canal 0 das linhas omnibus (249,250) do adaptador. A memória tampão (260) (64 octetos) é reservada para escrever mensagens (para S/370) e para operações de leitura I/O S/370 (transferências de dados da memória local (210) para a memória S/370 (162)). Esta memória tampão está associada ao canal (1) (251, 252) da linha omnibus do adaptador. O processador S/88 (62) inicializa os registradores dos endereços de memória das palavras de ordem superior do DMAC, canal 1 canal 2, para zero (0). Isso poupa um ciclo extraordinário da linha omnibus quando estes registradores são carregados durante sequências de operações, visto que a memória local (210) não requer mais de 16 bits de endereço.

a) Operações de escrita I/O: (Memória S/370 (162) para Memória local (210))

O processador S/88 (62) ajusta o endereço inicial da memória tampão local no registador MAR de endereços de memória do canal 1 do DMAC, colocando informação na linha omnibus (248) de endereços e dados DMAC (através da linha omnibus (161a), do excitador (217), da linha omnibus (247) e do retentor (233), como se mostra na fig. 45C, sendo os bits 31-08 = 007E00 = comando "DMAC REGISTER SELECT" e os bits 07-00 = Seleção (baixa) do registador de endereço de memória do canal 1 do DMAC. Notar que S/88 identifica bits mais e menos significativos na linha omnibus respectivamente como "31" e "0", o oposto do protocolo S/370.

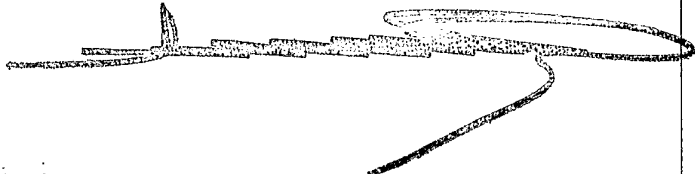
O conteúdo representado na fig. 45D (destinado ao MAR) é colocado na linha omnibus de dados (223), sendo os bits 31-16 = endereço inicial da memória tampão local na memória local (210) para os dados de escrita I/O. Os bits de ordem elevada na linha omnibus (31-36) serão carregados na parte de ordem inferior (15-00) do registador de endereços de memória do canal 1. Os bits de ordem elevada (31-16) do MAR foram ajustados para 0 durante a inicialização. O DMAC (209) responde com um sinal "DSACK" da porta de 16 bits nas linhas (266 a,b) através da lógica BCU (253) para a CPU do processador S/88. O processador S/88 (62) coloca os dados da BCU (contagem de octetos, chave da memória, dados da prioridade da linha omnibus do adaptador e dados do espaço utilizador (IOA)) e os dados de contagem de transferências de memória do canal 1 do DMAC pela linha omnibus de endereços local (247). A fig. 45E mostra o comando na linha omnibus de endereços, sendo os bits:

31-08 = 007E00 = comando "DMAC REGISTER SELECT"; e

07-00 = BCU SELECT e DMAC CHANNEL 1 MTC SELECT.

Os bits da contagem de octetos, da chave da memória (derivada da palavra CCW), a prioridade da linha omnibus do adaptador e do espaço utilizador) IOA serão colocados na linha omnibus de dados (223) pelo processador S/88 (62) no formato representado na fig. 45E, tendo os bits a designação seguinte:

- 
- 31-27 = Reservados
- 26 = Bit de ordem superior da contagem de octetos. Este bit será igual a 1 apenas quando estiver a transferir a contagem máxima de octetos (48 octetos).
- 26-16 = Contagem de octetos carregada no registrador MTC do canal 1 do DMAC
- 26-14 = Contagem de octetos carregados no registrador BCU (220) (máximo 4 096), sendo pelo menos parte da contagem carregada no registrador (221), como se descreverá mais adiante nas operações de contagem de octetos. O adaptador de linhas omnibus (154) requer uma contagem de 1111 1111 1111 para transferir 4 096 octetos (contagem de octetos - 1). Por conseguinte, a BCU (156) decrementará os bits dos limites de palavras duplas (26-16) uma vez, antes de a apresentar, juntamente com os bits 15-14 do desvio de octetos (em blocos de 64 octetos), ao adaptador da linha omnibus (154).
- 15-14 = Bits da contagem dos octetos de ordem inferior da BCU 156. Estes bits representam o desvio de octetos menos 1 (para os requisitos do adaptador da linha omnibus) a partir de um limite de palavras duplas. Estes bits não são usados pelo DMAC (209) nem pela BCU (156) visto que apenas transfere palavras duplas. São passados para o adaptador da linha omnibus (154), para apresentação ao BSM S/370 (162).
- 13-12 = prioridade dos canais da linha omnibus do adaptador
- 11-08 = chave da memória
- 07 = bit do espaço utilizador IOA
- 06 = O processador S/88 activará este bit (1) para indicar que é necessário um acesso de memória local adicional. Isso verifica-se quando um endereço inicial de memória S/370 não está num limite de palavras duplas (32 bits). Como todos os acessos BCU têm de começar num limite de palavras duplas,



o primeiro acesso conterá o ou os octetos designados com endereço inicial, bem como o ou os octetos anteriores contidos nesse endereço de palavras duplas. Rejeitam-se o ou os octetos anteriores.

05-00 = Reservados.

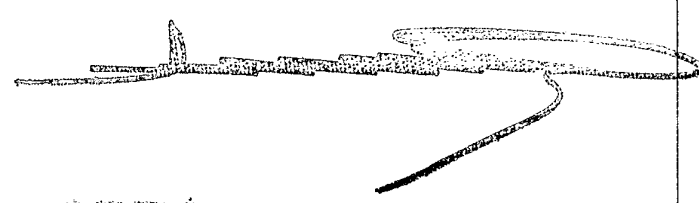
O DMAC (209) carregará a palavra de ordem superior (isto é, a contagem de octetos) da linha omnibus de dados no registrador MTC do canal 1. A BCU (156) capturará o conteúdo da linha omnibus de dados da seguinte maneira:

Bits 26-14 - para o contador (220) de octetos de BSM READ SELECT UP; e

Bits 13-06 - para o registrador (219) de endereços e dados do ADAPTER BUS CHANNEL 0, mas dispostos doutra maneira.

Para uma transferência de palavras duplas se verificar num ciclo de máquina do processador S/88, o endereço tem de estar num limite de palavras duplas. Como o endereço MTC do canal 1 do DMAC não está num limite de palavras duplas (bits 07-00 = 01001010), tem lugar a seguinte acção para carregar a BCU (156) e o DMAC (209) com um comando do processador S/88. A BCU (156) inverterá o bit 1 do endereço e apresentará o mesmo ao DMAC (209), juntamente com os outros bits de selecção do registrador. Isso permitirá que o registrador MTC para o canal 1 seja seleccionado apropriadamente (bits 07-00 do endereço = 01001010). Esta disposição aplica-se também à selecção do registrador MTC para as operações de leitura I/O do canal 2. O DMAC (209) responde com um sinal "DTACK" na linha (265) para a lógica BCU (253). A lógica BCU (253) converte o sinal "DTACK" numa resposta "DSACK" na porta de 32 bits pelas minhas (266 a,b) para o processador S/88 (62). A contagem de octetos de transferência, juntamente com os dados restantes da linha omnibus de dados serão apresentados ao adaptador (154) durante o comando "BSM READ SELECT UP" seguinte. O contador (21) de limites de leitura BSM ou o contador (220) de octetos BSM READ SELECT-UP serão carregados no registrador (214) de comandos de leitura do canal 0.

O processador S/88 (62) gerará então um comando "BSM READ SELECT UP" na linha omnibus (247) com o formato representado na fig. 45G, sendo os bits 31-00 = 007E0108 =



= BSM READ SELECT UP".

O processador S/88 (62) também colocará o endereço inicial na linha omnibus de dados (223) no formato representado na fig. 45H, na qual os bits 23-0 = endereço inicial na memória (162).

O endereço inicial BSM na linha omnibus (223) é armazenado no registrador A/D (219) e no registrador de endereços de leitura do BSM (231). Ele será depois emitido para o adaptador de linhas omnibus (154), para apresentação à memória S/370 (162). A BCU (156) activa depois as linhas "DSACK" (266 a,d) para o processador S/88 (62). Nesta altura, o processador S/88 é libertado e deixa de estar envolvido nesta operação.

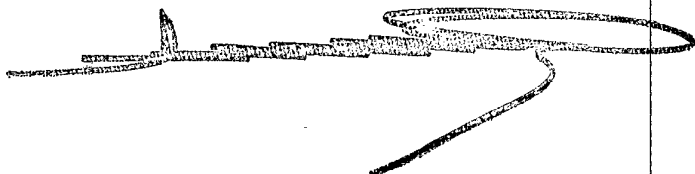
A BCU (156) coloca o comando (de leitura) "BSM SELECT UP" no registrador (214), através da linha omnibus (290), na linha omnibus (290), na linha omnibus de comando/estados (249) representada na fig. 45I, sendo os bits:

- 0-1 = 11 o comando (de leitura) "BSM SELECT UP" e
- 2-7 = comprimento do campo menos 1 (máximo 64 octetos)

O comprimento do campo foi previamente transferido do registrador (220) ou (221) para o registrador (214). O registrador (219) coloca a informação do endereço na linha omnibus (20) no formato representado na fig. J, sendo os bits:

- 0-3 = Chave da memória
- 4 = 1
- 5-6 = prioridade (adaptador de linhas omnibus (154) para linha omnibus do processador (170));
- 7 = 1 = acesso à área de utilizador;
0 = acesso à área de microcódigo;
- 8-31 = endereço do primeiro octeto no campo de dados na memória (162).

A lógica BCU (253) activa então a linha TAG UP (262a) para o adaptador de linhas omnibus (154), para reter os dados de comando e do comprimento do campo no registrador de comandos do adaptador (124) (fig. 13) e os dados de endereço no registrador (122). O adaptador de linhas omnibus (154) activa TAG DOWN para a lógica BCU (253), se os dados não forem válidos. A lógica BCU (253) espera até desaparecer TAG DOWN. O adaptador de linhas omnibus (154) converte o comando BSM SELECT UP na li-



nha omnibus do adaptador num comando de memória I/O da linha omnibus do processador, como se mostra nas fig. 45K e 45L, no qual os bits da linha omnibus (170) de dados/endereços do processador representam:

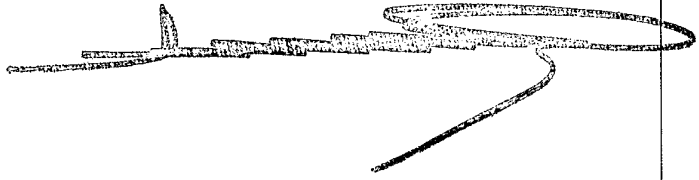
- 0 = 0 = Comando de memória I/O
- 1 = 1 = Operação de busca
- 2-7 = Comprimento do campo
- 8-31 = Endereço real do octeto

e no qual os bits da linha omnibus de chave/estados do processador representam:

- 0-3 = Chave da memória
- 4 = 0 Ausência de translação dinâmica.

Quando os dados endereçados são devolvidos da memória S/370 (162), eles são retidos na memória tampão (259) (canal 0) de dados do adaptador de linhas omnibus. O adaptador de linhas omnibus (154) desactiva então a linha TAG DOWN (262b) no canal 0 da linha omnibus do adaptador. Esta condição alerta a BCU (156) para reter dois octetos (16 bits) de dados, imediatamente seguidos de outros dois octetos na memória tampão de leitura do canal 0 (4 octetos) através de sinais de relógio da esquerda e da direita. A BCU (156) activa então a sua linha "REQ1" (263b) (pedido do canal 1 do DMAC) para o DMAC (209). O DMAC (209) envia um sinal "BUS REQ" pela linha (269) para a lógica (216) de arbitragem das linhas omnibus locais da BCU, para executar um ciclo de linha omnibus local.

Quando o sinal de aceitação da linha omnibus na linha (268) for devolvido da lógica (216) de arbitragem BCU, o DMAC (209) faz arrancar uma operação da memória tampão de leitura do canal 0 (259) para a memória local (210). Ele faz isso devolvendo ACK1 (confirmação do canal 1 do DMA) pela linha (264b) para a lógica BCU (253), e encaminhando o endereço de memória local para o registador do canal 1 DMAC (MAR) para os circuitos (não representados) de endereçamento da memória (210), através da linha omnibus (248), do retentor (233) da linha omnibus de endereços (247) e do multiplexador (232). A lógica BCU (253) utiliza o sinal ACK1 na linha (264b) e o sinal de selecção da memória RAM na linha (210a) para encaminhar os primeiros dados (4 octetos) da memória tampão (226) para a linha



omnibus de dados (223), para o armazenamento na memória (210) no endereço especificado pelo registador MAR. Quando DTACK for devolvido pela linha (265) pela lógica BCU (253), o DMAC (209) activa DTC (transferência de dados completa) na linha (267).

A BCU (156) decrementará a contagem de octetos que fora retida nos registadores (220) e (MTC); incrementa o registador MAR do canal 1 e decrementa o registador de endereços (231) para cada palavra dupla (4 octetos) de dados que foram recebidos do adaptador de linhas omnibus (154) até 64 octetos. A sequência descrita atrás é repetida para cada quatro octetos (até 64) do comando BCU. Se a contagem de transferências de octetos for maior que sessenta e quatro, então a BCU (156) apresentará um novo endereço inicial BSM ao adaptador de linhas omnibus (154) através dos registadores (231) e (219), a fim de buscar os 64 octetos seguintes. O registador (231) foi decrementado para cada transferência de quatro octetos, como se descreveu atrás e tem portanto o endereço inicial seguinte apropriado. O adaptador de linhas omnibus (154) regista em memória tampão 64 octetos de dados, para cada endereço inicial até se completar a transferência de dados pedidos pelo comando (até 4 KB).

A BCU (156) deixará o DMAC (209) em repouso (não excitando) REQ se a memória tampão (259) do adaptador de linhas omnibus estiver vazia e até ser recebida a palavra de dados válida seguinte; o estado de TAG DOWN reflecte a disponibilidade de dados válidos na memória (259). Os ciclos REQ/ACK continuam até a contagem de octetos chegar a zero, altura em que o DMAC (209) activa IRQ na linha (258) para o processador S/88 (62). Isto alerta o processador S/88 (62) para ler a memória tampão da memória local que contém os dados lidos da memória S/370 (162) para processamento apropriado:

- b) Operação de leitura I/O: (Memória local (210) para memória S/370 (162))

As operações de leitura I/O (sob o controlo do programa EXEC370) são desencadeadas quando existir pelo menos uma entrada na fila de espera de leitura I/O na memória (210). O processador S/88 (62) obtém o controlo da linha om



nibus local se ela não estiver a ser utilizada pelo DMAC (209). O processador S/88 (62) ajusta-se ao endereço inicial de leitura I/O da memória local no registador de endereços de memória (MAR) do canal 2 do DMAC, colocando a informação representada na fig. 45M na linha omnibus (247), onde os bits:

31-08 = 007E00 = Comando DMAC REGISTER SELECT

07-00 = Seleccionar o registador (inferior) de endereços da memória do canal 2 do DMAC;

e colocando o endereço inicial (da memória tampão na memória (210)) na linha omnibus de dados (223), como se mostra na fig. 45N, sendo os bits:

31-16 = endereço inicial dos dados de leitura I/O da memória tampão

15-00 = Reservados

Os bits de ordem elevada da linha omnibus de dados 31-36 serão carregados nos bits de ordem baixa (15-00) do registador de endereços de memória do canal 2. Os bits de ordem elevada (31-16) do registador MAR foram postos em zero durante a inicialização. O DMAC (209) responde com um sinal DTACK na linha (265), que é convertido em sinais DSACK nas linhas (266 a,b) para o processador S/88 (62). O processador S/88 (62) desloca então dados (até 4 KB) de um controlador I/O tal como (20) ou (24) para a memória local (210), através do controlo de programas S/88, utilizando o endereço inicial da memória tampão de leitura I/O da memória local.

Quando estiver completa a transferência de dados, o processador S/88 (62) coloca a selecção de contagem de transferência de memória do canal 2 do DMAC na linha omnibus de endereços (247) no formato representado na fig. 45O, sendo os bits:

31-08 = 007E00 = comando DMAC Register Select

07-00 = Seleccionar MTC do canal 2 do DMAC e BCU.

Os bits da contagem de octetos, da chave da memória (derivados da palavra CCW), da prioridade da linha omnibus do adaptador e do espaço do utilizador/IOA serão colocados na linha omnibus (223) pelo processador S/88 (62) no formato representado na fig. 45P, no qual os bits:

31-27 = Reservados

26 = bit de ordem superior da contagem de octetos. Este bit será igual a 1 apenas quando se estiver a transferir a contagem máxima de octetos.

26-16 = Contagem de octetos do registador MTC do canal 2 do DMAC

26-14 = Contagem de octetos carregada na BCU (156) (máximo 4 096).

O adaptador da linha omnibus (154) exige uma contagem 1111 1111 1111 para transferir 4096 octetos (contagem de octetos - 1). Por conseguinte, a BCU decrementará os bits 26-16 do limite de palavra dupla, uma vez, antes de os apresentar, juntamente com os bits 15-14 do desvio do octeto (em blocos de 64 octetos) para o adaptador de linhas omnibus (154).

15-14 = Bits de ordem inferior da contagem de octetos. Estes bits representam o desvio de octetos, menos 1 (para os requisitos do adaptador de linha omnibus) a partir de um limite de palavra dupla (32 bits). Estes bits não são usados pelo DMAC (209) ou pela unidade BCU (156) visto que apenas transferem palavras duplas. Os bits são passados para o adaptador de linhas omnibus (154) para apresentação ao BSM S/370 (162).

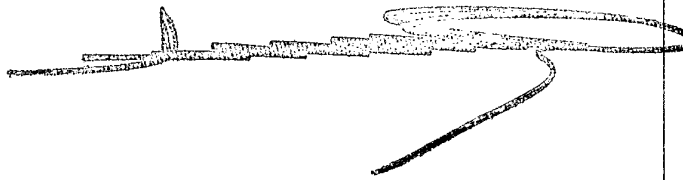
13-12 = prioridade dos canais da linha omnibus do adaptador

11-08 = chave da memória

07 = bit do espaço utilizador/IOA

06-00 = Reservados.

O DMAC (209) carregará a contagem de octetos da linha omnibus de dados (223) para o registador MTC do canal 2. A BCU (156) capturará o conteúdo da linha omnibus de dados quando o comando anterior aparece na linha omnibus de endereços (247). Os bits 26-16 são armazenados no contador (222) de octetos de selecção de escrita BSM. Os bits 13-07 são armazenados no octeto de ordem superior do registador (226) S/D do canal 1 da linha omnibus do adaptador. O DMAC responde com um sinal DTACK na linha (265) para a lógica BCU (253). A lógica (253) converte o sinal DTACK numa resposta DSACK na porta de 32 bits pe-



las linhas 266 a,b para o processador S/88 (62). A contagem da transferência de octetos, juntamente com os restantes dados da linha omnibus de dados será apresentada ao adaptador de linhas omnibus (154) durante o comando BSM WRITE SELECT UP seguinte. A contagem no contador (224) de limites de escrita BSM (todas as transferências excepto a última) ou no contador (22) de octetos de escrita BSM (última transferência) é carregada no registador (225) dos comandos de escrita do canal 1 do adaptador.

O processador S/88 (62) gera então um comando BSM WRITE SELECT UP na linha omnibus de endereços local (247) no formato representado na fig. 45Q, no qual os bits:

31-00 = 007E0104 = Comando BSM WRITE SELECT UP

O processador S/88 colocará também o endereço inicial do BSM na linha omnibus de dados (223), no formato representado na fig. 45R, no qual os bits:

31-24 = Reservados,

23-00 = Endereço inicial do BSM

O endereço inicial BSM na linha omnibus de dados (223) será capturado pelos octetos de ordem inferior do registador (227) A/D do canal 1 e o registador (228) de endereços de escrita BSM. Depois será enviado (como se verá mais adiante) para o adaptador de linhas omnibus (154) para apresentação à memória S/370 (162). A BCU (156) activa então as linhas DSACK (266 a,b) (porta de 32 bits) para o processador S/88 (62). Nesta altura, o processador S/88 (62) é libertado e deixa de estar envolvido nesta operação.

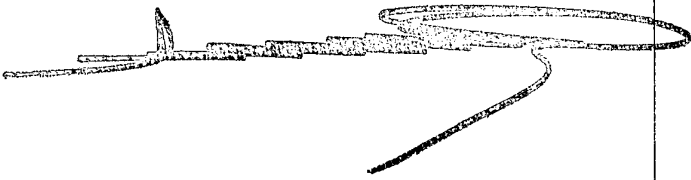
A lógica BCU (253) emite um comando BSM SELECT UP, encaminhando bits "01" para os bits de ordem superior do registador de comandos (225) através da linha omnibus (290) e coloca o comando e o comprimento do campo do registador (225) na linha omnibus (252), no formato representado na fig. 45S, sendo os bits:

0-1 = Comando (de escrita) BSM SELECT UP,

2-7 = Comprimento do campo, menos 1 (64 octetos no máximo)

O conteúdo do registador (227) é colocado na linha omnibus (251) de endereços/dados (251) (em dois subciclos), no formato representado na fig. 45T, no qual os bits:

0-3 = Chave da memória



4 = 1

5-6 = prioridade (adaptador de linhas omnibus para linha omnibus do processador)

7 = 1 = acesso à área de utilizador
0 = acesso à área de microcódigo

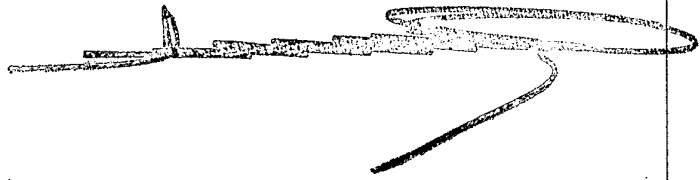
8-31 = Endereço S/370 do primeiro octeto no campo de dados

O comando e o comprimento do campo são armazenados no registador (125) do adaptador (154). Os dados da Chave/Endereço são armazenados no registador (123) do adaptador (154) através do registador SYNC (113). A lógica BCU (253) activa o sinal REQ2 na linha (263c) para o canal 2 do DMAC. O DMAC 209 emite o endereço inicial da memória tampão I/O do registador MAR para a memória (210), através da linha omnibus (248), do retentor (233), da linha omnibus (247), do multiplexador (232), para transferir uma palavra dupla de dados da memória (210) para o registador A/D (227). ACK2 (confirmação do canal 2 do DMA) é activado na linha (264a). Isso provoca um TAG UP na linha (262a) para o adaptador (154).

O adaptador (154) transfere então uma palavra dupla de dados do registador (227) para a memória tampão (260) do adaptador, em dois subciclos, através do registador (113). Repete-se uma sequência de escrita de sinais REQ/ACK seguidos de um comando TAG UP para transferir cada palavra de dados dupla. O BCU (156) decrementa a contagem de octetos nos registadores (222,224) e do endereço no registador (228) e no MTC do canal 2 do DMAC, para cada palavra dupla (32 bits) que é apresentada ao adaptador de linhas omnibus (154), até 64 octetos.

Se a contagem de octetos de transferência for maior que 64, então (como atrás se descreveu, relativamente às operações de escrita) a BCU (156) apresentará um novo endereço inicial de dados, para cada endereço inicial. Esta sequência repete-se até a contagem de octetos no registador (222) (máximo de 4KB) atingir zero.

Quando a memória tampão (260) do adaptador de linhas omnibus estiver cheia, a BCU (156) suspenderá a sequência de escrita até que o adaptador de linhas omnibus dê uma indicação de memória tampão disponível, através da linha TAG DOWN (262c).



O adaptador de linhas omnibus (154) converte o comando BSM SELECT UP na linha omnibus do adaptador, num comando I/O MEMORY na linha omnibus do processador S/370, num formato representado nas fig. 45U e 45V na linha omnibus (170) do processador, e na linha omnibus de chave/estado, sendo:

Bits na linha omnibus do processador:

- 0 = 0 = Comando I/O MEMORY
- 1 = 0 = Operação de memória
- 2-7 = Comprimento do campo
- 8-31 = Endereço real do octeto

Bits da linha omnibus de Chave/Estado:

- 0-3 = Chave da memória
- 4 = 0 = Ausência de translação dinâmica.

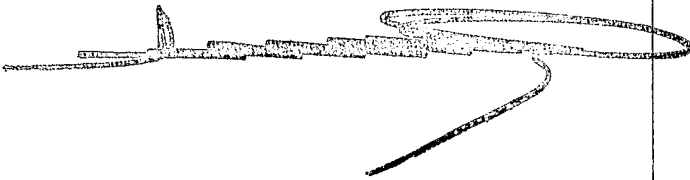
Quando se tiverem transferido todos os dados (contagem de octetos = 0) o DMAC (209) activará a linha de interrupção (258a) para o codificador de prioridades (212) do processador S/88.

c) FLUXO DA SEQUÊNCIA DE TRANSFERÊNCIA DE MENSAGENS DE PRIORIDADE ELEVADA S/370

Todas as transferências de mensagens de alta prioridade são originadas a partir do subsistema I/O (processador S/88 (62)). O canal 3 do DMAC será ajustado pelo processador S/88 (62) para efectuar a transferência de dados (16 octetos). A BCU (156) utilizará o canal 1 da linha omnibus do adaptador para a comunicação de dados (comando Q SELECT UP).

A BCU (156) detecta um pedido de mensagem de alta prioridade quando o processador S/88 (PE 62) efectua uma carga de contagem de transferências de memória DMAC para o registador MTC no canal 3. Como consequência disso, a BCU (156) gera um comando Q SELECT UP para o PE S/370 (95) na linha omnibus (252) do adaptador do canal 1. Se estiver em curso uma transferência de dados de leitura I/O S/370 (canal 1 da linha omnibus do adaptador) quando for detectado o pedido, então a BCU espera até estar completa a transferência do bloco corrente de 64 octetos antes de satisfazer o pedido.

SE não houver qualquer actividade no ca



nal 1 da linha omnibus do adaptador, então o pedido será satisfeito imediatamente.

Vai agora descrever-se com mais pormenor esta transferência de mensagens de prioridade elevada. O PE (62) obtém o controlo da linha omnibus local (223,247), se ela não estiver a ser utilizada pelo DMAC (209). O PE (62) armazena então os dados da mensagem na memória local (210) através do controlo do programa. O PE (62) coloca o endereço inicial da mensagem da memória tampão local no registador MAR de endereços de memória do canal 3 do DMAC, colocando a informação na linha omnibus de endereços local (247), no formato representado na fig. 45W, sendo os bits:

31-08 = 007E00 = comando DMAC Register Select

37-00 = DMAC Channel 3 Memory Address Reg (Low) Select

Coloca-se o endereço inicial dos dados da mensagem da memória tampão local pretendido para o registador de endereços de memória na linha omnibus de dados (223) no formato representado na fig. 45X, sendo os bits:

31-16 = O endereço inicial dos dados da memória tampão local na memória (210)

15-00 = Reservados

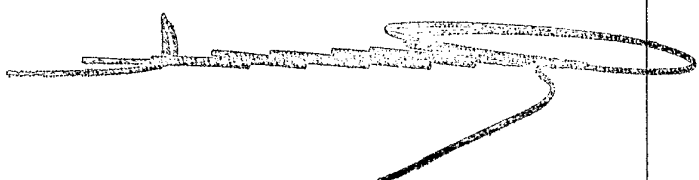
Os bits de ordem superior (bits 31-16) da linha omnibus de dados serão carregados na parte dos bits de ordem inferior (bits 15-0) do registador MAR de endereços de memória do canal 3 do DMAC. Os bits de ordem superior (bits 31-16) do registador MAR foram colocados em zero durante a inicialização. O DMAC (209) responde com um sinal DTACK na linha (265), que é convertido para um sinal DSACK da porta de 16 bits nas linhas (266 a,b), através da lógica BCU (253) para o processador S/88 (62).

O processador S/88 (62) coloca então um comando na linha omnibus de endereços local (247) no formato representado na fig. 45Y, sendo os bits

31-08 = 007E00 = comando DMAC REGISTER SELECT

07-00 = Seleccionar BCU e DMAC canal 3 MTC

Os bits da contagem de octetos, da chave da memória e do espaço utilizador/IOA serão colocados na linha omnibus de dados pelo processador S/88 (62), no formato re-



presentado na fig. 45Z, sendo os bits:

31-20 = Reservados

19-16 = bits de contagem de octetos da transferência. Estes bits são carregados no DMAC (209) e na BCU (156). Representam uma contagem de palavras duplas para o DMAC (209) e para a BCU (156) (máximo de 64 octetos).

15-12 = Zero

11-08 = Chave da memória

07 = Bit do espaço utilizador/IOA

06-00 = Reservados

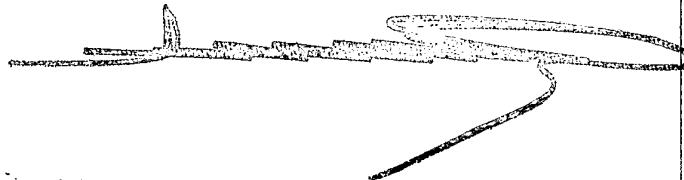
O DMAC (209) carregará a palavra de ordem superior (contagem de octetos) da linha omnibus de dados (223) no registador MTC de contagem de transferências na memória do canal 3. A BCU (156) capturará o conteúdo da linha omnibus de dados quando este comando particular aparece na linha omnibus de endereços (247), armazenando os bits 19-16 no contador Q Select Up (254) e os bits 11-07 no registador (227) de endereços e dados do canal 1.

O DMAC (209) responde com um sinal DTACK para a lógica (253), que o converte numa resposta DSACK da porta de 32 bits nas linhas (266 a,b) para o PE (62). Esta acção alerta a BCU (156) para iniciar uma transferência de mensagens de alta prioridade da memória local (210) para o BSM S/370 (162). A contagem de octetos de transferência, juntamente com os dados adicionais representados na fig. 45Z, são apresentados ao adaptador de linhas omnibus (154) durante um comando Q Select Up gerado na BCU. O contador (254) de Q Select é carregado nos bits 4-7 do registador (225) de comandos de escrita no canal 1. A BCU (156) coloca o comando Q Select Up no registador (225) através da linha omnibus (290); e os dados no registador (225) são colocados na linha omnibus (252) (canal 1) do adaptador no formato representado na fig. 45AA, sendo os bits:

01 = Comando (de escrita) Q Select Up

2-7 = Comprimento do campo, menos 1 (16 octetos)

A informação colocada na linha omnibus de endereços/dados (251) através do registador (227) está representada na fig. 45AB, sendo os bits:

- 
- 0-3 = Chave da memória
4-6 = Zero
7 = 1 = Acesso à área de utilizador
0 = Acesso à área de microcódigo
8-31 = Não considerar.

Os dados nas linhas omnibus (252) e (251) são transferidos para os registadores (125) e (123) do adaptador, respectivamente. A lógica BCU (253) activa então a linha REQ (263d) (pedido do canal 3 do DMA). O DMAC (209) coloca o endereço inicial da memória tampão I/O (proveniente do registador MAR) na linha omnibus local (247) e activa ACK (confirmação do canal 3 do DMAC) na linha (264d). A BCU (156) transfere então os primeiros quatro octetos de dados da memória tampão I/O endereçada da memória local (210) para a memória tampão (260) do adaptador, em dois subciclos, através do registador SYNC (113). Os blocos de quatro octetos seguintes são transferidos por sequências dirigidas pelo comando TAG UP para o adaptador de linhas omnibus (154), e as linhas REQ/ACK (263d, 264d) para o DMAC. A BCU (156) decrementa o contador de octetos para cada palavra dupla (32 bits) que é apresentada ao adaptador de linhas omnibus (154).

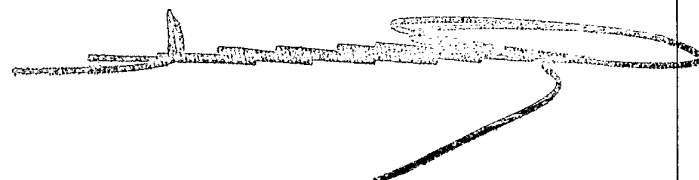
O adaptador de linhas omnibus (154) converte o comando Q Select Up em um comando de memória I/O da linha omnibus do processador para emitir a mensagem para a área (189) da memória (162); o formato do comando está representado na fig. 45AC, no qual os bits do PROC BUS (170) são:

- 0 = 0 = Comando de memória I/O
1 = 0 = Operação de armazenamento
2-7 = Comprimento do campo (máximo 64 octetos)
8-31 = Endereço real dos octetos (dos registadores (110, 112) do adaptador).

A linha omnibus KEY/STATUS do processador (85) tem dados no formato representado na fig. 45AD, sendo os bits:

- 0-3 = Chave da memória
4 = Ausência de translacção dinâmica.

Quando se tiverem transferido todos os dados da mensagem para o adaptador de linhas omnibus (154) (con-



tagem de octetos = 0), o DMAC (209) activará a sua linha de interrupção (258a) para o codificador de prioridades do processador (211). O DMAC (209) apresenta vectores de interrupção a partir do octeto menos significativo da sua linha omnibus de dados (248) para a linha omnibus de dados do processador S/88 (161D), os bits 23-16, através do receptor activador (234) e os bits 23-16 da linha omnibus de dados local (223). O DMAC devolve para o PE (62) um sinal DSACK de 16 bits.

d) Comando READ BCU STATUS

Um comando READ BCU STATUS pode ser formado pelo processador S/88 (62) para ler o estado corrente da BCU (156). O comando é colocado na linha omnibus de endereços (247) pelo processador S/88 (62), no formato representado na fig. 45AE, sendo os bits:

31-00 = 007E010C - Comando READ BCU STATUS

A BCU (156) colocará o estado representado na fig. 45AF na linha omnibus de dados e devolverá DSACK (porta de 32 bits) pela linha omnibus (266) da PE (62). Os bits da fig. 45AF representam:

31-29 = Estado do canal 0 da linha omnibus do adaptador -
verificação da chave, verificação do endereço

28 = 1 = Último ciclo de dados
0 = todos os outros ciclos

27-26 = estado do canal 1 da linha omnibus do adaptador,
verificação da chave, verificação do endereço

25 = Memória tampão não disponível (Comando Q Select
Up)

24 = 1 = Último ciclo de dados
0 = Todos os outros ciclos

23 = Canal 0 da linha omnibus Tag Down

22 = Canal 1 da linha omnibus Tag Down

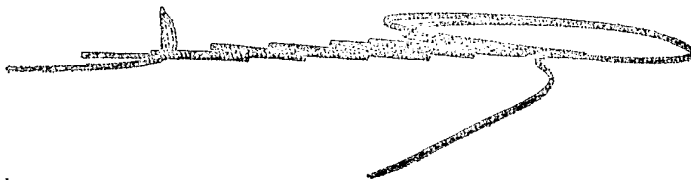
21 = BSM Read Sync Check

20 = BSM Read Select Up/Pending Latch

19 = BSM Write Select Up Request/Pending Datch

18 = Q Select Up Request/Pending Latch

17 = Read Mailbox em progresso

- 
- 16 = BSM Read em progresso
15 = BSM Write em progresso
14 = Q Select Up em progresso

O bit 21 de estado da BCU (BSM Read Sync Check) será repostado depois de ser lido pelo processador S/88 (62). Este bit indica que as contagens de octetos do adaptador de linhas omnibus (154) e da BCU (156) não concordam quando termina uma operação de leitura do BSM; portanto é detectado um erro que exige a ressinchronização.

Para a operação de escrita BSM, o adaptador de linhas omnibus (154) activará Tag Down em (262b), para indicar que foram recebidos todos os dados. Tag Down em (262b) será então desactivado pelo adaptador de linhas omnibus (154), altura em que os indicadores de estado serão apresentados a, e calculados pela unidade BCU (156). Se Tag Down não for desactivado dentro de 100 microssegundos. A BCU (156) activará uma linha de anulamento (não representada) para o adaptador de linhas omnibus (154). Isso fará então com que o adaptador (154) se desligue da unidade BCU (156). Tag Down em (262b) é também usado pelo adaptador de linhas omnibus (154) para indicar qualquer erro que não possa ser comunicado à BCU (156) através da linha omnibus de comandos/estados (252).

e) Reposição programada da BCU

Uma reposição programada da BCU, realizada pelo PE (62), desempenha a mesma função que uma reposição da BCU quando da ligação da energia. Ela pode ser realizada em qualquer instante para limpar a BCU de quaisquer condições anormais. Contudo, tem de ser reconhecido pelos meios físicos um ciclo da linha omnibus local (descodificação de 007EXXXX) para se executar este comando.

O comando é colocado na linha omnibus de endereços local (247) pelo processador S/88, no formato representado na fig. 45AG, sendo os bits:

31-00 = 007E0000 - Comando Reset BCU.

O conteúdo da linha omnibus de dados será ignorado pela BCU (156). A BCU (156) devolverá DSACK (porta



de 32 bits) nas linhas (266 a,b) para o processador S/88 (62).

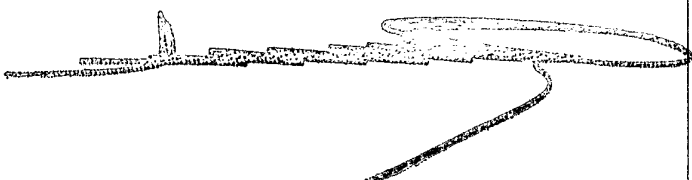
EMULAÇÃO DO FORMATO DA PISTA DE CONTAGEM, CHAVE E DADOS (fig. 46A-K)

A emulação do DASD S/370 no S/88 será descrita, a título de exemplo, para ilustrar uma maneira preferida de como os programas I/O S/370 podem ser executados pelos processadores e dispositivos I/O S/88. O S/370 será referido como Sistema Objecto e o S/88 como Sistema Objectivo. Os dados do DASD (Direct Access Storage Device - Dispositivo de memória para o acesso directo) para o sistema objecto são mantidos pelo sistema objectivo num Formato de Emulação. O código S/370 que se executa no processador S/370 é designado por meios de programação do sistema objecto. A discussão divide-se em quatro partes:

- 1) O Sistema Objecto - apresenta uma breve descrição do formato de gravação de contagens, chaves e dados, usado pelos produtos de memória de acesso directo S/370, existentes.
- 2) O Sistema Objectivo - Descreve o modelo de interface dos programas DASD.
- 3) O formato de emulação - descreve a atribuição de posições dos campos do sistema objecto nos formatos de emulação usados.
- 4) A função de emulação - descreve a distribuição das posições das funções do Sistema Objecto nas funções de emulação.

1. O SISTEMA OBJECTO

Os meios físicos do DASD estão divididos em cilindros e os cilindros em pistas. O número de cada um deles e a sua capacidade variam para os diferentes tipos e modelos de DASD. Cada cilindro pode ser endereçado, por programa, por um número do cilindro (CC), de dois octetos, e as pistas individuais dentro de um cilindro são acedidas por cabeças separadas de leitura/escrita, cada uma delas endereçável por um número de cabeça (HH), de dois octetos. A situação física de uma pista



é dada pelos seus números de cilindro e de cabeça, sendo portanto especificada pelo endereço (CCHH), de quatro octetos. Cada uma das pistas tem um endereço de repouso, um descritor da pista (registro 0) e um ou mais registros de dados. A dimensão de cada registro é programável; e quando se escrever o endereço de repouso e as dimensões dos registros numa pista, diz-se que essa pista está formada. Todas as pistas são formatadas a partir do seu índice da pista para o índice da pista seguinte. A fig. 46A ilustra uma tal pista.

A unidade de informação básica gravada no meio físico é um octeto de dados constituído por oito bits. Um grupo de octetos de dados constitui uma área, e o dispositivo separa estas áreas por intervalos de escrita entre as mesmas. Cada registro consta de duas (contagem, dados) ou de três (contagem, chave, dados) áreas, enquanto que o endereço de repouso é constituído por apenas uma área. As três áreas que formam um registro do sistema objecto são: contagem, chave (optativo) e dados.

A área de contagem contém os campos seguintes:

F	Indicador	1 octeto indicando a condição da pista, transbordo da pista de registros lógicos
CCHH	Endereço da pista	2 octetos indica o número do cilindro e o número da cabeça onde a pista está situada fisicamente
R	Número do registro	1 octeto indicando o número sequencial do registro na pista
KL	Comprimento da chave	1 octeto indicando o número de octetos na área da chave
DL	Comprimento dos dados	2 octetos indicando o número de octetos da área de dados
ECC	Código de erro	2 octetos usado para código de detecção/correção de erros

A área da chave contém os seguintes campos:

Se $KL = 0$, esta área e o seu intervalo

são omitidos)

KEY	Chave	Octetos KL	Dados do utilizador
ECC	Código de erros	2 octetos	usado para detecção/ /correção de erros

A área de dados contém os campos seguin

tes:

DATA	Dados	Octetos DL	Dados do utilizador
ECC	Código de erros	2 octetos	usado para detecção/ /correção de erros

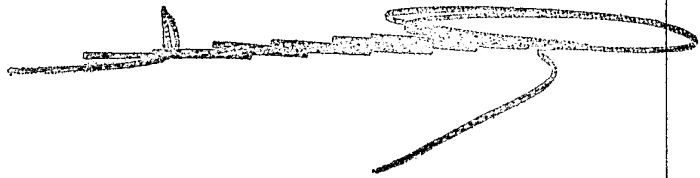
A primeira área em cada pista é o endereço de repouso. Ela contém os campos seguintes:

F	Indicador	1 octeto	indicando a condição da pista
CCHH	Endereço da pista	2 octetos	indica os números do cilindro e da cabeça onde está fisicamente localizada a pista
ECC	Código de erros	2 octetos	usada para detecção/ /correção de erros

O Registo 0 (descriptor da pista) é sempre o primeiro registo a seguir à área do endereço de repouso. No sistema de programação preferido, o registo 0 do campo CCHH define a pista alternada, se a pista tiver recebido um indicador como defeituosa. O Comprimento da Chave é normalmente 0 para a pista 0. O registo 0 pode ser seguido por um ou mais registros de dados. A área da chave é optativa e, se estiver presente, pode conter de 1 a 255 octetos. O número de um registo é obtido quando um comando de Write CCW escrever as áreas da contagem, da chave e dos dados. Depois de o registo ter sido formatado, as áreas de dados do utilizador podem ser lidas e/ou re-escritas (utilizando outros comandos CCW) sem destruir registros adjacentes na pista. Se um registo for reformatado, os que se lhe seguem são destruídos.

2. O SISTEMA OBJECTIVO

O DASD (fig. 46B) é apresentado ao mi-



crocódigo S/88 sob a forma de ficheiros que contêm blocos de 4 096 octetos de dados, numerados sequencialmente a partir de 1. O mecanismo de emulação distribui as posições do formato do sistema objecto e a função numa combinação do formato e função do sistema objectivo utilizável.

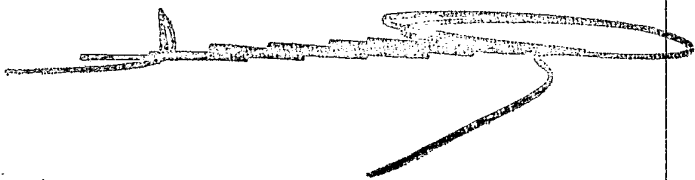
3. O FORMATO DE EMULAÇÃO

Os parâmetros físicos de diferentes tipos e modelos de DASD no sistema objecto variam. O número do tipo e modelo do DASD, juntamente com os vários parâmetros, são mantidos no primeiro bloco de dados, INFO, do ficheiro do sistema objectivo, fig. 46C. O restante do ficheiro contém os dados emulados das pistas objectos (fig. 46C). Os dados de cada pista são mantidos num número inteiro de blocos de dados. O número de blocos de dados do sistema objectivo necessários para cada pista é um parâmetro mantido no primeiro bloco de dados. Cada pista no sistema objecto, a começar com CCHH = 0000, é mantida sequencialmente no ficheiro do sistema objectivo. O seu número de bloco inicial pode ser calculado dados CCHH e as dimensões do disco objecto mantidas no bloco INFO.

Cada pista emulada (fig. 46D) contém uma lista dos registos existentes correntemente nessa pista, um cabeçalho da lista e os dados de utilizador (chave, dados) para cada pista. A lista é usada para localizar os dados para um registo específico, efectuar operações de busca no registo ou de chave, aceder ao último registo e tratar o excesso na pista.

Os dados do sistema objecto são tratados no ambiente de emulação de uma de três maneiras: mantido, retido, implicitamente ou não mantido.

Nem todos os intervalos são necessários e não são mantidos. Os dados ECC não são criados nem mantidos visto que a integridade dos dados é assegurada pelo sistema objectivo. Como o modelo de programa proporcionado pelo sistema objectivo elimina toda a área da superfície física avariada, as pistas alternadas no sistema objecto são realizadas de uma maneira sem faltas. Isto significa que a parte do octeto Indica-



dor (F), que indica a condição da pista, não é mantida e os octetos de Indicador (FLAG) escritos pelos meios de programação do sistema objecto são verificados para a validade e descartados.

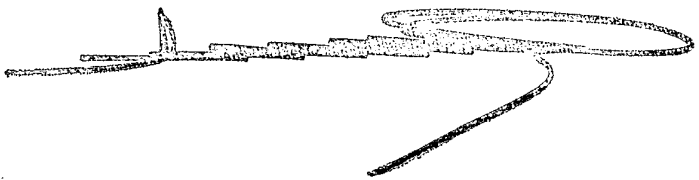
O CCHH (endereço da pista) passado pelos meios de programação do sistema objecto é usado para calcular a localização da pista emulada no ficheiro DASD do sistema objectivo. Ele é mantido no cabeçalho da pista descrito mais adiante, mas não se faz proliferar através de todas as áreas de contagem e do endereço de repouso da pista emulada. O endereço de repouso não é mantido como uma área explícita. O número (R) do registo, também passado pelos meios de programação do sistema objecto, é mantido implicitamente e não aparece como dados explícitos.

Os dados do utilizador, a CHAVE optativa e os campos de dados, para cada registo, são mantidos de uma maneira sequencial na pista emulada imediatamente a seguir à lista da pista, fig. 46D.

O restante dos dados do sistema objecto (F) (excesso lógico da pista de registos), KL e DL) é mantido na lista da pista, fig. 46E. Uma entrada da lista contém F, KL e DL, bem como um apontador (p) para os dados do utilizador (CHAVE e DADOS) para cada registo. R é mantido implicitamente como número de entrada da lista. A fig. 46E mostra a composição do cabeçalho, da lista e dos dados do utilizador, bem como a distribuição das posições de uma pista emulada nos blocos de 4 KB do sistema objectivo. Os apontadores p0-p2 apontam para os endereços iniciais (dentro dos blocos de 4 KB) dos registos de dados 0-2.

4. FUNÇÕES DE EMULAÇÃO

Esta secção dirige-se para a utilização dos formatos de emulação atrás descritos proporcionando alguns dos comandos DASD CCW do sistema objecto. As fig. 46F-K, inclusive, representam dados transferidos pelos meios de programação do sistema objecto durante operações de leitura e de escrita.



Para as operações (ops) CCW que implicam o endereço inicial, são calculados os valores de F e CCHH da fig. 46F, são verificados, mas não se escreve nada na pista emulada.

Para operações CCW que envolvem o registro 0, fig. 46G, os campos CCHH e R são verificados, mas nada se escreve. Os campos KL e DL são transferidos para/da entrada apropriada da lista. O registro zero tem um desvio zero na área de dados do utilizador. O registro 0 de leitura/escrita orienta a cabeça para o primeiro registro na pista.

As operações CCW que envolvem contagem orientam sempre a cabeça para o registro seguinte na pista, fig. 46H. Para operações que envolvem Chave e Dados, a localização e as dimensões dos dados do utilizador são encontradas na lista (fig. 46I). As operações que envolvem contagem, chave e dados orientam a cabeça de leitura/escrita para o registro seguinte na pista (fig. 46J). Para as operações que envolvem contagem múltipla, chave e dados, o processamento começa com a entrada seguinte da lista e continua até à última entrada válida da lista (fig. 46K).

Partilha da memória real (16) pelo S/88 e o S/370

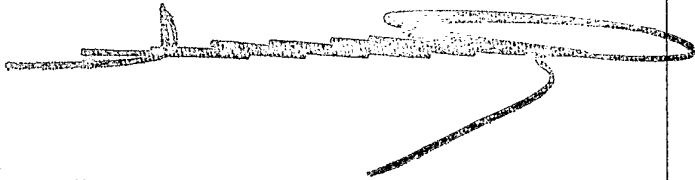
1. Introdução

Vai agora descrever-se com mais pormenor o "roubo" de uma ou mais áreas da memória real (física) (16) para um ou mais processadores S/370 na gestão e na distribuição das posições na memória (16), fazendo-se referência às figuras:

Fig. 10, que ilustra teoricamente a memória virtual S/88 (106) e a memória física (16) do S/88 e a atribuição das áreas de armazenamento da memória física S/370 (162-164) para os processadores S/370 (21,23), (25,27) e (29, 31);

A fig. 47, que ilustra esquematicamente o processo de captura de uma área de memória S/370 da memória física S/88 (16); e

As fig. 48 A-K, que ilustram a distribuição de posições dos meios de programação virtuais/físicos, tais como são usados na gestão da memória S/88, sendo essa dis-



tribuição de posições controlada para permitir a captura da área de memória S/370.

A memória (16) está dividida em páginas de 4 KB e uma pluralidade de entradas do mapa da memória (mme), uma para cada página de 4KB, está contida nas matrizes de entradas do mapa da memória (fig. 48A) que, em conjunto, formam o mapa de toda a memória (16). As entradas correspondentes a páginas não atribuídas para uso estão ligadas numa "lista livre" (isto é, a fila de espera de atribuição de memória) incluindo em cada entrada (fig. 48B) os números das páginas seguintes (apontadores) das entradas anteriores e seguintes na lista. Um apontador dos meios de programação no sistema operativo S/88 aponta sempre para o início da lista livre. As páginas da memória física são atribuídas a vários processos a partir do início da lista livre e as páginas devolvidas para a lista livre são de preferência colocadas no início da lista livre. Os números das páginas "anterior e seguinte" e o apontador dos meios de programação para o início da lista livre são actualizados de maneira apropriada.

Quando o sistema S/88 é inicializado, estas entradas são colocadas por ordem sequencial de endereços na lista livre; nesta altura apenas poucas páginas estão atribuídas para utilização. Portanto, há grandes áreas contíguas de memória (16) disponíveis para atribuição a partir da lista livre. Por conseguinte, no instante da inicialização, as áreas de memória (por exemplo (162), (163), (164)) têm de ser "roubada" para os processadores S/370. Subsequentemente, à medida que se atribuem páginas a partir da lista livre e se devolvem páginas à mesma, como for requerido, os grandes blocos contíguos na lista livre fraccionam-se e deixam de estar disponíveis. Se se fizesse uma tentativa para criar uma área S/370 contígua, seria necessário interromper todos os processos e executar rotinas complexas para atribuir de novo blocos de memória já atribuídos a vários processos até se dispor de memória contígua suficiente.

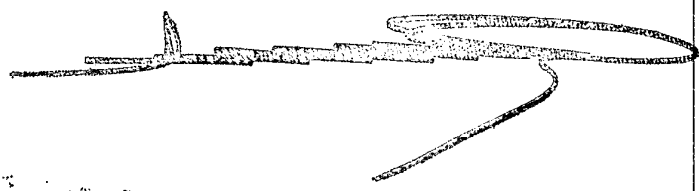
As rotinas de serviço, no programa de aplicação EXEC370, descritas mais adiante, proporcionam as funções para roubar áreas de memória S/370 a partir do sistema operativo S/88.

2. Distribuição de posições da memória S/88 (16)

Mas primeiro vai descrever-se uma forma preferida de gestão/distribuição de posições (elaboração do mapa) da memória principal S/88 (16), com referência às fig. 48 A-K, inclusive. A fig. 48A é uma vista geral simples da estrutura de meios de programação estabelecida pelo sistema operativo S/88 (S/88 OS) para manter um espaço de endereços virtuais do processo. A estrutura dos meios de programação inclui os elementos seguintes:

- pte - Process table entry (entrada da tabela de processos)
(representa um processo)
- pmb - Process map block(s) - Bloco(s) do mapa de processos. Encadeados entre si, contêm os apontadores (pme) para as apte para este espaço de endereços virtuais de processos
- pmbp - um apontador na pte para o primeiro pmb na cadeia
- pme - entradas do mapa de processos (apontadores para as apte) contidas nos pmb
- mme - entradas do mapa da memória física. Contidas nas matrizes das mme há uma mme para cada página de 4 KB de memória física no sistema, isto é na memória (16).
- apte - entrada da tabela da página activa. Contida em blocos aptos, há um bloco apto para cada página virtual única no sistema
- vpn - Número de página virtual no interior do espaço de endereços virtuais do processo
- pmt - Tabela de gestão de processos. Há um ponteiro ptep na pmt para cada processo (pte) no sistema
- ptep - Apontador da entrada na tabela de processos para um processo.

A estrutura do mapa da memória da fig. 48A é usada pela unidade de gestão da memória (105), fig. 10 e 47. Ela consiste em uma ou mais matrizes (fig. 48C) de mme, cada uma contendo 512 mme ordenadas, na forma de realização preferida. Cada mme representa uma página de 4 KB de memória real (16), e portanto uma matriz de mme representa 512 x 4 KB = 2 MB de memória contígua.



A caixa com a designação "Matriz do Mapa da Memória" da fig. 47 ilustra teoricamente todas as matrizes de mme dispostas por ordem sequencial de endereços.

As mme são usualmente arrumadas numa de três listas:

- 1) lista usada, mme atribuídas a um processo
- 2) lista recuperada, mme a devolver à lista livre
- 3) lista livre, mme disponíveis para atribuição a um processo.

As mme são deslocadas de uma lista para a outra, sendo os seus apontadores actualizados de maneira apropriada.

Se elas não estiverem numa lista, ou representam uma página com cablagem permanente ou estão num estado transitório. A estrutura dos dados das mme, usada pela unidade de gestão da memória (105) contém os apontadores das três listas indicados na fig. 48B, onde:

indicadores	cablados	página está cablada
	I/O em progresso	I/O de disco avançado agora indica a última I/O (ou a I/O corrente) para esta trama foi uma
	escrita	escrita para disco a página tem uma PTW (palavra da tabela física) nos registadores dos meios
	ligado	físicos
	modificado	última observação do bit modificado
	não usado (2)	Notifica posto para limpeza
	evicção, limpeza	Notifica o posto para limpar e libertar esta página
	não usado (1)	Alguma avaria da página aguardando nesta página
	evicção libertar	ppn (número da página física) para a mme seguinte
	avaria de página	ppn para a mme anterior
	mme seguinte	endereço de disco, enquanto em memória
	mme anterior	apontador para apto para esta
	endereço	
	aptep	

página

Os campos "mme seguinte" e "mme anterior" são usados para criar as listas encadeadas (usada, de recuperação e lista livre).

São os números das páginas físicas para a mme seguinte e para a mme anterior que serão alterados como se descreve adiante, quando a memória física do S/88 é capturada para uma área de memória S/370. Na forma de realização preferida, cada uma das matrizes mme (fig. 48C) é uma lista de 128 apontadores, cada um dos quais é um endereço virtual de uma matriz mme. Os primeiros n apontadores são uma lista ordenada de todas as matrizes mme. Os restantes $128-n$ apontadores não ZEROS. Isso proporciona a capacidade de manter o rasto de $128 \times 2 \text{ MB} = 256 \text{ MB}$ de memória real. Cada um destes apontadores compreende os 16 bits mais significativos (de ordem mais elevada) de um endereço físico, chamado um número de página física (ppn), e são usados como apontador para uma mme específica. Os sete bits de ordem superior do ppn seleccionam a matriz mme, e os nove bits de ordem inferior do ppn seleccionam a mme no interior da matriz. Os doze bits de ordem inferior do endereço físico são um desvio na página real (física) da memória (16).

Uma estrutura da informação do mapa da memória (mem map info) (fig. 48D) é usada para guardar o rasto da memória usada para os mapas, sendo:

mem map infop-1	apontador para a estrutura da primeira informação do mapa da memória
next mem map infop	apontador para a estrutura da informação seguinte do mapa da memória
n pages	número de páginas de 4K de memória real usadas por este mapa (máximo 16)
per page (16)	o restante da estrutura é uma matriz de informação por página
ppn	número de página física para mme para esta página.

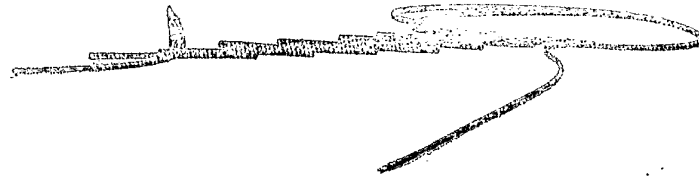
As entradas da tabela de páginas activas (apte) são usadas para manter o rasto da memória virtual. Há uma apte para cada página de 4 KB de memória virtual em todos os espaços de memória virtual no sistema. A estrutura apte

(fig. 48E) indica o(s) dono(s) do espaço virtual, o endereço virtual da página e o endereço de memória real do endereço do disco, se chamado.

Se mais de um processo compartilhar o mesmo espaço de endereços virtuais, todos os processos são identificados através de um aposto ("trailer") apte (fig. 48G); e o apte para cada página virtual aponta para o "aposto".

A estrutura apte inclui:

endereço	endereço de real página de 4K	(indicadores mem atribuídos = 1)
	endereço de disco endereço do apte livre seguinte se apte estiver na lista livre	(indicadores mem atribuídos = 0)
indicadores	processo per	página virtual não partilhada com outros processos
	página bifurcada	a página do processo per foi bifurcada
	mem atribuída espera	a página tem memória atribuída espera para esta página
	erro I/O libertar apte	ocorreu um erro I/O na página Libertar este APTE quando I/O se completar
	correção tipo cpu	a página foi corrigida na inicialização
	mau endereço, atribuído de novo contagem	erros forçaram um novo endereço
vpag	número de páginas virtuais.	número de processos que compartilham esta página
		O número de páginas virtuais (vpn) consiste nos 16 bits mais significativos dos 27 do endereço virtual.
processo	endereço de pte pa	
pta	ra o processo per ou	(se não for partilhada a memória virtual)



endereço do aposto (se for partilhada a memória
do apt virtual)

Cada apte tem um comprimento de doze octetos e estão contidas 256 entradas em cada bloco de tabelas de páginas activas (apt) (fig. 48F). A posição relativa das apte dentro de um bloco não tem qualquer significado. Todas as apte não usadas são incluídas numa lista de aptep livres. Se forem necessárias apte adicionais e a lista estiver em zero, é atribuído um novo bloco apt na nova pilha, sendo todas as 256 apte introduzidas nas listas rpteo livre.

O aposto apt (fig. 48G) é usado para zonas de programa partilhadas; é atribuído na pilha do sistema cablada e para ele aponta uma EITE (Executable image table entry - Entrada da tabela imagem executável) haverá quatro apostos por programa (um por zona). Os apostos permitem que o sistema encontre todas as palavras PTW que apontam para uma página quando se remove a mesma.

A estrutura do aposto apt inclui:

n procs	número de processos que usam o aposto
v base (zona base vpn)	primeira página virtual desta zona
n pages	número de páginas na zona
users	mapa dos bits dos utilizadores do aposto
pp info(o: :nnp)	o restante da estrutura é uma matriz da informação do processo per
npp	dimensão da matriz
n ptws	número de palavras PTW ligadas neste instante
aptep	apontador para APTE para esta página

A entrada da tabela do processo (fig. 48H) contém a informação necessária para gerir um processo; con

têm informação acerca do espaço de endereços virtuais do processo. Cada entrada da tabela de páginas inclui:

first pmb pta apontador para o primeiro pmb numa lista de pmb para este processo

map root tbl phys Endereço físico de map físico
addr

map root ptr phys Endereço virtual de map físico

pdr ptr Endereço da zona de dados do processo per

A estrutura do bloco do mapa do processo (fig. 48I) é usada para estabelecer o mapa de um espaço virtual do processo no espaço da memória real, e inclui:

nextp Apontador para o pmb seguinte para este processo

base vpn número da página virtual base, primeiro o número da página virtual base deste pmb (os seis bits menos significativos serão ZERO)

map addr endereço físico do map

pme entradas do mapa do processo (0-63), sendo o restante da estrutura uma matriz de informação da página per. O índice para esta matriz é constituído pelos seis bits menos significativos do vpn

indicadores usado cópia de bit usado
 em mem a página está em memória
 não usado (1)

linha divi-

sória esta página é uma página divisória

cablado cablar esta página quando chegar

copy on write copiar a página quando escrita

patched a página é uma página de código corrigida

ufence página divisória do utilizador

aptep apontador para APTE para esta página

A tabela de gestão do processo (fig. 48J) contém informação usada pelo listador, incluindo uma lista de apontadores ptep para todos os processos no sistema, o número de páginas disponíveis no sistema e o número de páginas entre-

gues.

48 K inclui:

ac1	código de acesso da PTW
ppn	número da página física da página desejada
ac2	código de acesso à palavra PTW
u	esta PTW é usada

3. Processo de arranque

O sistema S/88 inclui um procedimento de arranque que liga a energia do sistema e faz iniciação dos módulos de programas e de dados que estão incluídos num ficheiro de arranque.

No arranque automático, a memória só para leitura, programável (PROM) (181) (fig. 12), executa diagnósticos e faz ensaios automáticos quer nos componentes do sistema S/88 quer nos do S/370. Terminadas essas tarefas, a PROM (181) lê um programa de utilização que carrega o sistema operativo a partir de um disco mãe (não representado).

O código de arranque do módulo inicializa todos os dispositivos e discos configurados e ajusta o relógio a partir do relógio calendário do sistema. Este ficheiro contém comandos que o sistema operativo executa como parte do procedimento de arranque de um módulo. Este procedimento inclui funções de:

- leitura de ficheiros de tabelas que especificam configurações de cartões, discos e dispositivos ligados ao módulo;
- identificação dos módulos dentro do sistema; e
- arranque de vários processos de serviço do sistema.

O ficheiro do módulo fornece dados suficientes para criar um novo sistema e pode ser modificado pelo utilizador para o adaptar aos seus requisitos. Para capturar uma área S/370 (162-164) a partir da memória principal S/88, são inseridas certas declarações no ficheiro de comandos do código de arranque do módulo. Por exemplo, se partirmos da hipóte

-

car pelo menos um MB.

provided-less - O número de MB deslocados é menor
que o pedido.

- b) A sub-rotina Replace Storage devolve um bloco de memória física para as tabelas do sistema operativo S/88

Uso

```
declarar S/370 replace-stor entry (binary (15),  
                                     binary (15)  
                                     binary (15);  
chamar S/370 replace-stor (n-blbks,  
                             ppn,  
                             erro-code);
```

Argumentos

n-blks (entrada)

O número de megaoctetos contíguos que
estão a ser devolvidos ppn (entrada)

O número das páginas físicas de base do
bloco. Os oito bits menos significativos de ppn têm de ser
0.

error-code (saída)

cannot-free-connected - Tem de usar S/370 Close Storage antes de tentar repor a memória em VOS.

- c) A sub-rotina S/370 Open Storage liga uma parte ou toda a memória física anteriormente deslocada para o espaço de endereços virtuais do chamador e devolve o número de páginas virtual. Estabelecem-se todas as apte e pme apropriadas e estabelece-se a distribuição de posições de virtuais para físicas. O código de acesso é "Read/Write" e a memória é cablada.

uso

```
declarar S/370 open-stor entry (binary (15),  
                                binary (15)  
                                binary (15)  
                                binary (15);  
chamar S/370 open-stor(n-blbks,  
                        ppn  
                        vpn  
                        error-code);
```


Argumentos

n-blks (entrada)

O número de megaoctetos contíguos pedidos.

ppn (saída)

O número da página física da primeira página de 4K na zona.

Os oitos bits menos significativos de ppn serão 0.

vpn (saída)

O número de página virtual da primeira página de 4K na zona. Os oito bits menos significativos de vpn serão 0 e o endereço virtual é 4096 vpn erro-code (saída)

Um código de erro devolvido.

d) S/370 Close Storage

A sub-rotina S/370 Close Storage desliga a memória física anteriormente aberta do espaço de endereços virtuais dos chamadores. As APTE e PME apropriadas serão devolvidas para o sistema operativo S/88 e a distribuição de posições de memória virtuais para físicas é censurada. A memória física é devolvida à rotina S/370 Displace Storage.

Uso

declarar S/370 close-stor entry (Binary (15)
(binary(15)
(binary (15)

call S/370 close-stor (n-blocks
vpn,
error-code);

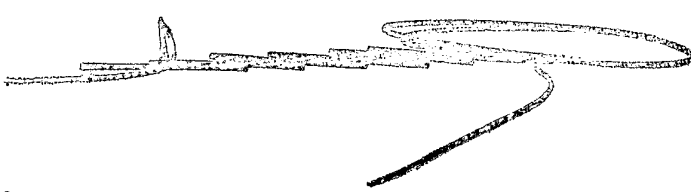
Argumentos

n-blks (entrada)

O número de megaoctetos contíguos que estão a ser devolvidos

vpn (entrada)

O número de página virtual da primeira página de 4K na zona que está a ser devolvida.



error-code (saída)

Um código de erro devolvido.

- e) Gain Freedom é uma sub-rotina que é chamada pelo programa START370. Ela coloca o programa START70 no modo supervisor S/88, de modo que podem efectuar-se as quatro sub-rotinas anteriores. Um vez que START370 esteja no modo supervisor, os apontadores de vectores podem ser modificados para retirar blocos de memória do sistema operativo S/88 e atribuir novamente a memória a cada processador S/370.

Esta sub-rotina é usada para alterar as atribuições de memória e para alterar os vectores manuais para o nível de interrupção 6 dos processadores S/88. Não é dado conhecimento aos utilizadores desta chamada, nem acesso à mesma, por razões de segurança.

Uso

```
declarar      S/370 gain-freedom entry (binary (15)
                                     binary 15);
               chamar S/370 gain-freedom (give-take,
                                     error-code);
```

Argumentos

give-take (entrada)

Um valor de 0 reenvia o chamador para o estado de aplicação do utilizador, e qualquer outro valor coloca o chamador no estado supervisor.

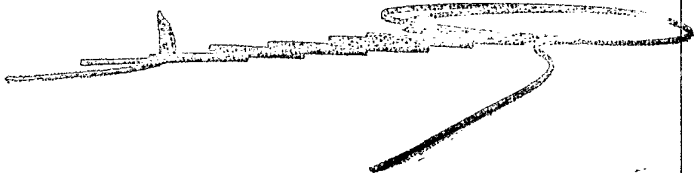
error-code (saída)

Um código de erro devolvido.

As fases de funcionamento das sub-rotinas anteriores são as seguintes:

S/370 Displace Storage

- 1) Ganhar a liberdade e bloquear a lista livre de matrizes de mme
- 2) Buscar a lista livre para a sequência maior de mme livres adjacentes
- 3) arredondar as duas extremidades para limites de MB e calcular o número de blocos (nblks), o número de blocos de 4 KB, em sequência
- 4) Se nblks > n_blks, ajustar nblks a n_blks (número de 4 KB pedido) e modificar o limite ppn de base

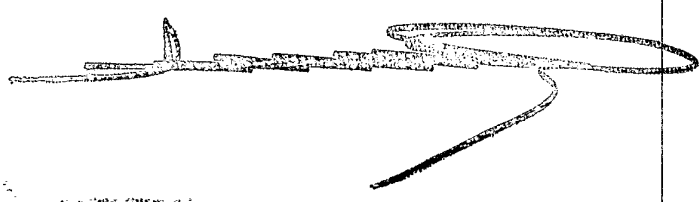
- 
- 5) Retirar a sequência escolhida de mme da lista livre
 - 6) Subtrair npages da contagem disponível do sistema
 - 7) Desbloquear a lista de mme_arrays e renunciar à liberdade
 - 8) Pôr: ppn = ppn de base
rc = erro se nblks < n_blks
rc = 0 se não houver qualquer erro.

S/370 Replace Storage

- 1) Verificar que todas as entradas não estão ligadas, ajustar os indicadores para zero e encadear entre si as mme de maneira apropriada. Devolver erro, se existir qualquer problema
- 2) Gain_freedom e bloquear a lista de mme_arrays
- 3) Buscar a lista livre para procurar a boa localização para introduzir as mme
 - a. Primeira candidata junto do ppn de base
 - b. Segunda candidata no fim da lista
- 4) Introduzir todo o bloco na lista livre
- 5) Adicionar npages de novo nas contagens disponíveis no sistema
- 6) Desbloquear a lista mme_arrays e renunciar à liberdade.

S/370 Open Storage

- 1) Procurar a entrada da tabela deste processo e procurar um furo na sua memória virtual, num limite pmb, suficientemente grande para o n_blks de MB. Assegurar-se de que há mme deslocadas suficientes para o serviço pedido. Devolver erro se houver algum problema.
- 2) Se necessário, atribuir espaço cablado para os pmb e as apte
- 3) Estabelecer toda a estrutura
 - mme cabladas e ligadas
 - mme.aptep- < apte
 - pme.aptep- < apte
 - todos os indicadores estão ajustados apropriadamente
 - apte.ptep- < pte
- 4) Ligar a cadeia pmb construída de novo com a cadeia



pmb de tarefas.

Close Storage

- 1) Procurar a entrada da tabela deste processo e procurar os pmb construídos por stopen_storage. Retroceder se não se encontrar nenhum.
- 2) Desligar estes pmb da cadeia de pmb do processo.
- 3) Para cada apte, chamar setup_ptw para censurar a distribuição de posições real
- 4) Devolver o espaço cablado para pmb e apte para o sistema operativo.
- 5) Devolver as mme para a rotina DISPLACE_STORAGE.

Gain Freedom

- 1) Obter o endereço do argumento give-take
 - 2) Passar à fase 7, se se renunciar à liberdade
- Se fases seguinte ganham a liberdade
- 3) Executar uma transferência (13), que faz com que o sistema operativo regresse ao chamador enquanto no estado supervisor.
 - 4) Obter o endereço da pilha do utilizador e permutar com o apontador da pilha do sistema.
 - 5) Conservar o endereço da pilha do sistema no apontador da pilha do utilizador
 - 6) Regressar ao chamador no estado supervisor na pilha do utilizador

As fases seguintes renunciam à liberdade.

- 7) Reaver o endereço da pilha do sistema e fazer a permuta com o apontador da pilha do sistema
- 8) Recolocar o endereço da pilha do utilizador no apontador da pilha do utilizador
- 9) Modificar a pilha de modo que o processador de transferências retorne à fase 11
- 10) Retornar ao processador das transferências
- 11) O processador das transferências retorna ao utilizador
- 12) Retornar ao chamador no estado do utilizador da pilha do utilizador.

5. Retirar a sequência de mme escolhida da lista livre

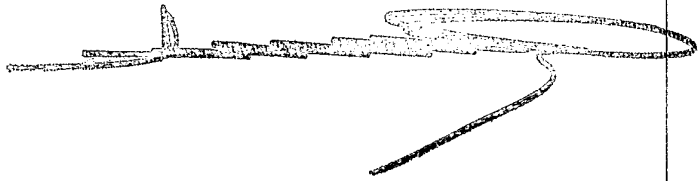
FIRST MME pertence à primeira mme na sequência que deve ser retirada, o ppn de base contém o seu ppn (número da página física). LAST MME pertence à última entrada mme na sequência. Se a FIRST MME estiver na cabeça da lista (o seu campo mme anterior igual a zero) o apontador da lista livre é ajustado igual ao campo next_mme da LAST MME; assim, a mme seguinte da LAST MME está agora na cabeça da lista. Por outro lado, o campo da mme seguinte da mme anterior à FIRST MME é ajustada igual ao campo next_mme da LAST MME. SE houver mme a seguir a LAST MME (o seu campo mme seguinte não zero), o campo mme anterior da mme a seguir a LAST MME é ajustado igual ao campo da mme anterior da FIRST MME.

6. Escrita da base e da dimensão da memória para a STCI

Depois de ter sido "roubada" do sistema operativo S/88, a memória é dividida entre os processadores S/370, de acordo com os requisitos estabelecidos no ficheiro de configuração. Forma-se uma matriz de configuração da memória nuclear S/88 contendo o ppn básico e o n blks para cada processador S/370. O termo n blks significa o número de mega-octetos contíguos de memória. É igual ao número de mme (não retiradas) roubadas, dividido por 256. Quando se inicia a tarefa EXEC370 para cada processador S/370 no seu processador S/88 respectivo, ela usa os valores do ppn de base e de n blks correspondentes para reunir uma palavra da interface STCI. Esta palavra é depois escrita no endereço virtual 007E01FC (no espaço de endereços da memória local (210) provocando a inicialização dos registadores (404) e (405) (fig. 32B) da STCI transparentes ao sistema operativo S/88.

O mecanismo de desacoplamento (216) e a lógica (253) da interface BCU, descrito mais atrás com referência às fig. 19A, 20, é usado para inicializar os registadores (404,405).

Porém, na forma de realização preferida,



como se mostra na fig. 32B, os registadores (404,405) são aco-
plados directamente à linha de dados do processador S/88 (161D)
(em vez de à linha omnibus de dados (223) local BCU). A lógica
de descodificação (280) da lógica (216) descodifica o endereço
virtual referido atrás para bloquear AS proveniente dos meios
físicos S/88 e devolver DSACK para o processador (62). Os regis-
tadores (404) e (405) são activados através da linha de selec-
ção STCI (458) proveniente da lógica (253). Os bits 27-20 da pa-
lavra da interface STCI formam o endereço "básico" da STCI e os
bits 23-20 formam o valor da "dimensão" da memória S/370. Os
bits 19-0 são zeros.

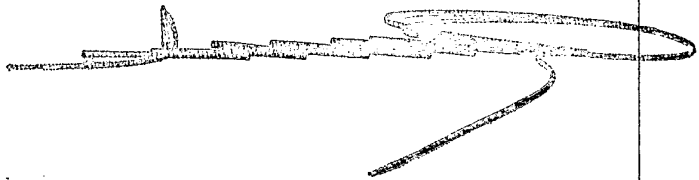
Funções de inicialização para as interrupções S/88 iniciadas pelo S/370

Há vários cenários para dirigir as in-
terrupções S/370 para o ou os processadores de interrupções re-
sidentes no S/88 sem o conhecimento do sistema operativo S/88.
Descrevem-se três deles.

Um primeiro processo implica a modifica-
ção do núcleo do sistema operativo S/88, introduzindo o código
do processador de interrupções S/370 no processador de interrup-
ções do primeiro nível do sistema operativo S/88, de modo que
ele é reunido como parte desse módulo objecto. A tabela de vec-
tores de interrupção está contida na fonte de assemblagem do
processador de interrupções e os vectores usados pelo S/370 são
modificados na fonte para apontar para o código do processador
de interrupções S/370.

Este processo difere grandemente do pro-
cesso S/88 architectado da seguinte maneira:

- 1) Cada dispositivo de interrupção deve ser introduzido no fi-
cheiro que o identifica, o nome do seu trajecto, o endereço
do seu cartão, etc. para o sistema operativo S/88.
- 2) Quando o processador de interrupções do primeiro nível rece-
be a interrupção, ele estabelece as pilhas formatadas apro-
priadas, conserva todos os registos e registadores da máqui-
na, verifica a validade da interrupção e passa a interrup-



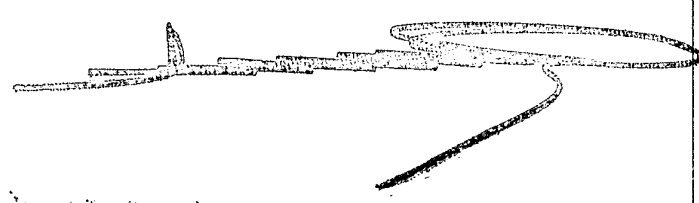
ção para um processador de interrupções "de segundo nível", que chama o código de interrupção do dispositivo escrito especificamente pelo programador.

- 3) Quando tiver terminado o código de interrupção, ele devolve o controlo para o processador de interrupções do sistema operativo que toma a seu cargo os ambientes de restauração.

O primeiro processo atrás referido contorna tudo isto. Reunindo os vectores de interrupção S/370 para apontar para a rotina de interrupção S/370, evitámos todo o processamento de interrupção normal efectuado pelo sistema operativo S/88 e não temos de identificar o S/370 através do ficheiro do dispositivo. Isto é na realidade um desacoplamento dos meios de programação, visto que o código foi antes modificado pelos meios físicos. Este primeiro processo é o mais rápido e o menos dispendioso para conseguir a função de interrupção desejada. Porém, este processo é susceptível de uma manutenção adicional para cada libertação subsequente do sistema operativo S/88. Ele exige pelo menos uma ligação nuclear e, se se tiver alterado o processador de interrupções, o código de interrupções S/370 tem de ser reinserido e de novo reunido o processador de interrupções.

Um segundo processo implica modificações dos vectores de interrupção do sistema operativo depois da inicialização do sistema, sendo este o processo implicado na descrição do mecanismo de interrupção por meios físicos da fig. 20.

Este segundo processo exige a colocação do código de interrupção S/370 no espaço de endereços virtuais do sistema operativo S/88 (na forma de realização preferida, precisamente por baixo de 007E0000) e a modificação dos vectores de interrupção apropriados no processador de interrupções do núcleo do sistema operativo. Este trabalho é feito pela rotina de inicialização S/370 depois de o sistema operativo se ter inicializado (ao mesmo tempo que a rotina de inicialização S/370 "rouba" memória). Como a rotina de inicialização está a modificar a área de memória do núcleo do sistema operativo S/88, ele tem de "ganhar liberdade" da maneira descrita para "roubar" memória na descrição anterior. Este segundo processo não exige a



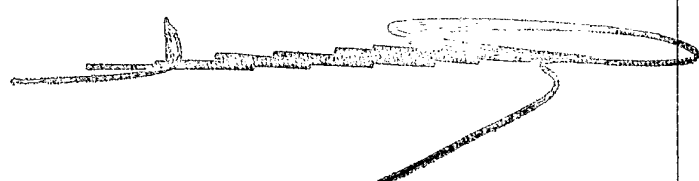
modificação da manutenção de cada núcleo do sistema operativo S/88 desenvolvido de novo. Porém, as interrupções S/370 não são funcionais até depois de o sistema operativo S/88 estar de pé e a funcionar.

Um terceiro processo implica a apresentação em meios físicos do conteúdo dos vectores de interrupção, sendo esta uma alternativa preferida porque não exige qualquer alteração no núcleos do sistema operativo S/88, isto é, não se faz qualquer alteração da tabela de vectores.

Este terceiro processo exige a colocação da rotina de interrupção S/370 no espaço de endereços virtuais do sistema operativo S/88 e/ou na memória local BCU, como um endereço conhecido de memória só para leitura (ROS). O ou os endereços da rotina de interrupção têm de estar disponíveis para os meios físicos S/370, de preferência em memórias ROS. O cenário seguinte é dado para ilustrar este processo:

- 1) O S/370 (por exemplo o controlador DMAC (209) na BCU (156)) activa o pedido de interrupção.
- 2) A unidade de processamento S/88 (62) activa a confirmação de interrupção, os impulsos de sincronização de dados e os impulsos de sincronização de endereços.
- 3) A BCU coloca um número de vector de interrupção (poderia ser tudo zeros, para fácil reconhecimento ou desvio para o espaço de vectores da nossa ROS) na linha omnibus de dados (223) e activa a confirmação de impulsos de sincronização. Este número de vector não tem, excepto para a paridade válida, qualquer consequência para o processador (62).
- 4) Eventualmente o processador (62) efectuará um ciclo de leitura da memória para obter o vector de interrupção de 4 octetos.
- 5) A BCU reconhece este acesso específico da memória (pelo endereço virtual), desacopla o processador (62) da memória de acesso e apresenta o seu próprio vector de interrupção de 4 octetos (encaminhado a partir da memória ROS). A memória ROS do S/370 contém tantos vectores quantos os necessários, vários para o DMAC, um para a sincronização da ROS, etc..

Este terceiro processo permite o desacoplamento durante a sincronização do cartão, para fins de



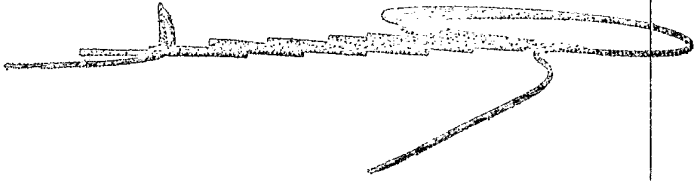
sincronização dos meios físicos S/370, etc.. Contudo, este processo exige mais meios físicos.

Ganho da liberdade sem modificar o sistema operativo S/88

Foi indicado atrás, em "Rotina de serviço de arranque S/370" um processo que descreve como o programa de aplicação pode ganhar a liberdade, isto é, obter o estado supervisor. Esse processo implica a escrita de uma rotina de chamada de serviço especial do sistema operativo "instrução de transferência 13" a adicionar ao núcleo do sistema operativo S/88.

Esta rotina de interrupção de transferência 13 não faz mais que "chamar" o programa criando a transferência na localização imediatamente a seguir à instrução da transferência. Como a rotina de interrupção de transferência está no estado supervisor, o programa em questão mudará para o estado supervisor. Para retomar o estado de programa de aplicação, o programa de aplicação modifica o endereço de retorno à pilha de interrupções e passa da "chamada" para a transferência 13 para o código de interrupções de transferência 13, que sai da interrupção usando o endereço modificado da pilha de interrupções. Este processo implica a adição de uma rotina de interrupção ao sistema operativo S/88.

Um segundo processo elimina a modificação do sistema operativo em questão. Define-se um registador especial (não representado) no espaço de endereços da memória de comando BCU que, quando gravado pelo programa de aplicação, provoca uma nova interrupção BCU, utilizando o terceiro processo para realização de interrupções atrás indicado. A rotina de interrupção de aplicação faz-se residente na memória só para leitura da BCU (não representada) e funciona da mesma maneira que o código da transferência (13). A rotina de Ganho da Liberdade atrás descrita funciona exactamente da mesma maneira, excepto que grava o registador da BCU especial em vez de criar uma instrução de transferência 13.



Roubo de memória sem modificar o sistema operativo S/88

Utilizando esta segunda forma de realização de Ganho da Liberdade, o "roubo de memória" não exige a reunião de qualquer código fonte de S/88 ou a ligação do núcleo do sistema operativo S/88. O endereço da cabeça da lista livre está disponível para o programa de aplicação.

Ligação da energia e sincronização de unidades simples e unidades emparelhadas (21,23) (Fig. 49,50)

Unidade de processamento S/88 como Processador de Serviço para a Unidade de Processamento S/370

1. INTRODUÇÃO

Esta secção descreverá resumidamente, com referência às fig. 49 e 50, alguns dos registadores, reten-
tores e lógica dos meios físicos que determinam o estado de, e controlam e ajustam o ambiente para a sincronização de unidades parceiras, tais como (21,23) da fig. 7.

Adicionalmente, descrever-se-ão certas funções de microcódigo para efectuar a inicialização, a sincronização e a ressinchronização de unidades simples ou emparelhadas, Dirige-se a atenção primeiro para o sistema S/88 (a forma de realização preferida) que funciona essencialmente sem alteração, relativamente à inicialização e à sincronização das unidades S/88, quer em ambientes de unidades simples, quer emparelhadas. Este processo de funcionamento será descrito apenas resumidamente. Além disso, serão aqui repetidas certas partes da descrição da patente de Reid.

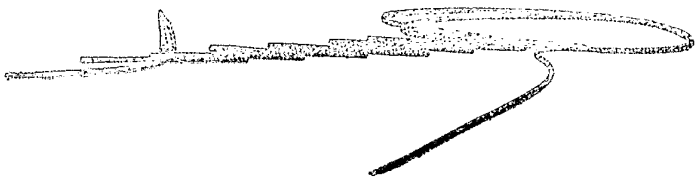
A verificação de erros é efectuada ao mesmo tempo que cada elemento de processamento S/88 (60,62) (fig. 8) da unidade (21) excita a linha omnibus A (42) e a linha omnibus B (44). Esta operação concorrente está em contraste com as unidades I/O no módulo processador (9) que realiza praticamente uma verificação de erros antes de excitar a estrutura de linhas omnibus. A unidade de processamento (21) opera desta



maneira porque a distribuição de tempos na mesma é suficientemente importante para que qualquer atraso no funcionamento seja indesejável para o caudal de transmissão do sistema. Um erro notado pela lógica de verificação durante o tempo em que a unidade de processamento está a actuar na estrutura de linhas omnibus faz com que a unidade aplique, quer um sinal de erro na linha omnibus A, quer um sinal de erro na linha omnibus B, à linha omnibus X (46), durante a fase seguinte do relógio do sistema.

Durante a mesma fase de relógio, a unidade de processamento central avariada (por exemplo a (21)) acciona uma interrupção de manutenção de nível 1, na linha omnibus X (46), que é recebida pela unidade de processamento central parceira (por exemplo a (23)). No fim dessa fase de relógio, a unidade avariada sai de linha, tornando-se incapacitada para enviar outros sinais para a estrutura de linhas omnibus (30), excepto em resposta a uma interrogação a partir da unidade de processamento central parceiro. Este funcionamento fora de linha automático assegura que qualquer ciclo de leitura ou de escrita será abortado, quer para a unidade de memória (16, 18), quer para um dispositivo periférico através da unidade de controlo durante o qual se detectou um erro, quer no endereço, quer nos dados, para as linhas omnibus A ou B. Além disso, qualquer transferência de dados durante o mesmo ciclo de operação é repetida utilizando apenas a unidade de processamento central parceira.

Mais especificamente, o comparador (12f) compara os dados de entrada que a secção de processamento (12a) recebe da linha omnibus A (42) com os dados de entrada que a secção de processamento (12b) recebe da linha omnibus B (44). Compara também os sinais de função, de endereço e de dados (incluindo a paridade) que a secção de processamento (12a) aplica aos emissores-receptores com sinais correspondentes, que a secção de processamento (12b) produz. Os sinais de distribuição de tempos e de controlo da secção (12a) são comparados com os sinais correspondentes provenientes da secção (12b). Esta comparação de sinais de controlo internos verifica as operações internas dos elementos de processamento (60,62) e facilita a detecção pronta de avarias, sendo útil para o diagnóstico e a manu-

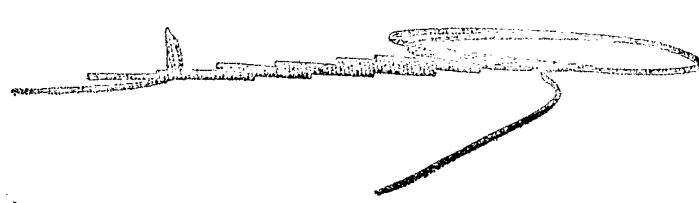


tenção da unidade do processador.

Em qualquer instante em que um ou mais sinais de entrada correspondentes para o comparador (12f) difiram, o comparador produz um sinal de Erro de Comparação, que é aplicado ao andar de controlo (86). O erro pode ser o resultado de introdução de dados, um erro de exclusão de dados, um erro de função ou um erro de controlo. Pode também ser ou um erro de ciclo ou um erro de controlo devido a tempos ou sinais de controlo diferentes. A detecção de um erro por circuitos de controlo de paridade produz um erro de paridade, que é aplicado ao andar de controlo (86). O andar de controlo (86) responde ao sinal Compare Invalid e o ao sinal Parity Invalid para produzir, na fase de relógio seguinte ($N + 1$), um Erro de Processador. Verifica-se uma excepção a este funcionamento se o sinal Compare Invalid for devido a uma comparação inválida de sinais de dados durante uma operação de leitura. Nesse caso, o andar de controlo (86) produz o sinal de Erro de Processador apenas se não se tiver produzido nenhum Erro de linha Omnibus com a fase de tempo seguinte. Um sinal de Erro de Linhas Omnibus indica uma condição de avaria na estrutura de linhas omnibus (30) e portanto identifica que a comparação inválida de dados de entrada foi o resultado de uma avaria na parte de linha omnibus A ou na parte de linha omnibus B da estrutura de linhas omnibus (30) e não na secção de processamento (12a), nem na (12b).

Uma função do sinal de Erro do Processador consiste em desactivar os circuitos lógicos e portanto parar substancialmente o funcionamento na secção de processamento (12) da unidade (21). Além disso, o sinal de erro na Linha Omnibus A e o sinal de Erro na Linha Omnibus B são aplicados na linha omnibus X (46) para sinalizar para todas as unidades no módulo (9) que devem ignorar a informação colocada na linha omnibus durante a fase imediatamente anterior, por exemplo, para ignorar a transferência na linha omnibus CPU. É aplicado um sinal de interrupção de nível 1 na linha omnibus X (46) para notificar a unidade de processamento parceira (23) que alguma unidade de no módulo detectou um erro que produziu uma avaria.

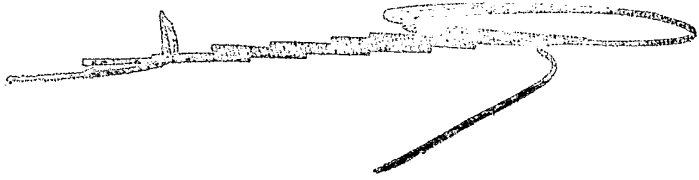
No início da fase ($N + 2$), o andar (86) ainda em resposta ao sinal de avaria, termina o estado asserti-



vo de direcção das linhas omnibus. Esta acção é acompanhada pela terminação dos sinais de Erro das linhas Omnibus. Quando a secção de processamento (12) se desliga do estado Director, ele desactiva todos os excitadores das linhas omnibus nos emissores-receptores (12e). Os excitadores dos emissores-receptores S/370 (13) são também desactivados através do controlo comum (75), sempre que esses emissores-receptores (12e) forem desactivados. Analogamente, no caso de ser produzido um sinal de Erro do Processador por um andar de controlo (75) da unidade (21), os emissores-receptores (12e), através do andar de controlo (86), e os emissores-receptores (13) são também desactivados.

Assim, as unidades de processamento (21,23) podem activar a estrutura de linhas omnibus apenas quando no estado Director, como é necessário para produzir o sinal Bus Enable que é aplicado aos activadores. O sinal de Erro do Processador muda prontamente, isto é, no fim da fase temporal seguinte, o estado Director. No caso de a secção de processamento (12) da unidade (21) produzir um sinal de Erro de Processador, a secção de processamento S/88 da unidade parceira (23) continua a funcionar substancialmente sem interrupção. Quando o sinal de Erro do Processador se verificar durante uma operação de escrita, a unidade processadora parceira (23) repete a transferência de dados. Quando o Erro de Processador surgir durante uma operação de leitura, a unidade parceira lê os dados repetidos que a memória aplica à estrutura de linhas omnibus na fase temporal subsequente.

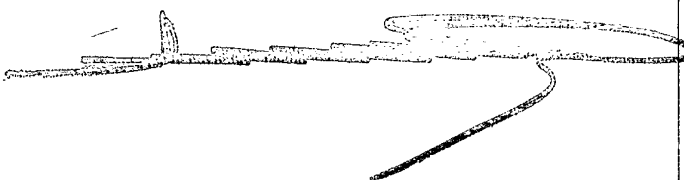
Além disso, a unidade de processamento parceira (23) responde à interrupção do nível 1, que é uma interrupção de baixa prioridade, para iniciar uma rotina de diagnóstico. No caso de do Erro do Processador parecer ser um fenómeno transitório, isto é, se a rotina de diagnóstico não identificar nem localizar qualquer avaria ou condição de erro, a unidade de processamento (21) pode ser restaurada para funcionamento sem manutenção. Numa forma de realização preferida, a ocorrência da avaria transitória é registada e se for repetida um certo número de vezes, que pode ser arbitrado, a unidade de processamento é retirada electricamente de serviço ou operação sem diagnósticos ulteriores.



Cada secção de processamento (12) das unidades (21,23) inclui circuitos lógicos, tipicamente no andar (86) de estados e controlo do processador, para levar as duas unidades parceiras à sincronização por arranque e paragem. A secção (12) atinge a sincronização por arranque e paragem sem a transição pelo estado Director. Cada secção (12) tem de estar no estado Director para aplicar sinais à estrutura de linhas omnibus. A sequência de inicialização armazenada em cada PROM (181) inclui tipicamente instruções para levar as secções emparelhadas ao sincronismo e para assegurar que nenhuma secção de processamento está inicialmente, isto é, depois de ser ligada, no estado Director.

As secções de processamento (12) das unidades (21,23) não estão em sincronização, inicialmente, na sequência de inicialização e uma unidade atinge o estado Director durante o ciclo multifásico antes do outro. A unidade que obtém o estado Director controla a operação de inicialização da outra unidade para a levar ao estado Director num instante seleccionado.

Quando a secção de processamento (12) da unidade (21) for inicializada, ela nega um sinal interno de Verificação de Erro, impedindo assim que um sinal de Paridade Inválida ou um sinal de Comparação Inválida de produzir um sinal de Retenção do Processador. Em vez disso, a secção (12) executa uma rotina de ensaio, tipicamente armazenada na PROM (181), que experimenta todas as condições que podem produzir um sinal de Erro de Processador. À medida que se vai criando cada condição potencialmente de avaria, a secção de processamento faz o ensaio para ver se é de factor produzido o sinal de participação da avaria correspondente. A ausência do sinal de Verificação de Erros inibe assim a unidade de processamento de atingir o estado Director, com o resultado de que as avarias produzidas durante a rotina de experiências lógicas não interrompem a unidade de processamento e não são comunicadas à estrutura de linhas omnibus (30). A rotina de ensaio na PROM (181) afirma o sinal de Verificação de Erro e activa o processador para assumir o estado Director apenas quando se completar com êxito essa rotina de verificação.

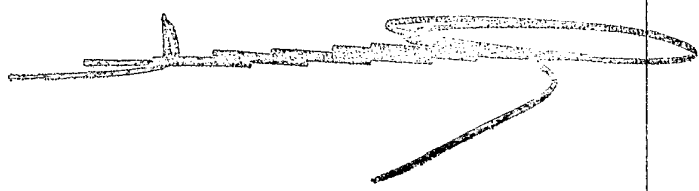


As unidades de processamento S/370 (a forma de realização preferida) têm tipicamente meios físicos proporcionados para as funções de inicialização e serviço do processador através de um acesso "pela porta das traseiras" aos vários componentes e à lógica em cada micropastilha. Como isto é bem conhecido, apenas se descreve resumidamente.

Rotinas de programas para o ensaio automático e a inicialização são igualmente bem conhecidas, não sendo necessário descrevê-las aqui em pormenor. O que se realça nesta secção é o mecanismo por meio do qual o ensaio automático e a inicialização do S/370 é conseguido através do S/88, sem que os sistemas operativos, nem do S/370, nem do S/88, tenham conhecimento das permutas. As rotinas de ensaio automático e inicialização (STIO) para o S/370 são colocadas na PROM (181) (fig. 19C), na forma de realização preferida, juntamente com as rotinas para a sincronização dos elementos de processamento S/370 nas unidades parceiras, O S/88 funciona portanto como processador de serviço do S/370. As atribuições I/O nas posições de memória do código S/88 na PROM (181) são proporcionadas no caso de certos estados S/88, ou outros conteúdos de registos, serem necessários para a realização do código S/370.

A maneira como este código trata a sincronização consiste em transferir uma cópia com as posições na memória do conjunto de registadores no interior de uma unidade de processamento parceira primária (ou directora), tal como (21) (uma unidade que esteja a funcionar apropriadamente) para o conjunto dos registadores no interior da unidade de processamento parceira secundária (ou dirigida), por exemplo (23) (uma unidade que ainda não está a funcionar apropriadamente).

Antes de descrever os pormenores do tracto de acoplamento do S/88 ao S/370 para o mecanismo de sincronização, vai dar-se uma breve vista de olhos pela estrutura e o ambiente do módulo (9) da fig. 7. As características do sistema operativo S/88 tais como a tolerância das avarias e a imagem de sistema único são preservadas para ambas as estruturas S/88 e S/370. O módulo (9) é constituído por uma ou mais unidades de processamento S/370 simples, tais como (21), ou pares de unidades de processamento duplicadas, tais como (21,23). Podem



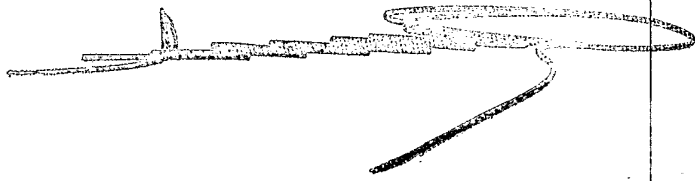
incluir-se unidades S/88 simples ou duplicadas, tais como (12) ou (12,14) no módulo para executar apenas programas S/88.

Cada unidade de processamento S/370 inclui dois elementos processadores S/370, tais como (85,87) e dois elementos processadores S/88 tais como (62,64), como se mostra na fig. 7, sendo os pares de elementos de processamento operados com sincronização por arranque e paragem, como uma unidade de processamento lógica simples. As unidades emparelhadas formam um dispositivo redundante, operado em arranque e paragem, entre si, para proporcionar uma unidade de processamento lógica com auto-verificação completamente tolerante das avarias.

Cada um dos elementos processadores S/370 (85,87) de um par é, em parte, um conjunto de micro pastilhas S/370, tal como (150) (fig. 11). Os conjuntos de micropastilhas S/370 e o seu equipamento físico associado estão montados num cartão tipo S/38, tal como (101) (fig. 9A) para acoplamento com a estrutura de linhas omnibus S/88 (30); e estão acoplados a elementos de processamento S/88 respectivos através de circuitos lógicos de interface (89) e (91) (fig. 8). Nesta secção, o par de conjuntos de micropastilhas S/370 e o seu equipamento físico associado numa unidade de processamento, tal como (21), serão designados como uma entidade S/370 e os seus elementos de processamento S/88 correspondentes, tais como (60,62) e os equipamentos físicos associados serão designados por entidade S/88. As entidades S/370 executam programas de aplicação e chamam entidades S/88 para executar as operações I/O S/370, conforme for necessário, utilizando os dispositivos I/O e os programas S/88, de modo que os sistemas operativos, quer do S/88, quer do S/370, não têm conhecimento um do outro.

2. SINCRONIZAÇÃO DOS MEIOS FÍSICOS TOLERANTES DAS AVARIAS

Uma das características únicas e mais significativas das unidades de processamento S/88-S/370 é a sincronização auto-determinada de qualquer unidade de processamento, tal como (21), por um parceiro que simultaneamente está em processamento (23). A entidade S/88 de cada unidade tem a capa-



cidade e a responsabilidade da sincronização de um parceiro novo ou avariado. Quando uma entidade S/88 de uma unidade assume esta responsabilidade, designa-se como a "directora". A sua parceira, que sofre a sincronização, é designada por "dirigida".

A estrutura de meios físicos/meios de microprogramas S/88 determina quando é necessária a sincronização e quem sincroniza quem. Os meios físicos/meios de microprogramas S/88-S/370 interligados utilizam esta mesma inteligência para seguir a condução do S/88 nas decisões de sincronização. Isto é, em qualquer instante, em que o S/88 determina que uma entidade S/88 (secundária) exige sincronização com a sua parceira (directora), permite-se que essa sincronização progrida até um ponto apropriado depois de a entidade dirigida S/88 ter sido "afastada"; depois, a execução é desviada para a entidade S/370 correspondente. As entidades S/370 são sincronizadas pelos elementos processadores S/88 executando códigos provenientes da PROM (181) para extrair o estado Director S/370 e restaurar esse estado em ambos os parceiros S/370.

Uma qualquer do par de entidades parceiras pode assumir o papel de directora ou de dirigida na sincronização das unidades de processamento, quer o pedido seja feito por uma Ligação Inicial da Energia, pelo aparecimento de uma nova unidade parceira ou por perda de sincronismo (qualquer dos casos forçando uma interrupção). Em cada caso, a entidade S/88 dirigida reconhece o seu estado e depende da entidade S/88 directora para a sincronização.

As entidades S/88 directora e dirigida assumem os seus papéis respectivos como resultado dos seus estados respectivos no instante em que ocorre a interrupção de manutenção. As entidades S/88 de todas as unidades de processamento detectam e processam a interrupção, admitindo cada uma delas que é uma unidade dirigida até ser estabelecida uma unidade directora, sem avaria. A unidade directora afasta então qualquer dirigida que se mantém em sincronização de arranque e paragem, retomando cada uma delas o ambiente pré-esvaziado da directora (depois de retornar da interrupção).

Analogamente, as entidades S/88 desacoplarão os processadores do resto da lógica, utilizam esses pro-



cessadores para emular o funcionamento do sistema operativo S/370 para estabelecer um estado pré-esvaziado idêntico no interior do par de parceiros S/370, restabelecendo depois o ambiente de execução normal e permitirão que o par de parceiros S/370 comecem a execução em arranque e paragem.

Uma situação que não exige sincronização:

Uma unidade de processamento simples é ligada a energia, isto é uma unidade tal como (21);

As situações que exigem sincronização são:

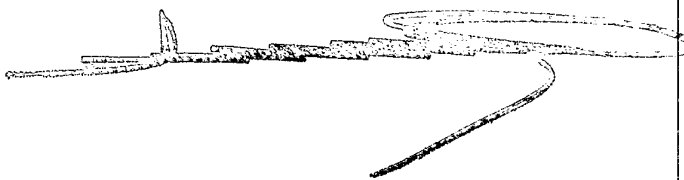
Unidades de processamento duplicadas (por exemplo 21,23) são ligadas à energia:

A unidade (21) é inserida enquanto a sua parceira (23) processa normalmente; e

Uma unidade de microprocessamento tal como (21) detecta uma comparação com resultado negativo na sua parceira (23) e tenta a recuperação.

A entidade S/88 tem facilidades de equipamentos físicos apropriados para estabelecer a sincronização. A secção de processamento S/370 tem meios físicos suficientes e os meios de programação ajudam a permitir que uma entidade dirigida seja inicializada exactamente no mesmo estado que a entidade directora. Isso inclui características tais como registadores de estados de leitura/escrita, registadores de modos susceptíveis de ser lidos, memórias "cache" susceptíveis de ser limpas, relógios susceptíveis de ser parados e anéis de contagem, etc..

Quando uma entidade S/370 que opera normalmente na unidade (21) deve ser posta em sincronismo com a sua entidade S/370 correspondente numa unidade parceira (23), é necessário levar a unidade parceira S/370 ao mesmo estado que a unidade que opera normalmente. Este processo é simplificado na forma de realização preferida, emitindo uma QUEUE SELECT UP MESSAGE dos processadores S/88 (60,62) (sob o controlo do microcódigo S/370 de inicialização e sincronização na PROM (181)) para os processadores S/370 (85,87). Esta mensagem interrompe as aplicações do utilizador para não fazer outros pedidos de servi

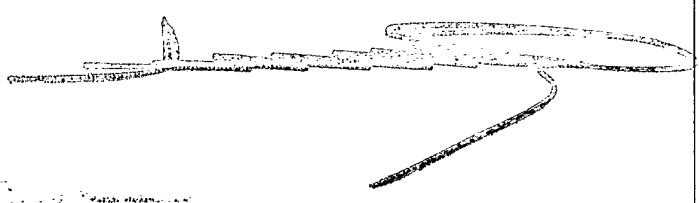


ço através do Sistema Operativo, para as BCU, tais como (156), durante o tempo de sincronização. Permite também que se complete a execução de todas as operações I/O incompletas.

Isso leva a entidade S/370 que funciona normalmente a um estado que é copiado para a memória (162), para ser usado pelas duas entidades S/370, depois da "separação" ("kickoff"). Nesta altura todos os registadores, contadores, apontadores e memórias tampões (contexto) no processador S/370, a memória "cache" S/370, a DLAT e o adaptador de linhas omnibus S/370 são copiados para a memória (162), numa pilha ordenada. Quando o processo de sincronização se inicia, todos os quatro processadores físicos terão o contexto S/370 restaurado, carregando esse contexto em todos os quatro processadores a partir da pilha comum. Ambos os processadores serão carregados com dados idênticos para os seus registadores, contadores e memórias tampão, iniciando então a execução do programa em sincronismo de arranque-paragem ou sincronismo completo.

A entidade de processamento S/370 proporciona dois processos para aceder aos vários registadores e "memórias "cache" para sincronização. Um é o processo normal de leitura/escrita programado pelo utilizador, usando registadores (560,561) (fig. 49) que acoplam a linha omnibus de dados BCU local (223) aos canais 0 e 1 do adaptador (154). O outro é uma solução em série "pela porta traseira", a Facilidade de Suporte Integrada (ISF)/Interface de Suporte Integrada (USI) (540,541). Emulando a interface/protocolo SISF/USI em série do processador de serviço do conjunto de micropastilhas S/370, o mecanismo de sincronização das entidades S/88 podem aceder a qualquer e a todas as facilidades associadas com a entidade S/370. Quando for necessária a sincronização de uma ou mais entidades S/370, podem empregar-se ambos os processos. O trajecto normal é usado quando ele existir e o trajecto USI é usado nos casos restantes.

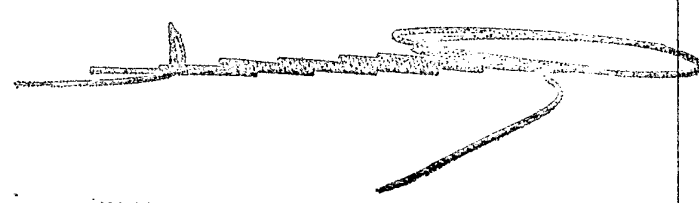
É importante notar que esta parte do processo de sincronização e inicialização (isto é, para as entidades S/370) tem de ser transparente para o sistema operativo S/88 que não tem conhecimento da presença da, ou da ligação a, uma entidade S/370. Esta transparência é conseguida de uma maneira genericamente semelhante à descrita atrás em relação às



operações I/O S/370. Isto é, a lógica de descodificação (280) descrita com referência à fig. 20, detecta um endereço 007EXXXX de cada vez que devem transferir-se dados entre o processador S/88 (62) e a lógica da fig. 49. Quando este endereço for descodificado pela lógica (280), ele acopla a linha omnibus do processador S/88 (1161A,161D) ao endereço BCU local e às linhas omnibus de dados (247,223) através dos circuitos (217,218), como se descreveu antes. A lógica de descodificação dos endereços (562) dos registadores descodifica os bits de ordem inferior do endereço na linha omnibus (247), para seleccionar um dos circuitos lógicos (549,550) ou registadores (560,561) para a transferência de dados com o processador (62).

Além disso, as interrupções nas linhas (562) e (563) são dirigidas para a lógica de interrupções S/88 (212) da fig. 20, através do circuito OU (292a). O sinal de pedido de interrupção é activado na linha (562) quando se tiverem recebido dados na lógica (549) provenientes das micropastilhas S/370 para transferência para o processador (62). Um sinal de pedido de interrupção na linha (563) notifica o processador sobre ter-se completado a transferência de dados da lógica (550) para uma micropastilha S/370. Um pedido de interrupção na linha (562) notifica o processador (62) de que se receberam dados pela lógica (549) provenientes de uma micropastilha S/370 para a transferência para os processadores (62). Os pedidos de interrupção são mantidos nas linhas (562) e (563) até aparecer um sinal IACK nas linhas (258c) e (258d), respectivamente. Os números de vectores para estas interrupções são derivados da lógica (564, 565) quando excitados por sinais IACK (258d) e (257e), respectivamente, vindos da fig. 20. Os números dos vectores são usados pelo elemento processador (62) para aceder às rotinas de tratamento de interrupções respectivas.

A facilidade de suporte integrada (ISF) (540) (fig. 49) representa uma entrada "pela porta traseira" para a lógica no conjunto de micropastilhas (150). A ISF consiste numa linha omnibus de suporte (541) de 5 linhas que liga às interfaces de suporte das unidades (USI) integradas nas micropastilhas (85) e (151-154). Na fig. 49 está representada uma parte da USI (542).



A linha omnibus de suporte (541) representa uma interface em série com as seguintes 5 linhas:

BIT OUT (dados para o conjunto de micropastilhas) linha (543)

BIT IN (dados do conjunto de micropastilhas) linha (544)

ADRESS MODE (controlo) (linha 545)

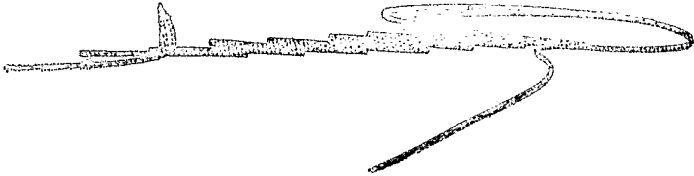
SHIFT GATE (controlo) linha (547)

A linha ADDR_MODE (545) sinaliza a transferência em série (deslocamento) quer de bits de endereço (nível superior) ou bits de dados (nível inferior) nas linhas BIT OUT/BIT IN (543,544). As linhas BIT OUT e BIT IN linhas (543, 544) são a interligação entre os registadores de deslocamento (SR), tais como (548), no interior de uma micropastilha e os registadores de deslocamento, (tais como (549,550)). O número de bits deslocados entre um registador interno (548) e um dos registadores externos (549,550) é determinado pelo número de impulsos aplicados à porta de deslocamento (SHIFT GATE), linha (546).

O SET PULSE é usado para sincronizar as actividades internas das micropastilhas com base nos endereços ou nos padrões de dados precisamente deslocados para o interior da micropastilha. SET PULSE é activado depois de terminado o deslocamento para sinalizar a disponibilidade da informação do lado da micropastilha, por exemplo no registador (548). Isto significa que as actividades baseadas na informação podem ser iniciadas por exemplo a partir deste momento.

O exemplo seguinte ilustra o funcionamento. Uma função de arranque é atribuída a um padrão de endereço específico. Este endereço é deslocado para o interior de registadores, tais como (548), de cada micropastilha. Quando todos os bits do endereço tiverem sido transferidos, o descodificador de endereços (551), numa das micropastilhas, detecta o seu endereço. À transferência do endereço segue-se o SET PULSE. A descodificação e o SET PULSE formam um impulso de arranque interno na saída da porta (552).

A parte específica da micropastilha de uma USI contém cadeias de controlos e dados, derivados da concepção específica da micropastilha. Para reter o estado corren-



te dos elementos de memória não afectados por uma operação de deslocamento, os relógios funcionais têm de ser parados antes da iniciação de quaisquer actividades USI. Os acessos à USI que necessitam da paragem do relógio como pré-requisito são definidos como "estáticos". Os acessos dinâmicos ou funções são as operações que podem ser executadas enquanto as micropastilhas estão em funcionamento.

O SET PULSE é usado para sincronizar as funções para as distribuições de tempos internas da micropastilha. As funções são descodificadas a partir do padrão de endereços ou do padrão de dados no registador SERDES, adicionalmente encaminhadas por portas pela linha ADDR MODE (modo de endereço ou de dados):

Ajustar o estado da micropastilha no SERDES

Ajustar o registador de modos no SERDES

Carregar o registador de modos a partir do SERDES

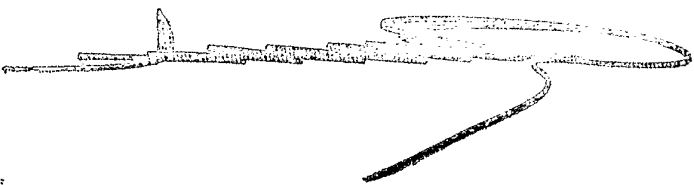
Ajustar o retentor de Pedido de Transferência de Suporte (SPR)

Repôr o retentor de Pedido Controlado do Processador (PCR)

Funções dinâmicas adicionais, como for necessário para apoiar as micropastilhas individuais

A linha omnibus de cinco fios em série (541) da ISF, que proporciona um acesso "pela porta traseira" para as várias entidades endereçáveis no interior do conjunto de micropastilhas S/370 (150) está acoplada à Interface de Suporte das Unidades (USI) de cada micropastilha, por exemplo USI (542) da pastilha (85). A USI (542) proporciona um Registador de Endereços de 8 bits (566) e um Seriador/Desseriador (SERDES) (548). O registador de endereços USI (566) recebe o endereço da micropastilha e o endereço da entidade objectivo no interior da pastilha, enquanto o SERDES (548) é o mecanismo actual de emissão/recepção. A USI também proporciona lógica de sincronização para o mecanismo de deslocamento de entrada/deslocamento de saída ("shift-in/shift-out").

Cada micropastilha no interior do conjunto de micropastilhas S/370 (150) é associada a um endereço ISF/USI de 4 bits (de ordem superior), por exemplo PE (85), ao controlador da memória "cache" (153), ao relógio (152), ao adap



tador (154), ao coprocessador em vírgula flutuante (151) e à interface STCI (155), sendo atribuídos os valores hexadecimais 2, 4, 6, 8 e A e B, respectivamente. Os 4 bits de ordem inferior do endereço ISF/USI definem a entidade da micropastilha interna (por exemplo, registador, função ou cadeia), endereçada pelos 4 bits de ordem inferior.

O esquema de comunicações é constituído por Cadeias de Deslocamento (denominadas também Cadeias de Funções) que, por sua vez, são constituídas por campos que identificam o comando, a micropastilha da fonte, a micropastilha do destino, os dados e a entidade objectivo no interior das micropastilhas. As cadeias de deslocamento são como segue:

- Bits 0-7 - Função/Comando
- 8-11 - Unidade (de controlo) da fonte
- 12-15 - Unidade (detectada/controlada) objectivo
- 16-23 - Mensagem/dados
- 24-27 - Registador (escrito) controlado
- 26-31 - Registador (lido) detectado.

Estas Cadeias de Função são designadas por Cadeias de Deslocamento por causa da natureza em série da ISF/USI e pelo facto de as cadeias terem de ser "deslocadas" para dentro/para fora da lógica (549,550) e dos registadores SERDES, tais como (548).

O Campo de Comando da Cadeia de Função pode conter um Comando de Escrita/Controlo (E61) ou um Comando de Leitura/Detecção (F61). Um exemplo de uma Cadeia de Função é como segue:

E602XX10 = Escrever para o Registador de Modo do processador (85)

onde E6 = comando = escrever

0 = endereça da fonte - PE (62) para ensaio

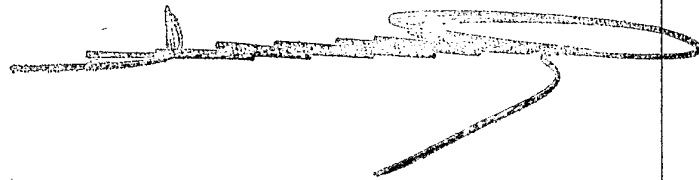
2 = Destino - PE (85)

XX = mensagem (dados)

1 = Registador Controlado (registador de modos)

0 = Registador de Detecção (nenhum, visto que o comando é uma "escrita")

As soluções para estabelecer a sincronização aqui descritas usam o código de programas S/88 armazenado



na PROM (181). O código faz determinações associadas com cada uma das quatro situações anteriores e ajusta indicadores em conformidade. As rotinas de sincronização utilizam então os indicadores para controlar o trajecto do código para efectuar a sincronização e/ou a inicialização apropriadas. Dois exemplos:

Determinar se ou não a memória num cartão particular S/88 estava comprometido por uma Falha de Energia e deve ser inicializado a partir do seu parceiro.

Determinar se ou não um cartão S/88 particular deve assumir o papel de DMPU (Defaulted Master Processing Unit - Unidade de processamento directora sem avaria).

As subsecções seguintes (3 a 6) apresentam duas realizações diferentes do mecanismo de sincronização. Uma é assistida por meios físicos e permite um processo com um menor "tempo para ficar pronto". Ele exige, evidentemente, circuitos de controlo adicionais pelo menos na entidade S/370 e pode ser melhorado para além da capacidade definida, expondo fisicamente certos circuitos de controlo S/88 à "interface" S/370. Esta "interface" é, na realidade o "acessário parasita" dos circuitos S/370 para os circuitos S/88.

A outra forma de realização aqui definida é apenas o microcódigo, que permite o tratamento da sincronização S/370 pelas entidades do processador S/88 na emulação de um Processador de Serviço S/370. Esta técnica pode ser usada quando não forem críticos a eficácia e "tempo para ficar pronto".

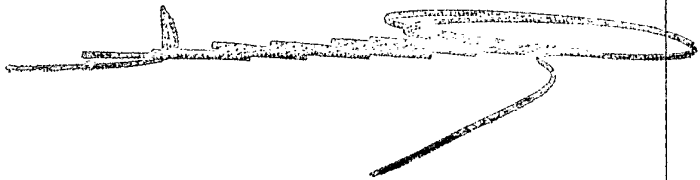
3. Numa Unidade de Processamento Simples (21) é ligada a energia

a) Realização em meios físicos

Esta situação pode ser provocada por qualquer das condições:

1. Esta unidade entra em linha em consequência de uma inicialização da ligação da energia.
2. Esta unidade entra em linha em consequência de uma Recuperação de Avaria da Alimentação de Energia.

Para as duas condições, o curso do código

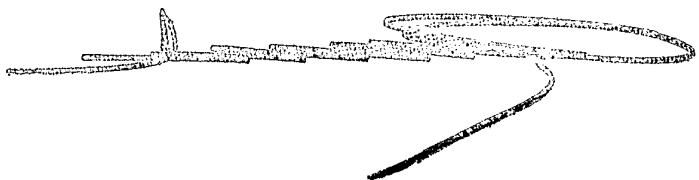


go é o mesmo:

A entidade S/88 da unidade (21) executa uma porção da sua rotina de Ensaio Automático e Inicialização (STIR) e depois procura determinar se ou não o conteúdo da sua memória (16) associada estava comprometida (Estado de Avaria De Alimentação de Energia). Se sim, regressará ao trajecto de alimentação normal na rotina STIR. Se não, tentará determinar se tem uma unidade de processamento parceira ou co-residente que possa ser a DMPU. Se encontrar uma, ela assumirá a responsabilidade de DMPU e tentará sincronizar quaisquer outras unidades de processamento.

A entidade S/370 da unidade (21) simplesmente segue a direcção da entidade S/88. Isso faz-se pelo processador S/88 (62), executando o código residente na PROM (181) do S/88, completando o Ensaio Automático normal e depois determinando se é uma Ligação da Energia Inicial ou uma "Recuperação da Falha de Energia". Se for uma Ligação da Energia, continua com a inicialização normal; depois, admitindo que é a DMPU, tenta gerar um sinal SYNC. O sinal é captado pela lógica S/370, que força uma interrupção de nível 6 para o processador S/88 (62). A interrupção de nível 6 formará os vectores para o microcódigo de sincronização S/370 na PROM (181) S/88 (fig. 19A) (que é disposta nas posições num mapa do espaço de endereços S/88).

Entretanto, a partir da Ligação da Energia/Inicialização, o PE S/370 (85) executou a sua própria rotina STIR, depois suspensa no seu Ponto de Sincronismo. Durante este tempo, o relógio S/370 (152) também se inicializou. A Subrotina de Serviço de Interrupção de nível 6 (ISS) (isto é, o microcódigo de sincronização S/370) usa a interface ISF/USI da fig. 44 para emular o Processador de Serviço S/370. Este Emulador SP produzirá Correntes de Funções para invocar a função IML da memória de comando S/370 (171), embora não ocorra qualquer transferência de código efectiva (o microcódigo está na PROM (181) S/88). A fase seguinte da Emulação IML é a difusão do sinal de sincronização SYNC para a entidade S/370 (processadores (85) e (87)), fazendo com que a unidade de processamento (21) comece a execução. A fase final da rotina ISS é o Retorno da In



terrupção, fazendo com que a unidade de processamento comece a execução da carga IPL.

Como parte da execução "podule_start-up.cm" da unidade de processamento S/88 será emitida uma corrente de Funções emulada do processador de serviço "IPL Button Pushed" para a unidade de processamento para efectuar a função IPL, carregando a memória principal S/370 a partir de um disco. A fase final da IPL é então passar o controlo para o endereço especificado na posição 0.

B. Forma de realização com apenas o Microcódigo

A entidade S/88 da unidade (21) executa o seu Ensaio Automático e a Rotina de Inicialização (STIR) e depois determinará se é uma Ligação Inicial da Energia (IPO) ou uma Recuperação de uma Falha de Energia (PFR). Se for uma IPO, o código determina que a unidade (21) é uma entidade simples e procede à carga do Sistema Operativo e executa a rotina de "arranque".

Se for uma PFR, o código determina se ou não está incluída a sua memória associada. Se foi, o código prossegue como se se tratasse de uma IPO. Se a memória for encontrada com o seu conteúdo intacto, o código PFR continua com as tarefas de re-arranque normais.

Em qualquer dos casos anteriores, a função de sincronização torna-se uma operação "simulada", pois não há parceiro associado para ser sincronizado.

4. Nas Unidades de Processamento Duplicadas (21,23) é ligada a energia - Forma de realização em meios físicos

Esta situação pode ser provocada por uma ou as duas condições seguintes:

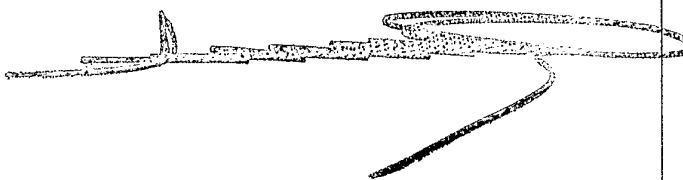
1. Estas unidades entram em linha como resultado de uma ligação da energia/inicialização
2. Estas unidades entram em linha como resultado de uma recuperação da energia avariada.



A entidade S/88 de cada unidade de processamento (21,23) executa uma parte da sua rotina de Ensaio Automático e Inicialização (STIR) e depois procura determinar se ou não o conteúdo da sua memória (16) associada foi comprometido (estado de falha de energia). Se sim, regressará ao trajecto normal na rotina STIR de ligação da energia. Se não, tentará determinar se tem uma unidade de processamento parceira ou co-residente que possa ser a DMPU ou se sim ou não é a DMPU. Se for, tomará a responsabilidade de DMPU e tenta sincronizar quaisquer outras unidades de processamento. Se não for a DMPU, prosseguirá para o Sync Point e aguardará o sinal SYNC.

Cada entidade S/370 simplesmente segue a direcção da entidade S/88. A entidade S/88, que executa o código residente na sua PROM (181), completa o Ensaio Automático normal e depois determina se é uma Ligação da Energia ou a Recuperação de uma Falha de Energia. Se for uma Ligação da Energia, continua a inicialização normal; depois prossegue para o Sync Point. Se for uma Recuperação de uma Falha de Energia, é examinada a memória "cache" para determinar se sim ou não é válida. Se for, pode ter de actualizar a sua memória parceira, se essa memória "cache" foi encontrada inválida. Se a sua própria memória "cache" estiver inválida tem de depender da sua parceira actualizar a com o conteúdo válido da memória "cache". Se nenhuma das parceiras puder assegurar a memória válida têm de, como um par, continuar a Ligação de Energia e Inicialização normal. Enquanto as entidades S/88 do par de unidades de processamento se aproxima do Sync Point, cada entidade S/88 determina se ou não tem de assumir a responsabilidade de DMPU. Se verificar que é a DMPU, ela tenta produzir um sinal SYNC.

O sinal Sync é capturado pela lógica S/370 e força uma interrupção de nível 6 para a entidade S/88. A interrupção formará os vectores para o microcódigo de sincronização S/370 na PROM (181) (que está nas posições no espaço de endereços S/88). Entretanto, a partir da Ligação da Energia/Inicialização, a entidade S/370 (por exemplo os elementos de processamento (85,87)) executou a sua própria rotina STIR e depois suspendeu a execução no seu Sync Point. Se for uma Recuperação de Falha de Energia, a entidade S/370 prossegue por um processo



análogo ao processo da entidade S/88 para determinar até onde tem de retroceder-se na rotina de inicialização para garantir a integridade da memória e a sincronização, Durante este tempo, o relógio S/370 inicializou-se a si próprio.

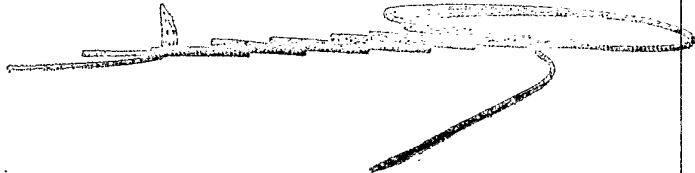
Vai agora fazer-se uma descrição resumida de um mecanismo preferido para a captação do impulso SYNC S/88 pelos processadores S/370, com referência às fig. 20, 49, 50.

Os processadores S/88 obtêm a sincronização pelo envio, por um do par de processadores S/88 da unidade (23), de um sinal SYNC OUT na linha (570) (fig. 50). Se a unidade parceira foi inicializada e efectuou o ensaio automático e determinou não estar BROKEN, ela tem um nível do sinal na linha BROKEN (571), que é invertido pelo circuito (572) para encaminhar o sinal SYNC OUT através da porta AND INVERT (AI) (573).

No sistema S/88 original (por exemplo o módulo (10)), o sinal SYNC era aplicado à linha SYNC IN (580) do excitador (d) do processador S/88 de uma unidade (14) através da linha (577) e do inversor (574). É também aplicado à linha SYNC IN (575) do processador S/88 de verificação da unidade (12), através da linha omnibus C e do inversor (576) para iniciar a "separação" de todos os quatro processadores S/88 de unidades (12,14) em arranque e paragem.

Nas unidades S/370-S/88 aperfeiçoadas, tais como (21,23), a saída (577) do circuito (573) é desligada das linhas SYNC IN (580) e (575) para impedir a separação dos processadores S/88. Em vez disso é ligada através da linha (581) para ajustar um flip-flop (FF) (582) na BCU (156) da unidade parceira (21), fig. 48. Também ajusta um flip-flop correspondente na BCU parceira (não representada) na unidade (21). A descrição refere-se apenas a um S/370 e ao equipamento físico associado na unidade (21), mas compreender-se-á que as duas entidades S/370 operam de maneira semelhante.

O flip-flop (582) aplica um sinal de interrupção de nível 6 ao processador S/88 (62) através da linha (583), dos circuitos OU (292a e 292b) (fig. 20), da lógica de interrupção (293) e das linhas IPO-2. Esta acção é designada por "captura" do sinal SYNC S/88 pelo S/370.



Admite-se que as entidades S/370 da unidade (21) executarem com êxito as suas rotinas de ensaio automático e inicialização (STIR) e estão prontas para a separação.

Como atrás de descreveu com referência à fig. 20, relativamente ao outro DMAC e interrupções BCU de nível 6, o processador S/88 (62) inicia um ciclo de confirmação da interrupção em resposta a um sinal SYNC na linha (583). O código de funções e os sinais do nível de prioridade provenientes dos processadores (62) são descodificados na lógica (281), faz-se um pedido da linha omnibus local pela linha (190) através da saída (283) da lógica de descodificação (281) e da porta (291), da linha (287) e do circuito OU (284).

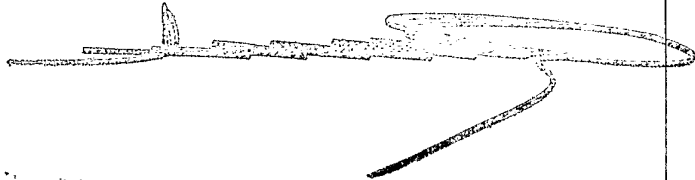
Quando for concedido um ciclo da linha omnibus ao processador (62) na linha (191) ele (juntamente com sinais na linha SYNC (583), na linha AS (270) e na linha de descodificação (283)) activa a porta E (295-4), para aplicar um sinal na linha IACK (258f). Este sinal é aplicado à lógica dos bits do vector (584) (fig. 49) para aplicar um número de vector apropriado ao processador S/88 através da linha omnibus local S/88 (233), dos emissores-receptores (218) e da linha omnibus do processador (161D). O sinal na linha (258f) também repõe o flip-flop (582).

Se a função STIR S/370 já estiver completa, como se admitiu, o processador S/88 (62) executa um ciclo de leitura para obter o número do vector, que é depois usado pelo processador (62) para aceder à primeira instrução de uma rotina de interrupção para a sincronização S/370.

A última instrução da rotina de sincronização gera um comando SYNC que aplica um sinal SYNC à linha (586) (fig. 50).

Este sinal é aplicado às linhas SYNC (580) e (575) para "separar" os processadores S/88 (bem como os S/370) das unidades parceiras (21,23) em arranque-paragem.

Como uma parte da execução de "module start up em" será enviada uma sequência de funções emulada SP "IML Button Pushed" para as entidades S/370 nas unidades (21, 23). Em vez de executar toda a operação de carga IPL completa dos acessos aos DASD, etc., esta carga IML cutocircuitará os



processos I/O e fará a carga a partir da memória principal S/88. O código EXEC370 terá já buscado o código IPL a partir dos DASD e colocado o mesmo na memória principal S/88, aguardando a IPL. A fase final da IPL é então passar o controle para o endereço especificado pela posição 0.

B. Forma de realização apenas como microcódigo

Quer para os cartões PU a que se ligou a energia em consequência de uma ligação inicial (IPO), quer em consequência de uma Recuperação de uma Falha de energia (PFR).

Considerando primeiramente o caso da IPO:

Em consequência do sinal Power Good S/88 que está a ser declarado pelo IPO, uma interrupção de manutenção invoca o código da PROM (181) S/88. Este código sincroniza a entidade S/88 da unidade (21), depois chama a rotina STIR S/370, também residente na PROM (181). A rotina STIR S/370 determina que, tratando-se de uma IPO, não foram carregadas facilidades suficientes para permitir que ela se inicialize e se sincronize, requerendo as facilidades do S/88 e do seu Sistema Operativo. Como consequência disso, a rotina STIR S/370 regressa, sem qualquer outra acção, ao código da PROM (181) S/88 que procede à carga do Sistema Operativo. Como parte da inicialização do Sistema Operativo, é chamado um módulo "Start up". Este módulo também chama a rotina STIR S/370 residente na PROM (181). Nesta altura, a rotina STIR determina que estão disponíveis as facilidades e utiliza as mesmas para sincronizar então a própria "Initial Microcode Load" (IML).

Em segundo lugar para o caso de uma PFR:

Como consequência do sinal S/88 Power Good declarado pelo IPO, uma Interrupção de Manutenção invoca o código S/88 da PROM (181). Este código sincroniza a entidade S/88 da unidade (21), depois chama a rotina STIR S/370, também residente na PROM (181). A rotina STIR S/370 determina que, sendo uma PFR, estão disponíveis as facilidades necessárias e procede à sincronização e à inicialização da unidade S/370 ou uni-

dade (21).

5. É introduzido um parceiro (23) enquanto a outra unidade (21) funciona normalmente

A. Forma de realização com meios físicos

Será enviada uma interrupção de nível 6 para a entidade S/88 da unidade corrente (21) após a inserção de um novo cartão. Enquanto a nova unidade de processamento está a executar a sua rotina STIR, a unidade de processamento corrente reconhecerá a interrupção de nível 6. O nível 6 ocupa-se do processo de arquivar o ambiente de tarefas pré-esvaziadas, determinando se a nova unidade de processamento está em linha; e, quando ela estiver, repõe a interrupção. Como uma função de reposição da interrupção, as duas unidades procederão à sincronização por arranque e paragem, retomando a tarefa pré-esvaziada.

B. Forma de realização só com o microcódigo

Como consequência de se introduzir o novo cartão, uma Interrupção de Manutenção invoca o código S/88 na PROM (181). Este código re-sincroniza a entidade S/88 da unidade (21), chama depois a rotina STIR S/370, também residente na PROM (181). A rotina STIR S/370 determina que, sendo isto semelhante a uma PFR, estão disponíveis as facilidades necessárias e procede à sincronização e à inicialização da entidade S/370 da unidade (21).

6. Um parceiro detecta uma falha de comparação

A. Forma de realização por meios físicos

A unidade de processamento avariada será forçada para a sua rotina STIR, enquanto a unidade de processamento que funciona normalmente será interrompida por uma interrupção de nível 6. A sub-rotina de Serviço de Interrupção de nível 6 executará o processo de arquivar o ambiente de tarefas



pré-esvaziadas, determinando se a nova unidade de processamento está em linha e, se estiver, retornará da interrupção. Como uma função do retorno da interrupção, as duas unidades prosseguirão na sincronização por arranque e paragem, retomando a tarefa pré-esvaziada. Se a unidade de processamento avariada falhar na saída da sua rotina STIR correctamente (por exemplo uma vez ou num número escolhido de tentativas), a unidade de processamento que funciona normalmente ajustará, após um tempo apropriado, um sinal BROKEN para a parte S/88 da unidade de processamento avariada e as suas várias facilidades referentes aos estados.

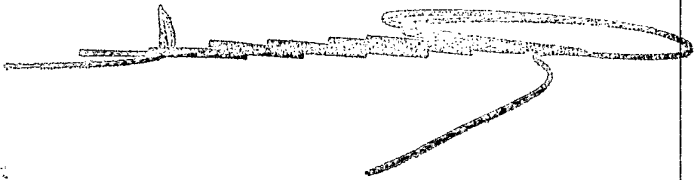
B. Forma de realização com apenas o microcódigo

Como consequência da detecção de uma falha na comparação, e estando o cartão fora de linha, uma Interrupção de Manutenção invoca o código S/88 na PROM (181), Este código re-sincroniza a entidade S/88 da unidade (21), depois chama a rotina STIR S/370, também residente na PROM (181). A rotina STIR S/370 determina que, sendo semelhante a uma PFR, estão disponíveis as facilidades necessárias e procede à sincronização e à inicialização da entidade S/370 da unidade (21). Resultará uma nova falha de comparação sendo repetida a mesma acção. Depois de um número pré-determinado de iterações, o cartão será posto fora de linha definitivamente e participada uma avaria.

Formas de realização alternativas

1. Utilização noutros (não S/88) sistemas tolerantes das avarias

Na forma de realização preferida, os meios físicos tolerantes das avarias têm, como se tem verificado, pelo menos três características. Há um isolamento eléctrico instantâneo de uma unidade substituível no campo, sem a propagação de erros de dados para outro elemento do sistema. É proporcionado um código de reconfiguração dinâmica para retirar ou adicionar componentes, como se tornar necessário ou quando os componentes se avariarem. Proporciona-se a capacidade para des-



ligar a energia e ligar a energia a um subsistema ou unidade substituível no campo, sem a perda do sistema, isto é a capacidade de enfiar unidades com a alimentação ligada. O utilizador não nota qualquer perda de funcionamento ou de eficácia.

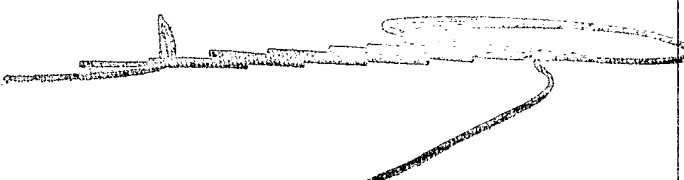
Compreender-se-á que os presentes aperfeiçoamentos podem ser usados em diferentes ambientes tolerantes das avarias, tais como sistemas tolerantes de avarias nos meios de programação sem certos dos requisitos estritos anteriores.

Um exemplo de um outro sistema (sem certos dos requisitos estritos anteriores) com o qual pode usar-se o presente aperfeiçoamento está descrito na patente US 4 356 550, intitulada "Microprocessar System", publicado em 26 de Outubro de 1982, em nome de James A. Katman, et al. Na fig. 1 desta patente três subsistemas de processamento operam assincronamente uns com os outros e estão acoplados a linhas omnibus duplicadas. Se um sistema se avariar, os restantes dois continuam a execução do programa. Todos os erros são determinados em pontos de verificação no programa, em vez de instantaneamente, como na forma de realização preferida segundo a presente invenção.

Os processadores, tais como os processadores S/370, estranhos aos subsistemas da patente, podem ser ligados aos referidos subsistemas de uma maneira análoga à representada na presente invenção relativamente ao S/88. Utilizando e controlando linhas seleccionadas nos subsistemas da patente de uma maneira semelhante à descrita com referência à linha de impulsos de sincronização de endereços (AS) da presente invenção, os processadores dos subsistemas podem ser desacoplados para permitir a sua utilização como controladores de I/O para os processadores estranhos parasitas ligados.

2. Transferências de dados directa entre controladores I/O S/88

Na forma de realização preferida, admite-se que a memória "cache" (340) pode ser a memória exclusiva para alguns dados I/O válidos (em vez da memória (162), que armazena todos os dados válidos de dados I/O), como é o caso actualmente nos sistemas de memória "cache" S/370. Na forma de



realização da fig. 51, na qual se admite que a memória (162) armazena todos os dados I/O válidos, as transferências de dados I/O podem ter lugar directamente entre um dispositivo I/O S/88, tal como um controlador de discos (20) e a memória S/370 (162) para uma operação mais eficiente.

Contudo, desta forma de realização alternativa, a BCU (156) pode ainda ser usada para transferir comandos I/O S/370 para o S/88. Os endereços da memória S/370 associadas com os comandos têm de ser alterados para endereços físicos S/88 pelo código EXEC370, enquanto que os comandos são convertidos para comandos S/88.

Durante as transferências de dados da memória (162) para dispositivos I/O, um processo é primeiro limpar a secção da memória "cache", relacionada com a operação I/O, para a memória (162) antes de efectuar a operação I/O.

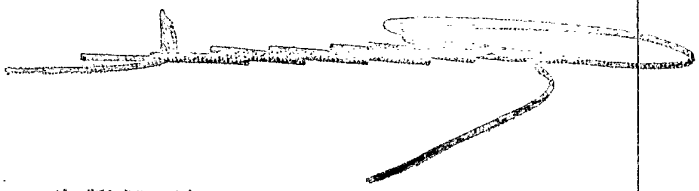
Durante as transferências de dados dos dispositivos I/O para a memória (162), a secção da memória "cache" relacionada com a operação I/O é invalidada antes de efectuar a operação I/O.

Se for necessária uma conversão de dados, a função pode ser executada no ou nos controladores dos dispositivos I/O, por rotinas semelhantes às usadas pelo EXEC 370 no interior do processador S/88 (62).

A conversão de dados pode também ser efectuada pela chamada pela aplicação EXEC370 de rotinas de conversão no sistema operativo S/88, tais como a conversão de ASCII para EBCDIC.

3. Desacoplamento dos dois processadores de um par ligado directamente

A fig. 52 ilustra o fluxo de dados para uma forma de realização alternativa, na qual ambos os processadores de um par de processadores acoplados directamente são desacoplados dos seus meios físicos associados, de preferência de uma maneira genericamente análoga à descrita com referência ao processador S/88 (62) da forma de realização preferida para



transferir comandos e/ou dados entre os processadores de uma maneira transparente aos seus sistemas operativos.

Dois processadores (640) e (641) estão acoplados entre si através de linhas omnibus (642) e (643) dos processadores, circuitos excitadores/receptores (DRV/RCVR) (644) e (645) e uma unidade de memória local comum (646). Os processadores (640) e (641) podem ter a mesma arquitetura ou arquiteturas diferentes e o mesmo sistema operativo ou sistemas operativos diferentes. Cada processador (640) e (641) pode ter os seus próprios meios físicos (não representados), incluindo a memória principal e dispositivos I/O para o processamento normal de programas sob o controlo dos sistemas operativos respectivos. Nenhum dos sistemas operativos tem conhecimento da existência do processador associado com o outro sistema operativo nem do acoplamento com esse processador.

Mas quando o processador (640) desta forma de realização alternativa for controlado por um programa de aplicação para enviar comandos e/ou dados para o processador (641), de preferência ele coloca um endereço pré-determinado na linha omnibus de endereços do processador (647) que é descodificado pela lógica (648) para fazer com que os circuitos (644) acoplem a linha omnibus (642) à memória local (646) através da linha omnibus local (652) para a transferência de comandos e dados do processador (640) para a memória (646). A descodificação do endereço também desacopla o processador (640) dos seus meios físicos associados para tornar a transferência transparente para o sistema operativo do processador (640).

O desacoplamento da lógica (649) interrompe o processador (641) quando os comandos e/ou os dados I/O destinados ao processador (641) tiverem sido transferidos para a memória local (646). O processador (641) (através do seu dispositivo de tratamento de interrupções do programa de aplicação) é desacoplado dos seus meios físicos e lê os comandos e/ou os dados da memória (646) para a sua memória principal (não representada), de uma maneira transparente ao seu sistema operativo. Se os comandos e/ou os dados exigirem conversão, o processador (641) utiliza o microcódigo de emulação na memória (650) para executar a conversão requerida. O processador (641) processa en



tão os comandos convertidos sob o controlo do seu sistema operativo.

Compreender-se-á que o "desacoplamento" dos processadores (640) e (641) pode permitir a transferência contínua de um segmento substancial de comandos e/ou de dados para e da memória local (646) antes do reacoplamento" de cada processador aos seus meios físicos ser permitido. Deste modo, conseguir-se-ão transferências de dados rápidas e eficientes.

Os comandos e/ou os dados podem ser transferidos no sentido oposto, do processador (641) para o processador (640), de uma maneira análoga. Os comandos e/ou os dados podem ser convertidos quando necessário pelo microcódigo de emulação situado na memória (651) e os comandos convertidos podem ser processados no processador (640) sob o controlo do seu sistema operativo.

Esta forma de realização alternativa difere num aspecto significativo da forma de realização preferida, isto é, o processador "indicador" da transferência de dados é desacoplado dos seus meios físicos para emitir dados para o processador "receptor". Isso exige a função adicional do controlo da transferência para um programa de aplicação semelhante ao EXEC370/ETIO da forma de realização preferida quando se tiver que efectuar uma função I/O (transferir comandos e/ou dados para outro processador).

Os meios para efectuar a transferência de controlo para certas operações I/O de um sistema operativo para um programa de aplicação dependerão das características do sistema.

Por exemplo, na forma de realização preferida, o S/370 executa uma instrução I/O de Arranque, que é processada pelo sistema operativo de uma maneira normal sem "desacoplamento" do processador S/370 dos seus meios físicos associados.

Na forma de realização alternativa da fig. 52, para os casos em que o processador S/370 (640) emite comandos e/ou dados para o processador (641), pode usar-se um OP CODE inválido seleccionado em vez da instrução Start I/O. A descodificação do OP Code inválido seleccionado transfere o con



trolo para um programa de aplicação especial que "desacopla" o S/370 dos seus meios físicos para a transferência de informação com o processador (641), através da memória (646).

Para impedir o excesso de escrita por um processador de dados transferidos pelo outro processador para a memória (646), o processador (640) pode ser controlado para escrever em apenas uma secção específica da memória (646) e o processador (641) é controlado para apenas ler a partir da referida secção. O processador (641) tem permissão para escrever apenas numa segunda secção da memória (646) e o processador (640) pode ler apenas a partir da referida segunda secção. Os processadores (640) e (641) são inibidos de escrever para, respectivamente, as referidas primeira e segunda secções.

Os mecanismos de desacoplamento e interrupção são operados de maneira transparente para os sistemas operativos dos dois processadores (640) e (641), como se descreveu com referência ao processador S/88 (62) da forma de realização preferida.

As funções de emulação podem ser executadas por programas de aplicação (em vez de por microcódigo na memória local), da mesma maneira que se descreveu relativamente ao EXEC370, na forma de realização preferida.

Podem usar-se técnicas de inquirição em vez do mecanismo de interrupção para transferir dados entre os processadores (640,641); porém, tais técnicas seriam ineficientes.

Compreender-se-á que, como cada um dos processadores (640) e (641) pode realizar operações I/O para o outro processador, qualquer dos processadores pode adquirir certas características do ambiente I/O do outro.

Também se compreenderá que uma aplicação num processador pode comunicar para uma aplicação igual ou diferente num segundo processador sem usar os serviços do sistema operativo de qualquer dos sistemas processadores.

Em algumas das reivindicações é usada a expressão "programa ou código de aplicação", no seu sentido convencional, tal como é interpretado pelos entendidos na técnica do processamento de dados, isto é, tipicamente distinto do código



go do sistema operativo, da seguinte maneira:

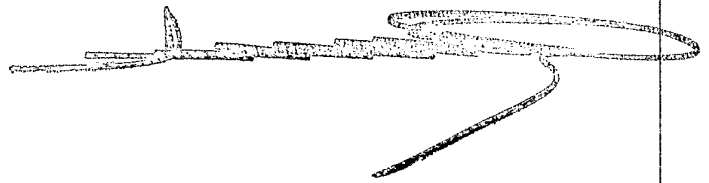
1. Os programas de aplicação estão no topo de um sistema operativo e tipicamente têm de chamar o sistema operativo para serviços tais como Leitura, Escrita e Controlo de I/O, hora do dia, etc..
2. O código de aplicação arranca ou é iniciado por um utilizador e é carregado através de serviços do sistema operativo.
3. O sistema operativo controla a chamada dos programas de aplicação para e da memória.
4. O sistema operativo atribui memória principal aos programas de aplicação. Porém, um tal código de "aplicação" tem agora que executar funções adicionais.

"Estranho" é usado em algumas das reivindicações para definir aparelhos que não são do conhecimento de um sistema operativo porque não são definidos nas tabelas de configuração do sistema operativo e portanto o sistema operativo não tem qualquer dispositivo actuador para o aparelho, não podendo controlar o mesmo. Porém, um programa especial de aplicação que é executado no sistema operativo tem conhecimento do aparelho e pode exercer controlo sobre o aparelho.

Nas reivindicações, "discernir" é usado no sentido de que um sistema operativo não tem conhecimento de aparelhos estranhos ligados a um processador no qual o sistema operativo é executado, ou que são tomadas acções pelo processador que são isoladas do sistema operativo para impedir que o sistema operativo rejeite tais acções.

Na memória descritiva, o termo "transparente" foi usado com frequência com este mesmo sentido.

Embora a presente invenção tenha sido apresentada e descrita particularmente com referência a uma forma de realização específica. Os entendidos na matéria compreenderão que as alterações e alternativas atrás sugeridas e várias outras alterações de forma e de pormenor podem ser introduzidas sem nos afastarmos dos ensinamentos da presente invenção. Pretende-se por isso que a descrição anterior e os desenhos anexos sejam interpretados como ilustrativos e não limitativos, pretendendo-se também que as reivindicações anexas cubram todas as alterações e modificações que estejam dentro dos verdadeiros espí



rito e objectivos da presente invenção.

•
•
•

ÍNDICE

Fundamento da invenção	3
INtrodução	3
Sumário da invenção	3
Breve descrição dos desenhos	4
Descrição da forma de realização preferida	9
Introdução	9
1. Operação de um processador normalmente não tole- rante para as avarias num ambiente tolerante pa- ra as avarias	16
2. Desacoplamento de um processador do seu equipa- mento físico associado para apresentar comandos e dados provenientes de um outro processador a si próprio	19
3. Apresentação de interrupções a um sistema trans- parente para o sistema operativo	23
4. Partilha de uma memória real entre dois ou mais processadores que executam sistemas operativos de memória virtual diferente	25
5. Imagem de um sistema único	30
6. Sumário	36
Introdução do sistema/88 da técnica anterior	37
Módulo S/370 tolerante de avarias (9) interliga do através de HSDI; redes	46
Descrição geral das unidades emparelhadas (21, 23) do processador	47
Acoplamento de elementos de processamento S/370 e S/88 (85,62) (fig. 11 e 12)	52
Interface (89) entre processador e processador	59
1. Adaptador I/O (154)	59
2. Linha omnibus dos canais 0 e 1 do adaptador I/O	68
3. Unidade de controlo de linhas omnibus (156) - Descrição geral	74
4. Controlador do acesso directo à memó- ria (209)	75

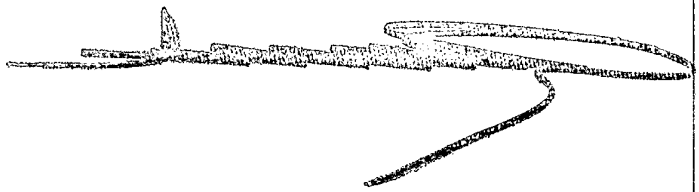
5. Unidade de controlo de linhas omnibus	
- Descrição pormenorizada	83
a) Registadores de interface para trans- ferência de dados de alta velocidade .	83
b) Lógica de desacoplamento da BCU e de interrupção	85
c) Atribuição de posições de endereços na memória na BCU	95
d) Operação das linhas omnibus de ende- reços e dados locais	99
e) Endereçamento do processador S/88 e do controlador DMAC para/da memória (210)	101
f) Operação do contador de octetos BCU (BSM) RD/WR	103
g) Sequências de transmissão com passa- gem de testemunho BCU (156)/Adapta- dor (154)	107
Elemento processador S/370 (85)	109
Linha omnibus do processador e comandos da linha omnibus do processador	118
Unidade de gestão da memória S/370 (81)	124
1. Controlador da memória "cache" (153)	124
2. STCI (155)	126
a) Introdução	126
b) Fases da linha omnibus do sistema	130
c) Característica da interface STCI	134
d) Operações de memorização de dados	142
e) Operações de busca de dados	148
Suporte de I/O S/370	153
Operações de I/O S/370. Vista geral dos meios de microprogramas	156
Concepção do microcódigo do sistema	166
1. Introdução	166
2. Interface de programa ETIO/EXEC370	167
3. Protocolo do microcódigo S/370, EXEC370	174
4. Fluxos de instruções entre o microcódigo S/370 e EXEC370	177



Funcionamento da unidade de controlo da linha

omnibus BCU (156)	186
1. Introdução	186
2. Fluxo da sequência Start I/O S/370. Descrição geral e pormenorizada	189
3. Fluxo da sequência de transferência de dados I/O S/370. Descrição geral	193
a) Operações de escrita I/O	194
b) Operação de leitura I/O	199
c) Fluxo de sequência de transferência de mensagens de prioridade elevada S/370 ..	204
d) Comando BCU STATUS	208
e) Reposição programada da BCU	209
Emulação do formato da pista de contagem, chave e dados	210
1. O sistema objecto	210
2. O sistema objectivo	212
3. O formato de emulação	213
4. Funções de emulação	214
Partilha da memória real (16) pelo S/88 e o S/370 ...	215
1. Introdução	215
2. Distribuição de posições da memória S/88 (16)	217
3. Processo de arranque	223
4. Rotina de serviço S/370	224
5. Retirar a sequência de mme escolhida da lista livre	230
6. Escrita da base e da dimensão da memória para a STCI	230
Funções de inicialização para as interrupções S/88 iniciadas pelo S/370	231
Ganho da liberdade sem modificar o sistema operativo S/88	234
Roubo de memória sem modificar o sistema operativo S/88	235
Ligação da energia e sincronização de unidades simples e de unidades emparelhadas (21,23)	235

Unidade de processamento S/88 como processador	
de serviço para a unidade de processamento	
S/370	235
1. Introdução	235
2. Sincronização dos meios físicos tolerantes das avarias	241
3. Numa unidade de processamento simples é ligada a energia	249
a. Realização em meios físicos	249
b. Forma de realização com apenas o microcódigo	251
4. Nas unidades de processamento duplicadas (21,23) é ligada a energia	251
a. Forma de realização com meios físicos .	251
b. Forma de realização com o microcódigo -	255
5. É introduzido um parceiro (23) enquanto a outra unidade (21) funciona normalmente.	256
a. Forma de realização com meios físicos .	256
b. Forma de realização só com o microcódigo	256
6. Um parceiro detecta uma falha de comparação	256
a. Forma de realização por meios físicos .	256
b. Forma de realização com apenas o microcódigo	257
Formas de realização alternativas	257
1. Utilização noutros (que não o S/88) sistemas tolerantes das avarias	257
2. Transferências de dados directa entre controladores I/O S/88	258
3. Desacoplamento dos dois processadores de um par ligado directamente	259



REIVINDICAÇÕES

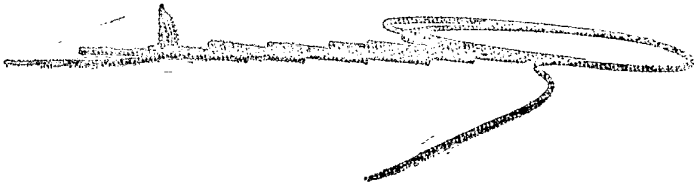
- 1ª -

Combinação, num sistema de informação, caracterizada por compreender um primeiro sistema de processamento de dados que possui uma unidade de memória principal e que opera segundo um primeiro sistema operativo virtual de acordo com uma primeira estrutura de instruções, um segundo sistema de processamento de dados que opera num segundo sistema operativo de acordo com uma segunda arquitectura de instruções e acoplado à referida unidade de memória principal, meios no referido primeiro sistema de processamento de dados para atribuir uma parte da referida unidade de memória principal, indescernível para ambos os sistemas operativos, para uso exclusivo do segundo sistema de processamento de dados e para o seu sistema operativo; e meios nos primeiro e segundo sistemas de processamento de dados que incluem um meio de registador que define a referida parte atribuída da memória principal, para aceder à referida porção atribuída em resposta à execução de instruções em ambos os sistemas de processamento de dados.

- 2ª -

Combinação de acordo com a reivindicação 1, caracterizada por compreender meios que inibem o primeiro sistema operativo de atribuir ulteriormente a referida parte atribuída da memória principal.

- 3ª -



- 3a -

Combinação de acordo com a reivindicação 1, caracterizada por os referidos meios de atribuição compreenderem ainda

um programa de aplicação do referido primeiro sistema, operado selectivamente no modo de supervisão, para atribuir a referida parte da memória principal ao referido segundo sistema, indescernível para ambos os sistemas operativos.

- 4a -

Combinação de acordo com a reivindicação 2, caracterizada por os referidos meios de acesso incluírem um programa de aplicação no referido primeiro sistema de processamento para aceder à referida parte atribuída da memória principal de acordo com a execução das instruções I/O do segundo sistema de processamento.

- 5a -

Combinação de acordo com a reivindicação 2, caracterizado por os referidos meios de acesso incluírem meios no primeiro sistema de processamento, excluindo o primeiro sistema operativo e incluindo meios que armazenam dados de endereço que definem a parte atribuída da memória principal, acedendo à referida parte atribuída para transferir dados I/O do segundo sistema de processamento para e da referida parte atribuída.

- 6a -



- 6a -

Combinação, num sistema de processamento de informação, caracterizada por compreender um primeiro sistema de processamento de dados com uma unidade de memória principal e operando num primeiro sistema operativo virtual de acordo com uma primeira arquitectura de instruções, um segundo sistema de processamento de dados operando num segundo sistema operativo virtual de acordo com uma segunda arquitectura e acoplamento à referida unidade de memória principal, meios no referido sistema de processamento de dados para atribuir uma parte da referida unidade de memória principal, indiscerníveis para ambos os sistemas operativos, para uso exclusivo do segundo sistema de processamento de dados e do seu sistema operativo; e meios no segundo sistema de processamento de dados, incluindo um meio de registador que define a referida parte atribuída da memória principal, para aceder à referida parte atribuída em resposta à execução de instruções na segunda unidade de sistema operativo.

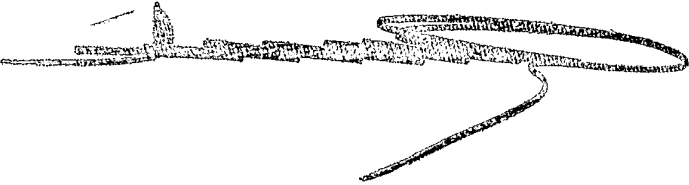
- 7a -

Combinação de acordo com a reivindicação 6, caracterizada por compreender ainda meios que inibem o primeiro sistema operativo de atribuir anteriormente a referida parte atribuída da memória principal.

- 8a -

Combinação de acordo com a reivindicação 6, caracterizada por os referidos meios de atribuição compreenderem ainda

- um programa de aplicação no referido primeiro sistema, operando



selectivamente no modo de supervisão para atribuir a referida parte de memória principal ao referido segundo sistema, indiscernível para ambos os sistemas operativos.

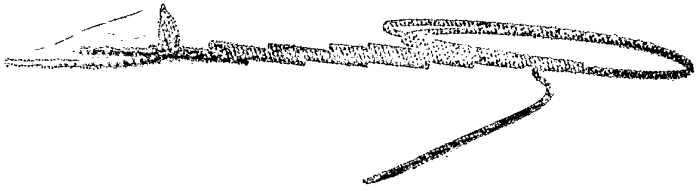
- 9a -

Combinação de acordo com a reivindicação 6, caracterizada por compreender ainda meios que incluem um programa de aplicação no referido primeiro sistema de processamento para aceder à referida parte atribuída de memória principal de acordo com a execução de instruções I/O do segundo sistema de processamento.

- 10a -

Mecanismo de acesso partilhado a uma memória, caracterizado por compreender um primeiro sistema de processamento de dados que possui uma primeira unidade de processamento, uma unidade de memória principal a vários dispositivos I/O que operam num primeiro sistema operativo virtual de acordo com uma primeira arquitectura, incluindo um gestor de memória para a referida memória principal, um segundo sistema de processamento de dados que inclui uma segunda unidade de processamento acoplada à referida unidade de memória e operando num segundo sistema operativo virtual de acordo com uma segunda arquitectura, meios efectivos durante a inicialização do primeiro sistema para alterar o gestor da memória para a captura do primeiro sistema operativo de uma parte da referida unidade de memória principal, indiscernível a ambos os sistemas operativos, para uso exclusivo da segunda unidade de processamento e para o seu sistema operativo,

- meios de registador,
- meios efectivos durante a referida inicialização para transfe-



rir para os referidos meios de registador, indiscerníveis para os dois sistemas operativos, informação de endereços que definem a referida parte capturada da unidade de memória principal, e meios no segundo sistema que incluem os referidos meios de registador para aceder à referida parte capturada durante a execução das instruções pela segunda unidade de processamento.

- 11a -

Mecanismo de acordo com a reivindicação 10, caracterizado por os referidos meios de captura compreenderem ainda um programa de aplicação no referido primeiro sistema operado selectivamente no modo de supervisão para a referida captura da referida parte de memória a partir do referido gestor da memória.

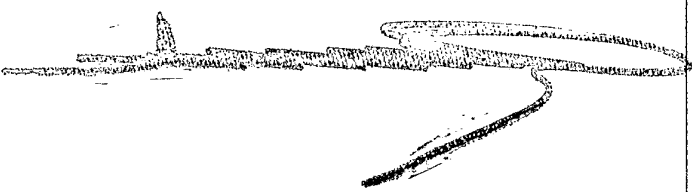
- 12a -

Mecanismo de acordo com a reivindicação 10, caracterizado por compreender meios que inibem o primeiro sistema operativo de aceder à referida parte atribuída de memória principal, e meios que incluem um programa de aplicação no referido primeiro sistema de processamento de dados para aceder à referida parte atribuída da memória principal para a transferência de dados I/O entre os primeiro e segundo sistemas de processamento de dados.

- 13a -

Mecanismo de acesso partilhado a uma me

- 273 -



mória, caracterizado por compreender um primeiro sistema de processamento de dados com uma primeira unidade de processamento, uma unidade de memória operativa e vários dispositivos I/O que operam num primeiro sistema operativo virtual de acordo com uma primeira arquitectura,

um segundo sistema de processamento de dados que inclui uma segunda unidade de processamento acoplada à referida unidade de memória e adaptada para operar num segundo sistema operativo virtual de acordo com uma segunda arquitectura,

meios que incluem um programa de aplicação executado em modo de supervisão na referida primeira unidade de processamento enquanto o referido segundo sistema é mantido numa condição de reposição, para capturar a partir do primeiro sistema operativo uma parte da referida unidade de memória principal, indiscernível para ambos os sistemas, para uso exclusivo da segunda unidade de processamento e do seu sistema operativo,

meios de registador,

meios controlados por um programa de aplicação executado na primeira unidade de processamento para transferir para os referidos meios de registador informação de endereços que define a referida parte capturada da unidade de memória principal de uma maneira indiscernível para o primeiro sistema operativo, e

meios no segundo sistema, incluindo os referidos meios de registador para aceder à referida porção capturada durante a execução de instruções pela referida segunda unidade de processamento.

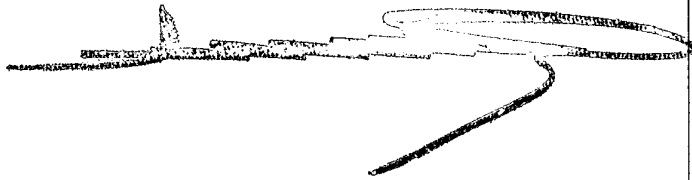
- 14a -

Mecanismo de acordo com a reivindicação 13, caracterizado por compreender ainda

meios que inibem o primeiro sistema operativo de aceder à referida parte atribuída de memória principal e

meios que incluem um programa de aplicação no referido primeiro sistema de processamento para aceder à referida parte atribuída da memória principal para a transferência de dados I/O entre os

- 274 -



primeiro e segundo sistemas de processamento de dados.

- 15ª -

Combinação, com um sistema de processamento de dados que possui um primeiro sistema que inclui uma primeira unidade de processamento e uma unidade de memória principal operando num primeiro sistema operativo de acordo com uma primeira arquitectura de instruções, caracterizada por compreender:

um segundo sistema que inclui uma segunda unidade de processamento que opera num segundo sistema operativo de acordo com uma segunda estrutura de instruções,

meios que fazem o acoplamento da segunda unidade de processamento à referida memória principal,

meios que incluem um programa de aplicação executado no modo de supervisão na referida primeira unidade de processamento durante a inicialização do sistema para capturar, sem discernimento pelo primeiro sistema operativo, uma porção da referida unidade de memória principal para o uso exclusivo da segunda unidade de processamento e do seu sistema operativo,

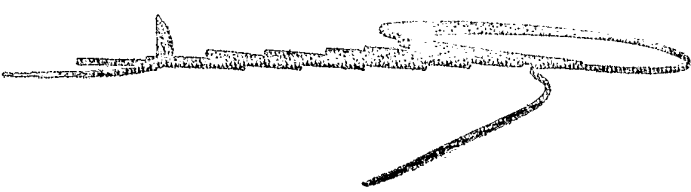
meios de registador,

meios controlados por um programa de aplicação executado na referida primeira unidade de processamento para transferir para os referidos meios de registador informação de endereços que define a referida porção capturada da unidade de memória principal sem discernimento por qualquer dos sistemas operativos, e meios no segundo sistema que incluem os referidos meios de registador, para aceder à referida parte capturada através dos referidos meios de acoplamento durante a execução das instruções pela referida segunda unidade de processamento.

- 16ª -

Mecanismo de acordo com a reivindicação 10, caracterizado por compreender ainda: meios que inibem o primeiro sistema operativo de aceder à referida parte atribuída da memória principal, e meios que incluem um programa de aplicação no referido sistema de processamento de dados para aceder à referida parte atribuída de memória principal para transferir dados I/O entre o primeiro e o segundo sistema de processamento de dados.

Mecanismo compartilhado de acesso à memória, caracterizado por compreender: um primeiro sistema de processamento de dados que inclui uma primeira unidade de processamento, uma unidade de memória principal e vários dispositivos I/O que operam num sistema operativo virtual de acordo com uma primeira arquitectura, um segundo sistema de processamento de dados que inclui uma segunda unidade de processamento acoplada à referida unidade de memória e operando num segundo sistema operativo de acordo com uma segunda arquitectura, meios efectivos durante a inicialização do primeiro sistema para capturar a partir do primeiro sistema operativo uma parte da referida unidade de memória principal, indescernível para ambos os sistemas operativos, para uso exclusivo da segunda unidade de processamento e do seu sistema operativo, meios de registador, meios efectivos durante a referida inicialização para transferir para os referidos meios de registador, indescerníveis para ambos os sistemas operativos, informação de endereços que definem a referida parte capturada da unidade de memória principal, e meios no segundo sistema de processamento, incluindo os referi-



dos meios de registador, para aceder à referida parte capturada durante a execução de instruções pela referida segunda unidade de processamento.

- 18a -

Mecanismo de acordo com a reivindicação 17, caracterizado por os referidos meios de captura compreenderem ainda:

meios que inibem o primeiro sistema operativo de aceder à referida parte atribuída de memória principal.

- 19a -

Mecanismo de acordo com a reivindicação 18, caracterizado por os referidos meios de captura compreenderem ainda:

um programa de aplicação no referido primeiro sistema operados selectivamente no modo de supervisão para capturar a referida parte da memória principal para o referido segundo sistema indiscernível para ambos os sistemas operativos.

- 20a -

Mecanismo de acordo com a reivindicação 19, caracterizado ainda por compreender:

meios incluindo um programa de aplicação no primeiro sistema de processamento para aceder à referida parte da unidade de memória principal para a transferência de dados I/O entre os primeiro e segundo sistemas de processamento.

Processo de captura de uma secção da memória principal a partir de um primeiro sistema de processamento de dados, incluindo um primeiro elemento de processamento, sendo a referida memória principal e os aparelhos de I/O operados num primeiro sistema operativo, para uso por um segundo elemento de processamento com meios de acoplamento do segundo elemento de processamento à referida memória principal e operando sob o controlo de um segundo sistema operativo, de uma maneira indescernível para ambos os sistemas operativos, caracterizado por compreender as fases de:

dividir a memória num certo número de blocos de dimensões fixas de memória contígua;

criar no primeiro sistema um mapa de memória com entradas correspondentes a cada bloco,

ligar entre si, numa lista livre no primeiro sistema, as entradas correspondentes aos blocos não presentemente atribuídos para ser usados num processo, pela introdução em cada entrada de um indicador para a entrada seguinte na lista;

ordenar a lista livre para ligar entre si, na ordem sequencial de endereços, as entradas para uma grande área de memória contígua;

colocar um programa de aplicação, adaptado para ser executado no primeiro sistema de processamento de dados, no modo de supervisão;

alterar os indicadores da lista livre sob o controlo do referido programa de aplicação executado no modo de supervisão para retirar da lista livre um grupo de entradas que definem uma área contígua de memória com uma dimensão pré-determinada;

colocar dados de endereço correspondentes à área definida de memória em registadores associados com os referidos meios de acoplamento para permitir apenas o acesso da área definida de memória pelo referido segundo elemento de processamento durante a execução do programa sob o controlo do segundo sistema operativo.

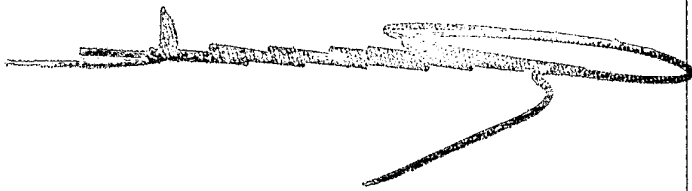
Método para a captura de uma secção da memória principal a partir de um primeiro sistema de processamento de dados, incluindo um primeiro elemento de processamento, sendo a referida memória principal e os aparelhos de I/O operados num primeiro sistema operativo, para uso por um segundo elemento de processamento que tem meios de acoplamento do segundo elemento de processamento à referida memória principal e operando sob o controlo de um segundo sistema operativo, de uma maneira indiscernível para ambos os sistemas operativos, caracteriza-se por compreender as fases de:

criar uma lista de entradas, correspondentes aos blocos não usados de memória, usados pelo primeiro sistema operativo para atribuir blocos de memória a processos,

alterar a referida lista sob o controlo de um programa de aplicação executado no modo de supervisão no primeiro elemento de processamento para retirar da lista um grupo de entradas correspondente a uma área contígua de memória de dimensão pré-determinada, tornando assim a referida área contígua não disponível para o sistema operativo,

armazenar o endereço correspondente à referida área contígua em registadores associados com os referidos meios de acoplamento para permitir o acesso da área contígua pelo referido segundo elemento de processamento durante a execução do programa sob o controlo do segundo sistema operativo.

Processo de acordo com a reivindicação 22, caracterizado por ambos os sistemas operativos serem sistemas operativos virtuais.



- 24a -

Processo de acordo com a reivindicação 23, caracterizado por as entradas da lista de entradas serem ligadas entre si inserindo em cada entrada um indicador para a entrada adjacente seguinte na lista.

- 25a -

Processo de acordo com a reivindicação 24, caracterizado por as entradas estarem dispostas por ordem para definir por ordem de endereços sequencial blocos que compreendem áreas contíguas de memória.

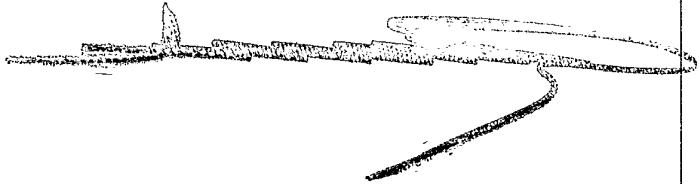
- 26a -

Processo de acordo com a reivindicação 25, caracterizado por se alterarem selectivamente certos indicadores na lista para retirar da lista o referido grupo de entradas correspondente à referida área contígua de memória com dimensão pré-determinada.

A requerente reivindica a prioridade do pedido norte-americano apresentado em 17 de Maio de 1989, sob o número de série 07/353,113.

Lisboa, 16 de Maio de 1990.
O AGENTE GERAL DA PROPRIEDADE INDUSTRIAL:

A handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke.



RESUMO

"MEMÓRIA PRINCIPAL FÍSICA ÚNICA COMPARTILHADA POR DOIS OU MAIS PROCESSADORES QUE EXECUTAM SISTEMAS OPERATIVOS RESPECTIVOS"

A invenção refere-se a uma memória principal física única compartilhada por dois ou mais processadores que executam sistemas operativos respectivos.

Fundem-se num sistema físico as funções de dois sistemas operativos virtuais (por exemplo S/370, VM, VSE ou IX370 e S/88 OS). Pares associados de processadores S/88 executam o S/88 OS e tratam os aspectos de tolerância de erros e de imagem único. Um ou mais pares associados de processadores S/370 são acoplados a processadores S/88 correspondentes directamente e através da linha omnibus S/88. A cada processador S/370 são atribuídos de 1 a 16 megaoctetos de memória contíguos da memória principal S/88. Cada sistema operativo virtual S/370 considera que a sua atribuição da memória começa no endereço 0 e gere a sua memória por meio de técnicas normais de atribuição e paginação de memórias dinâmicas S/370. O S/370 é controlado nos seus limites para impedir que o director de todos os dispositivos dos equipamentos físicos ("hardware") e dispositivos de entrada/saída. Os processadores acedem ao espaço de endereços S/370 em resposta directa a um programa de aplicação S/88 de modo que o S/88 pode deslocar dados da entrada/saída (I/O) para as memórias tampão I/O do S/370 e processar as operações I/O do S/370. Os pares de processadores associados executam os seus sistemas operativos respectivos num ambiente de sistema único sem reescrever de maneira significativa nenhum dos sistemas operativos. Nenhum dos sistemas operativos é conhecedor do outro sistema operativo nem do outro par de processadores.

Figura 7

-
-
-

PROCESSOS ACTUAIS DE INTERLIGAÇÃO DE PROCESSADORES

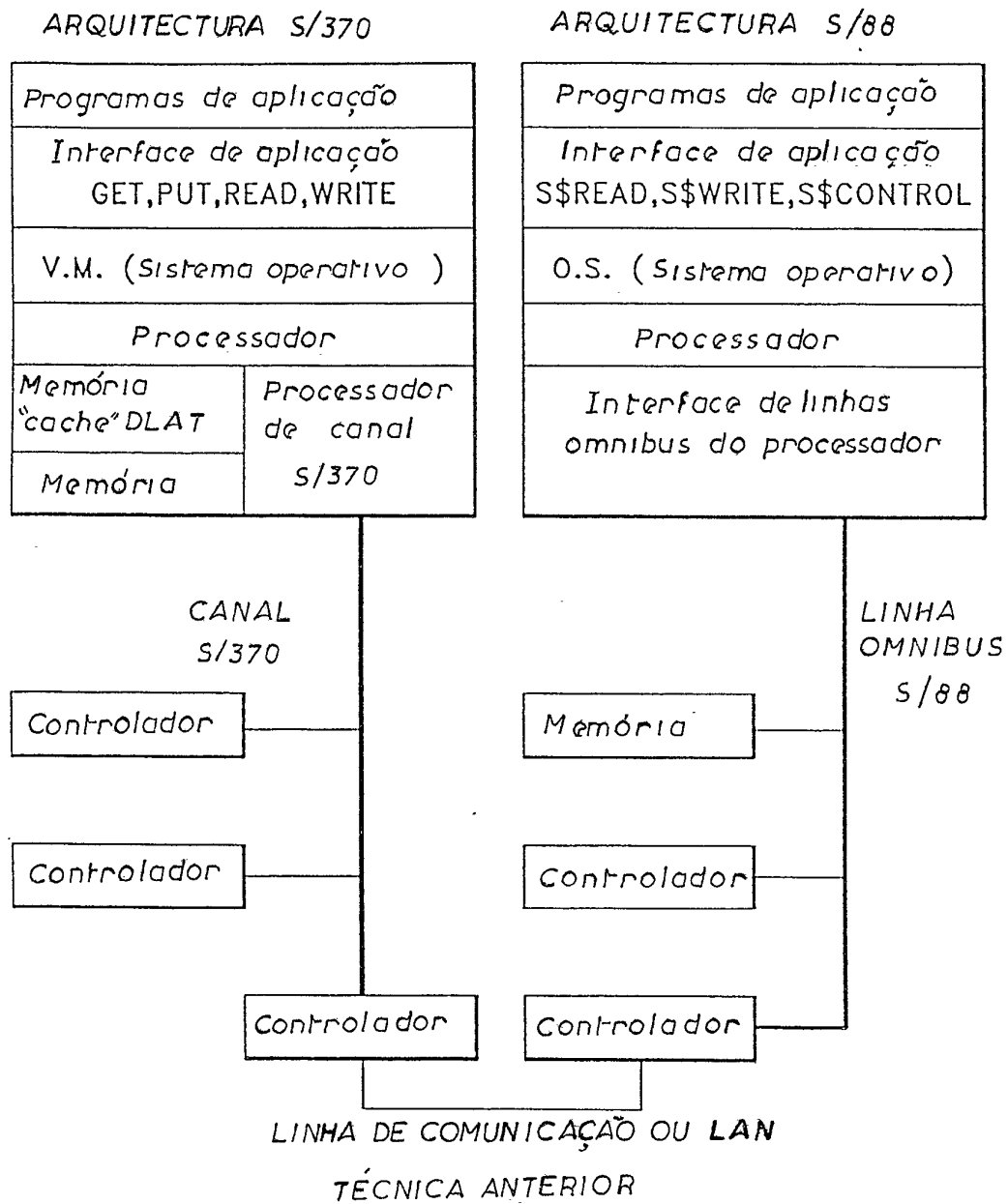


FIG. 1

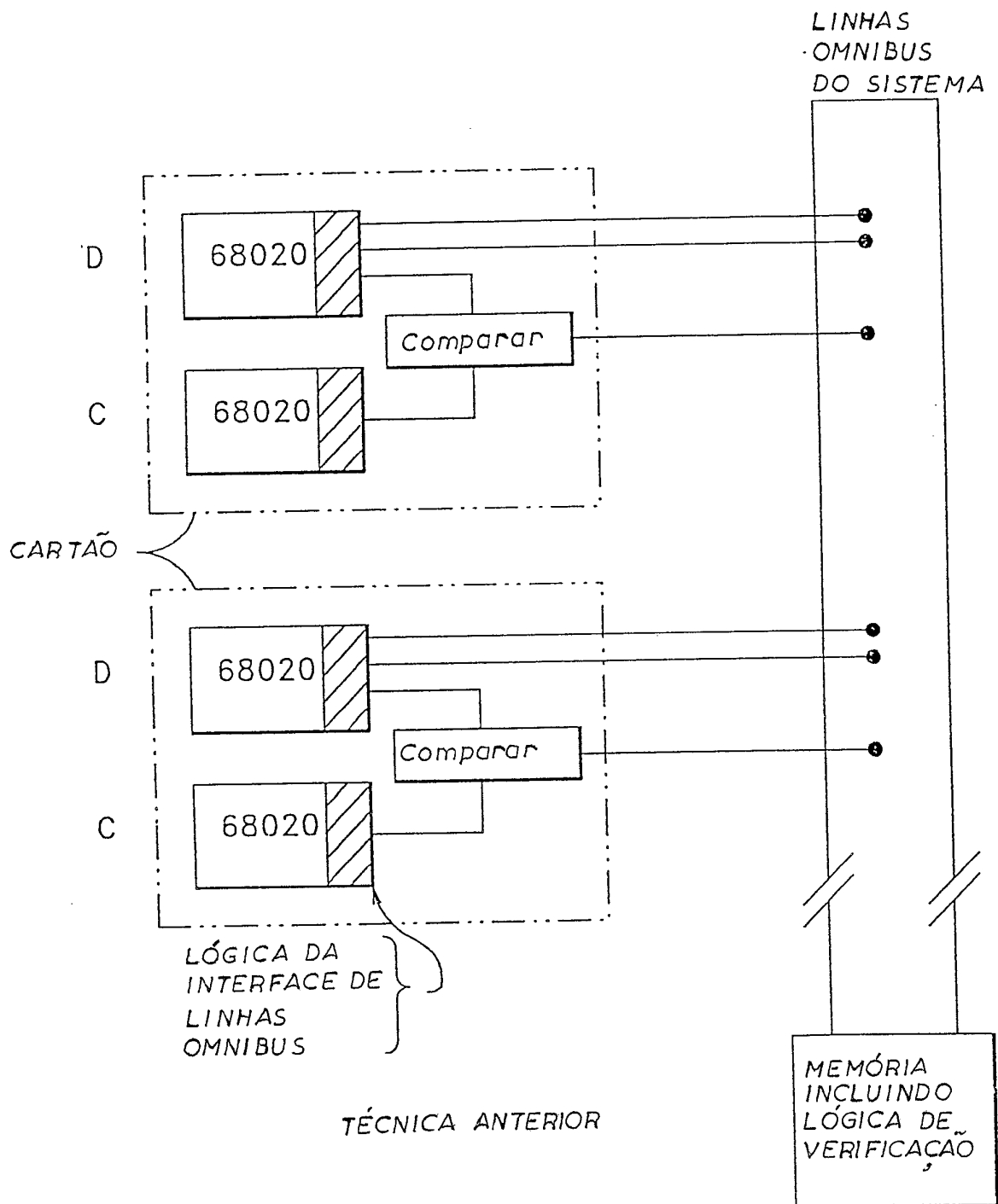


FIG. 2

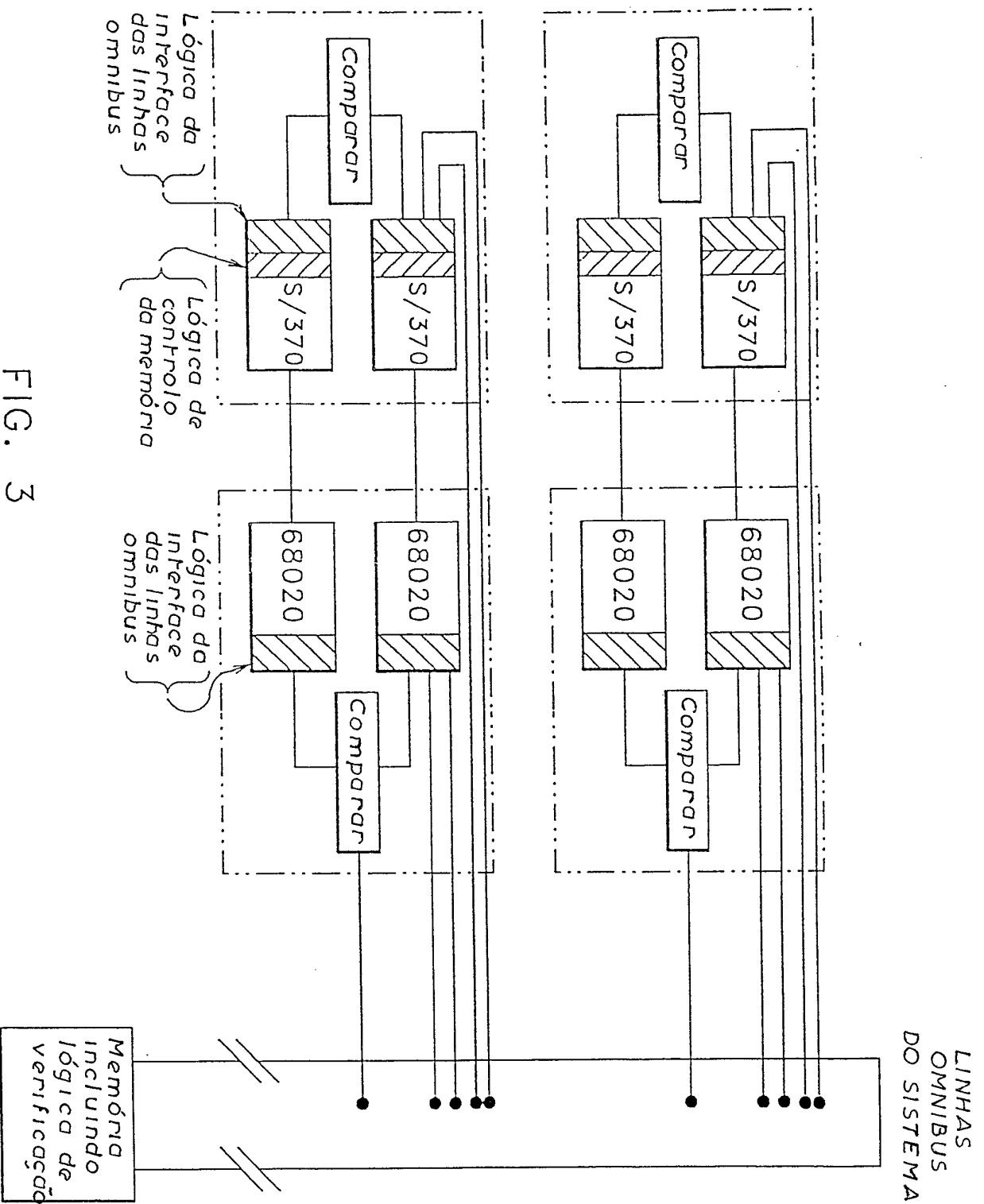


FIG. 3

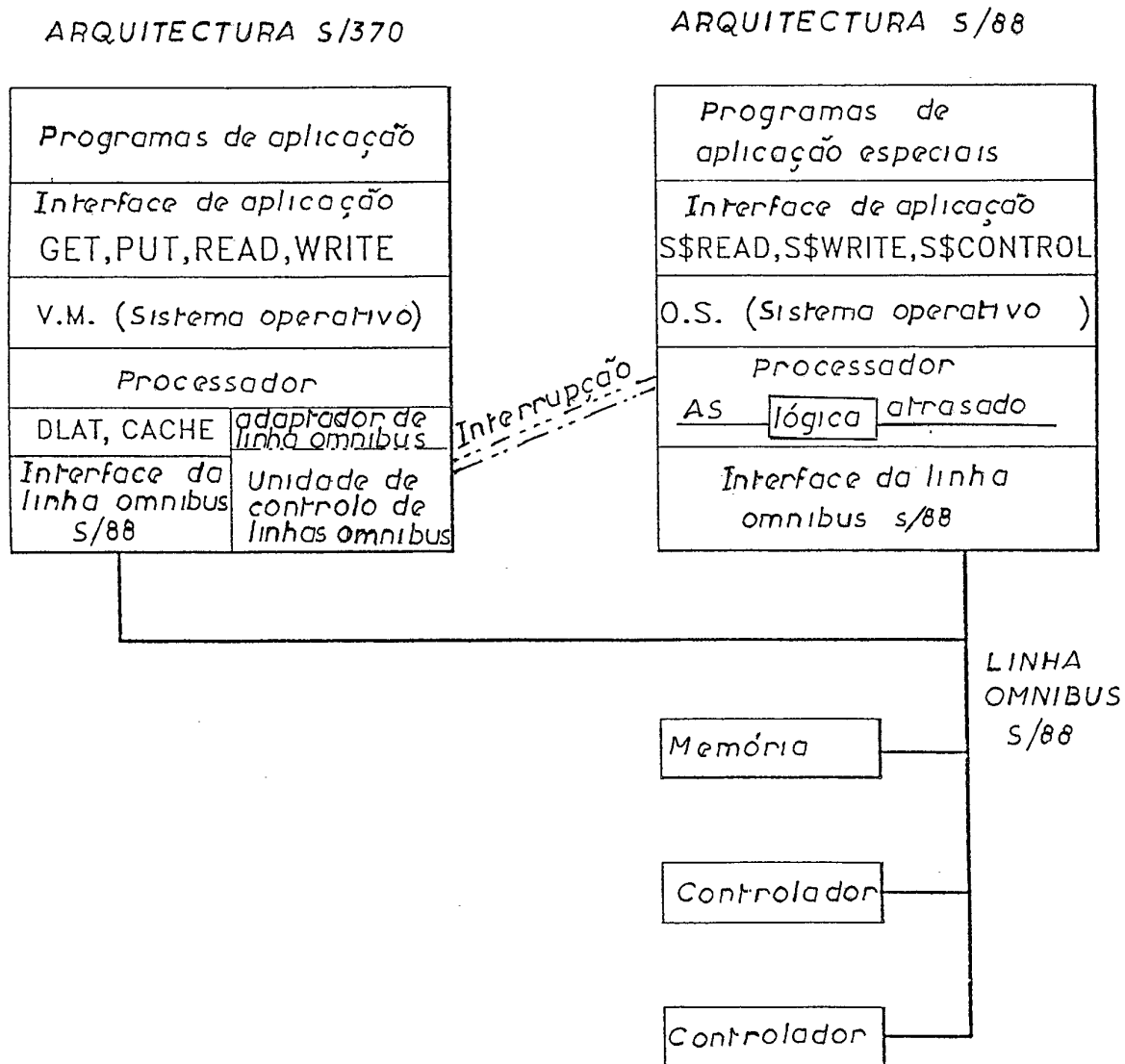


FIG. 4

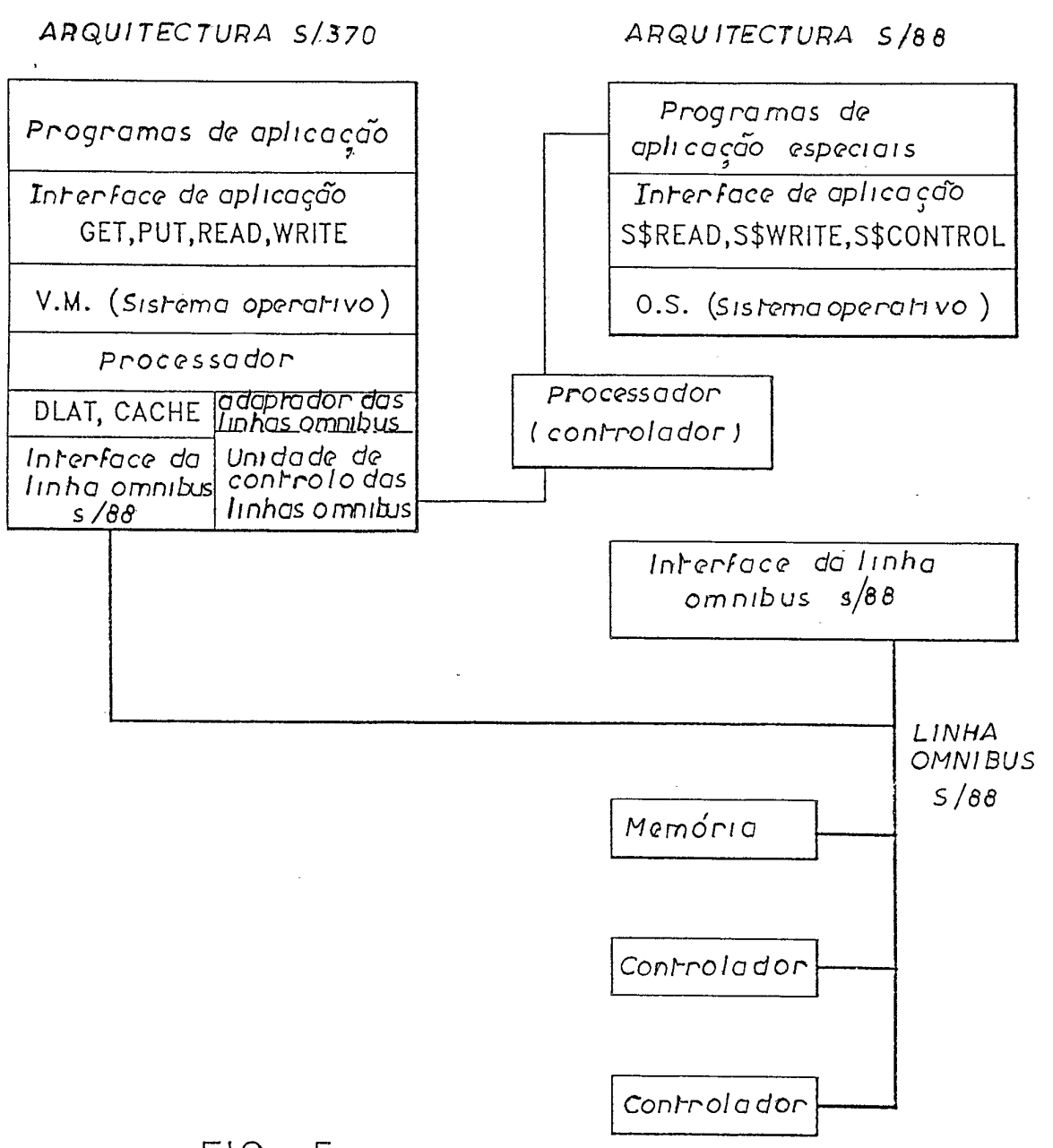
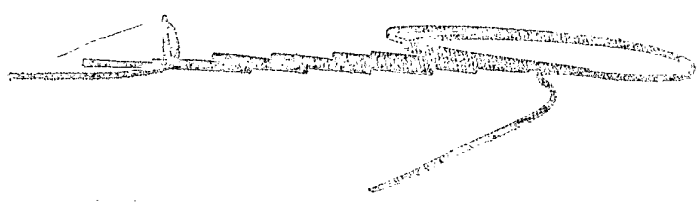
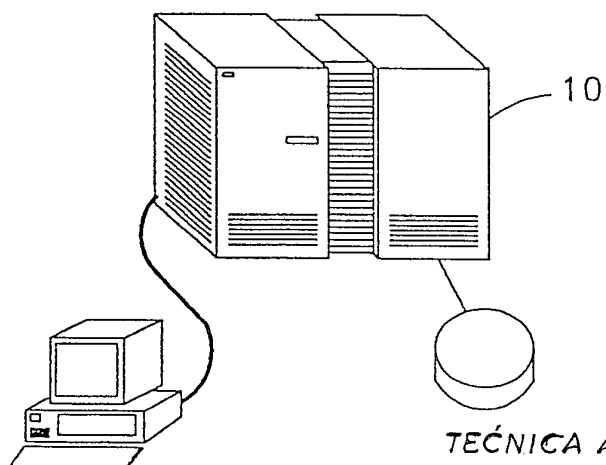


FIG. 5



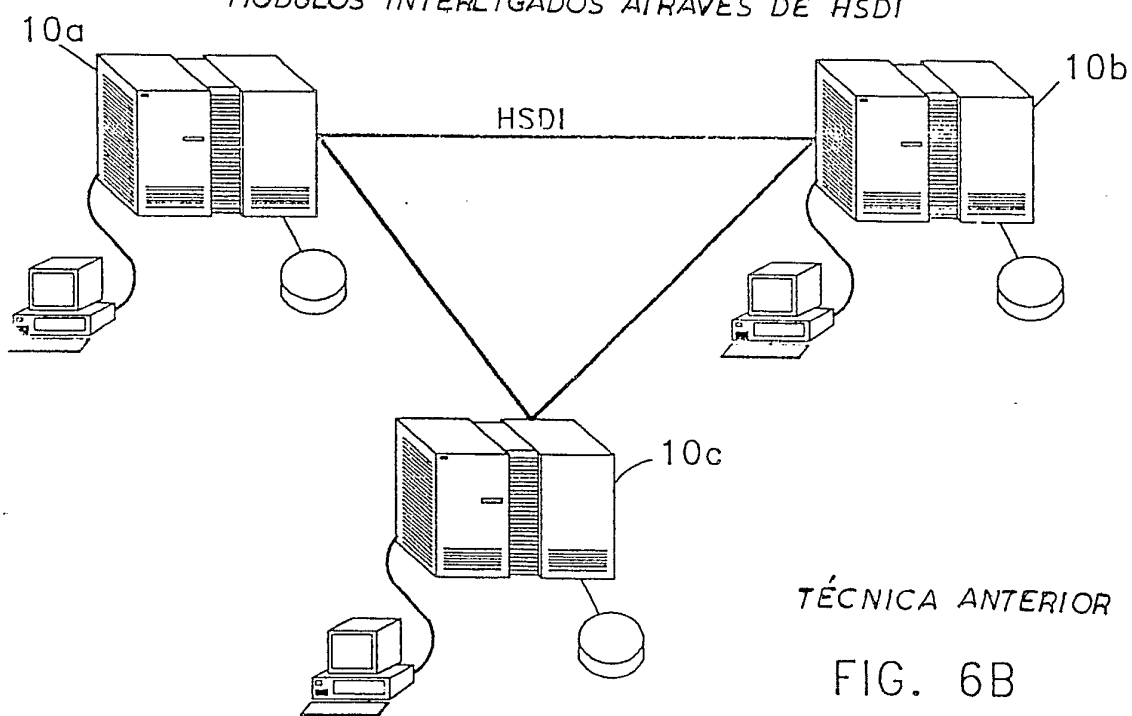
MÓDULO SYSTEM/88



TECNICA ANTERIOR

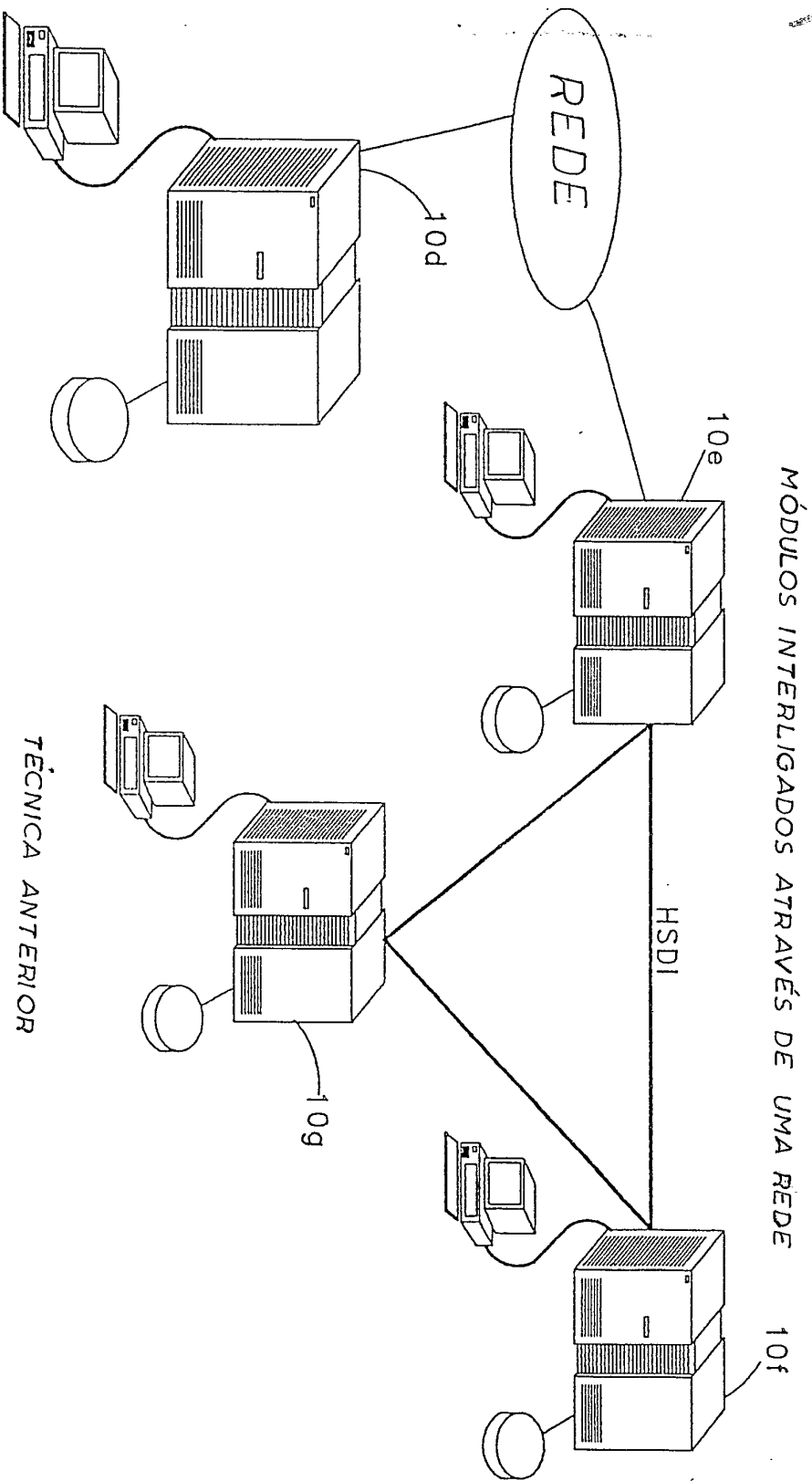
FIG. 6A

MÓDULOS INTERLIGADOS ATRAVÉS DE HSDI



TECNICA ANTERIOR

FIG. 6B



TÉCNICA ANTERIOR

FIG. 6C

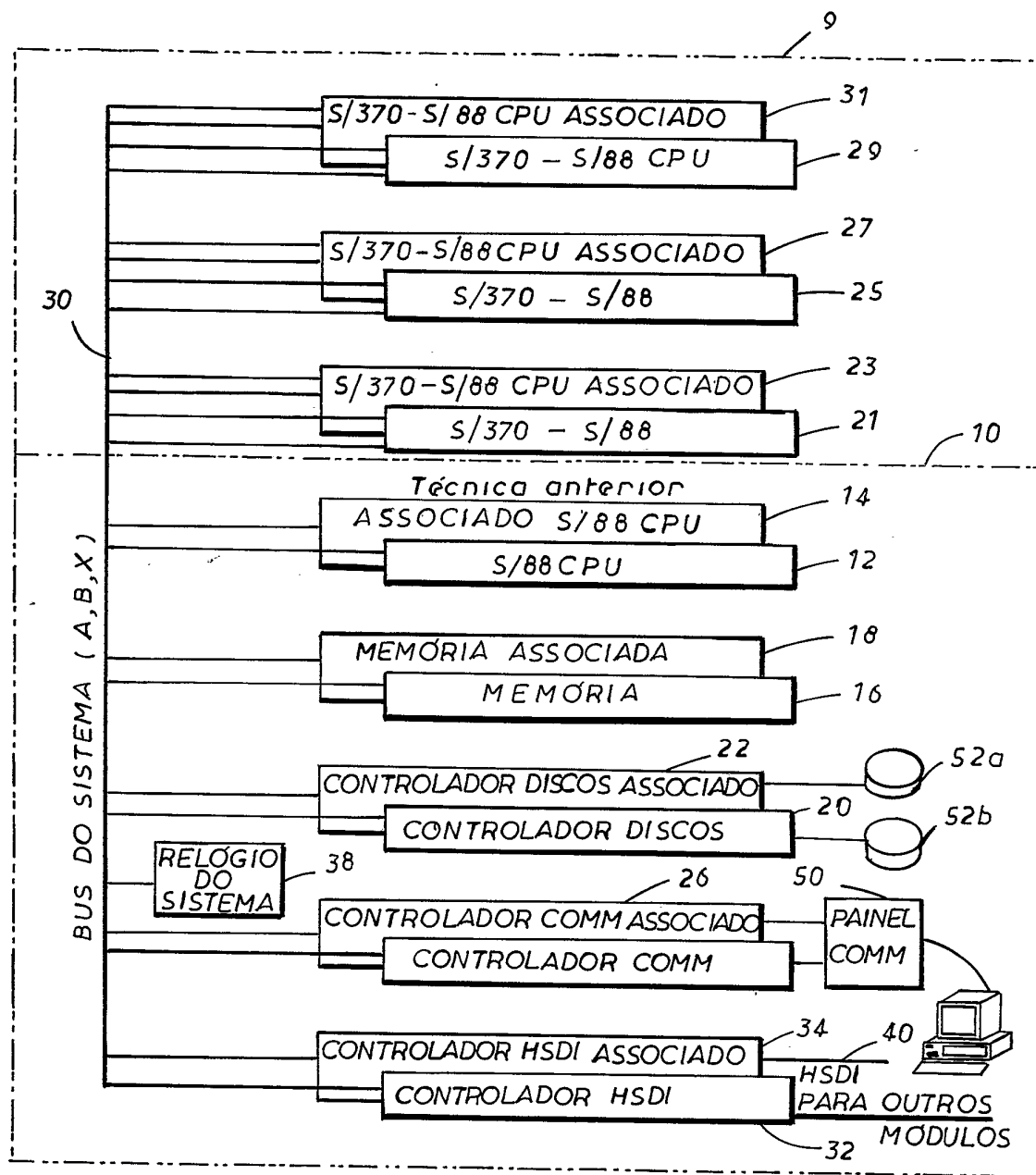


Fig. 7

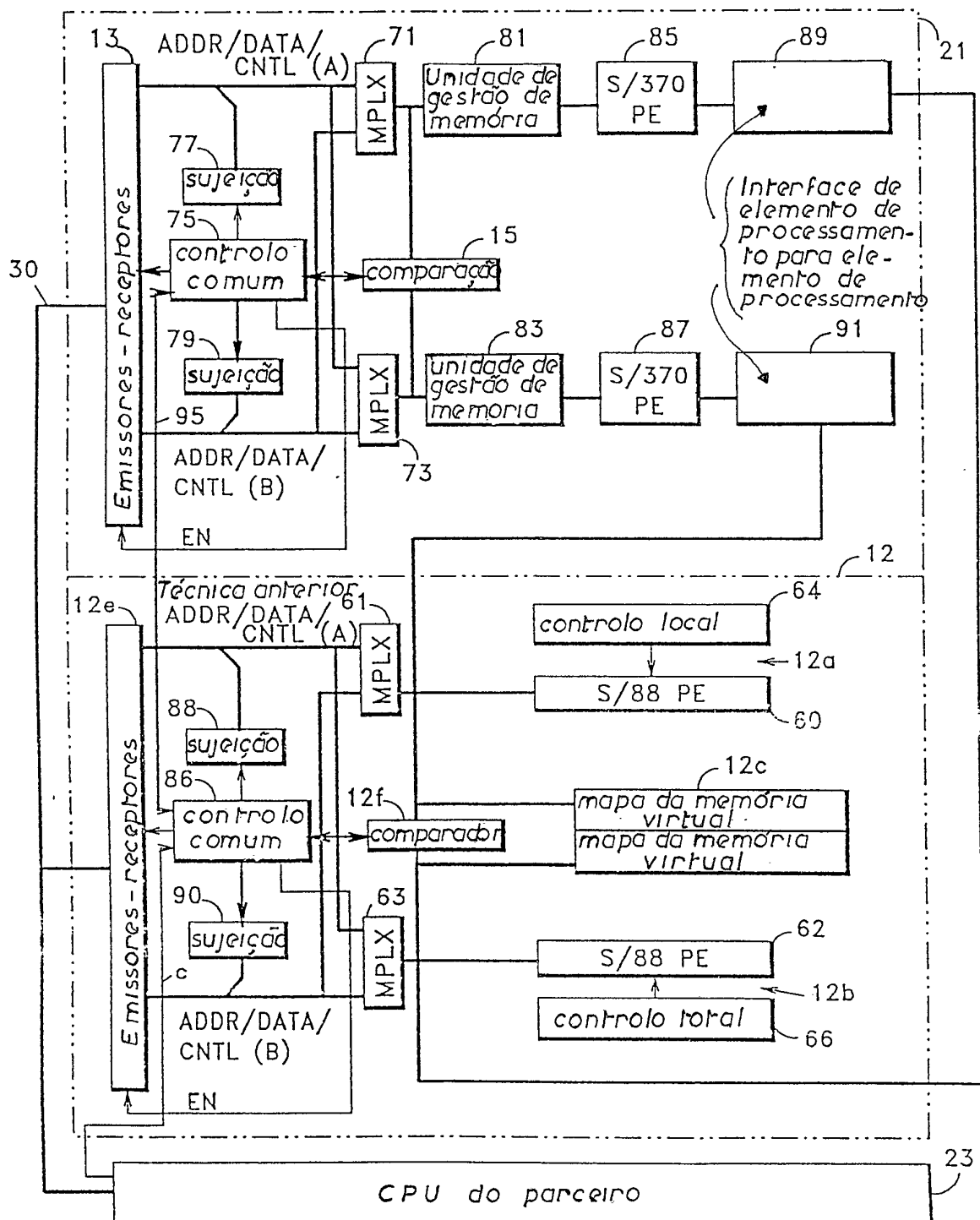
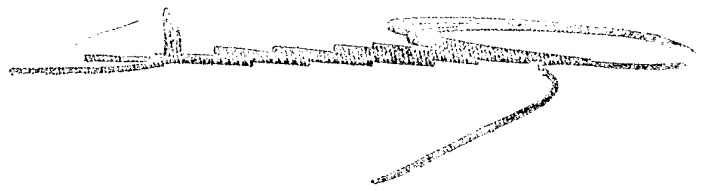


FIG. 8



ACONDICIONAMENTO FÍSICO

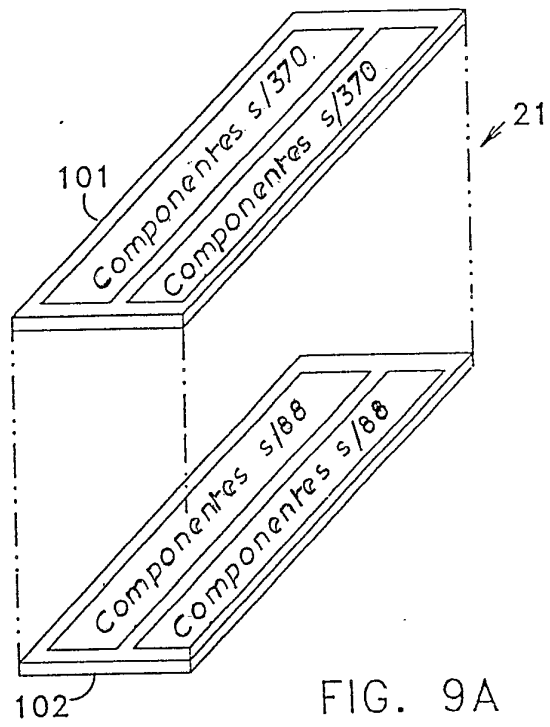


FIG. 9A

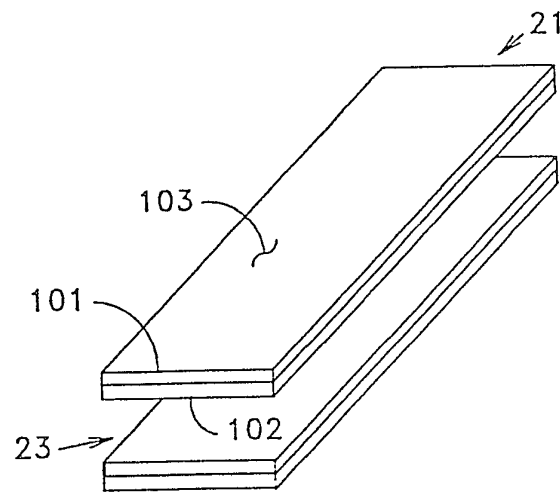


FIG. 9B

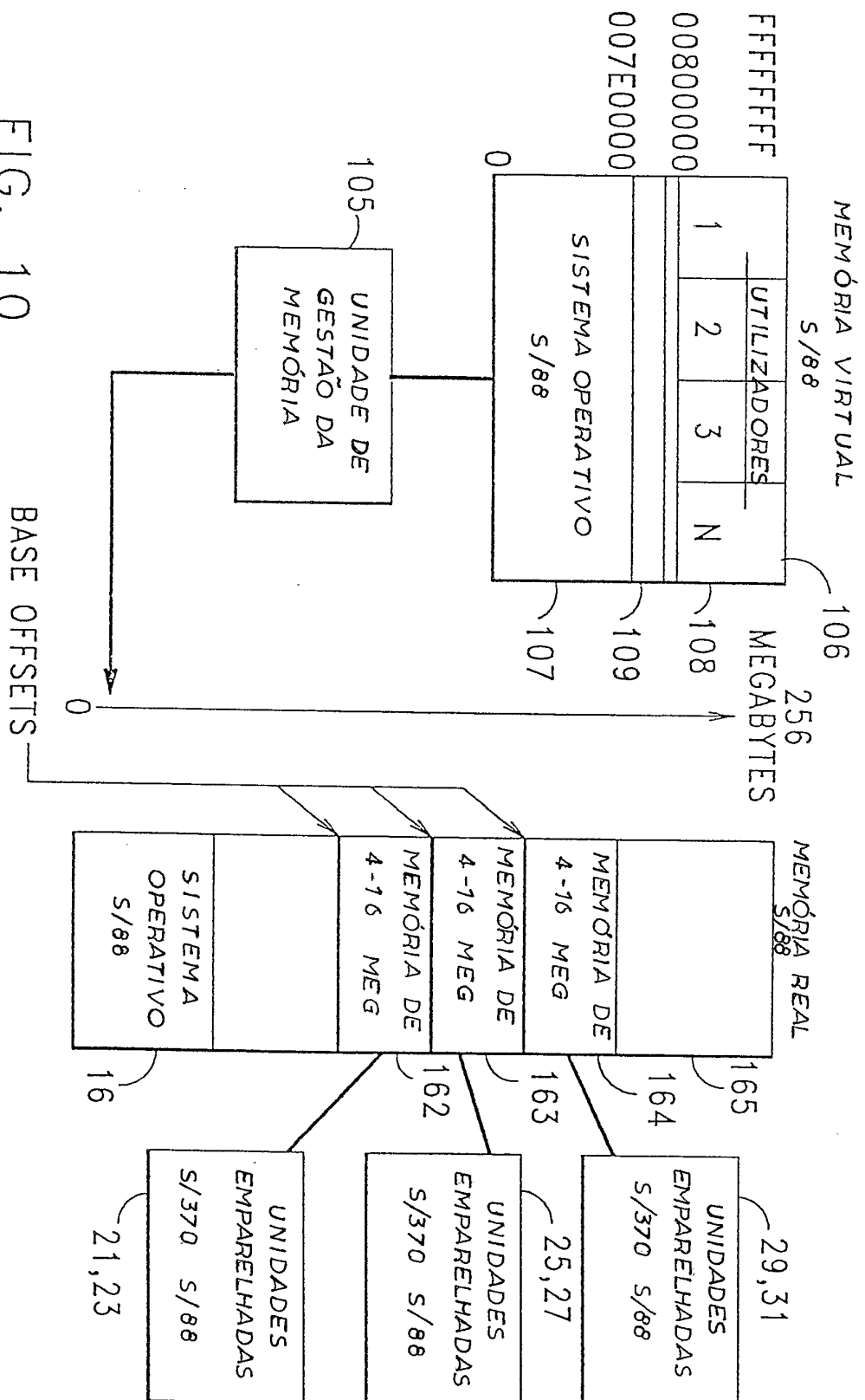


FIG. 10

BASE OFFSETS

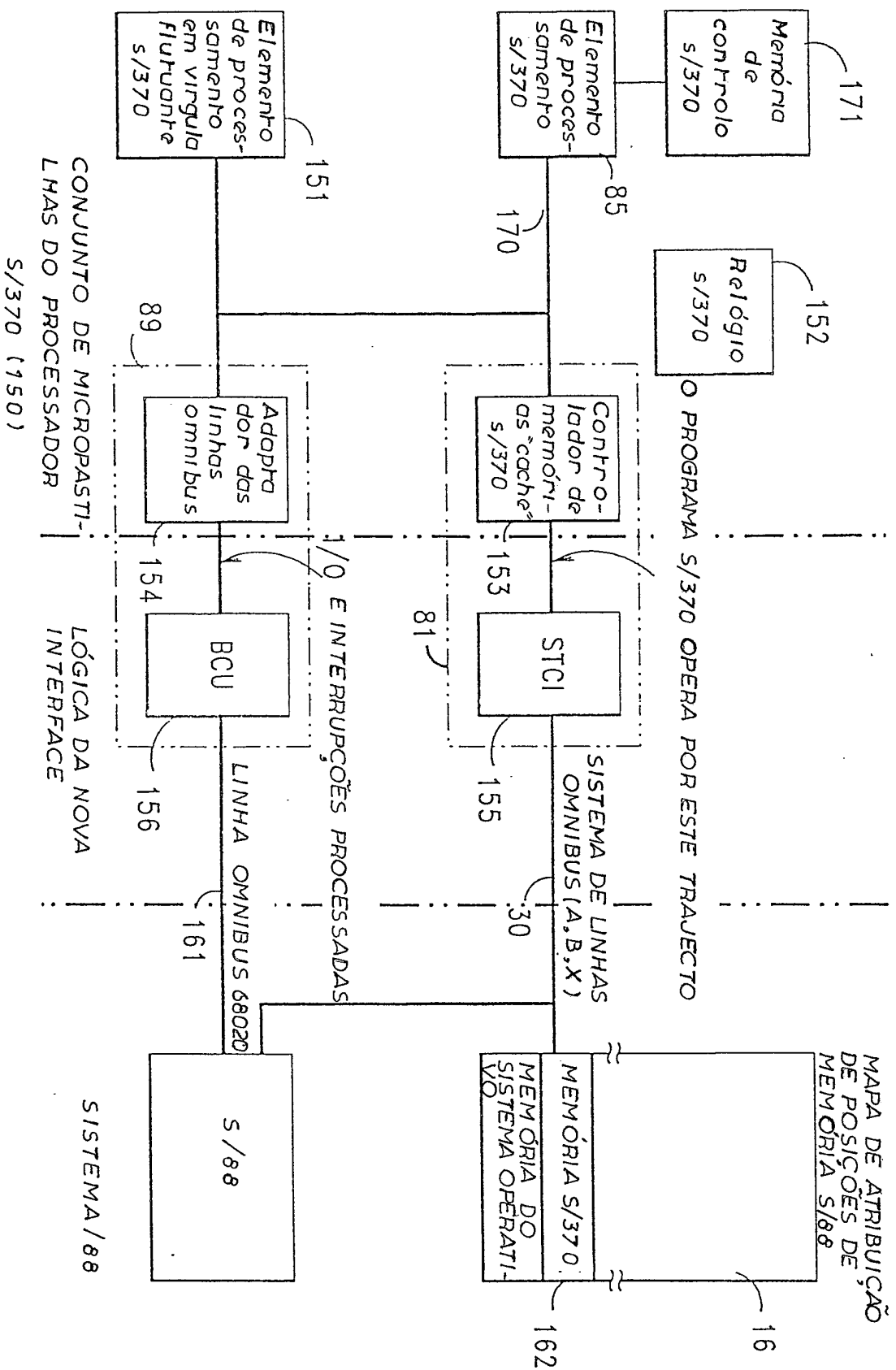


FIG. 11

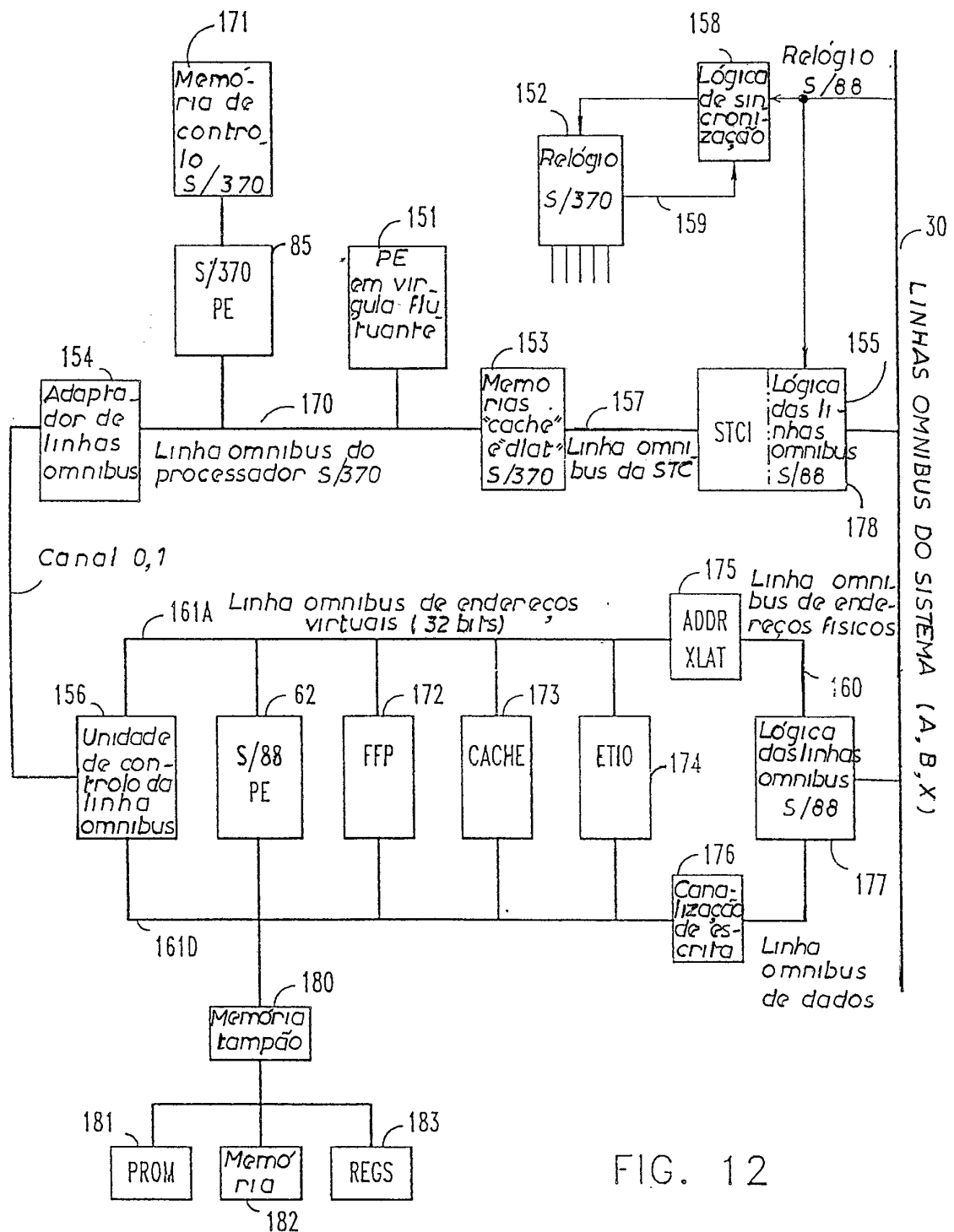


FIG. 12

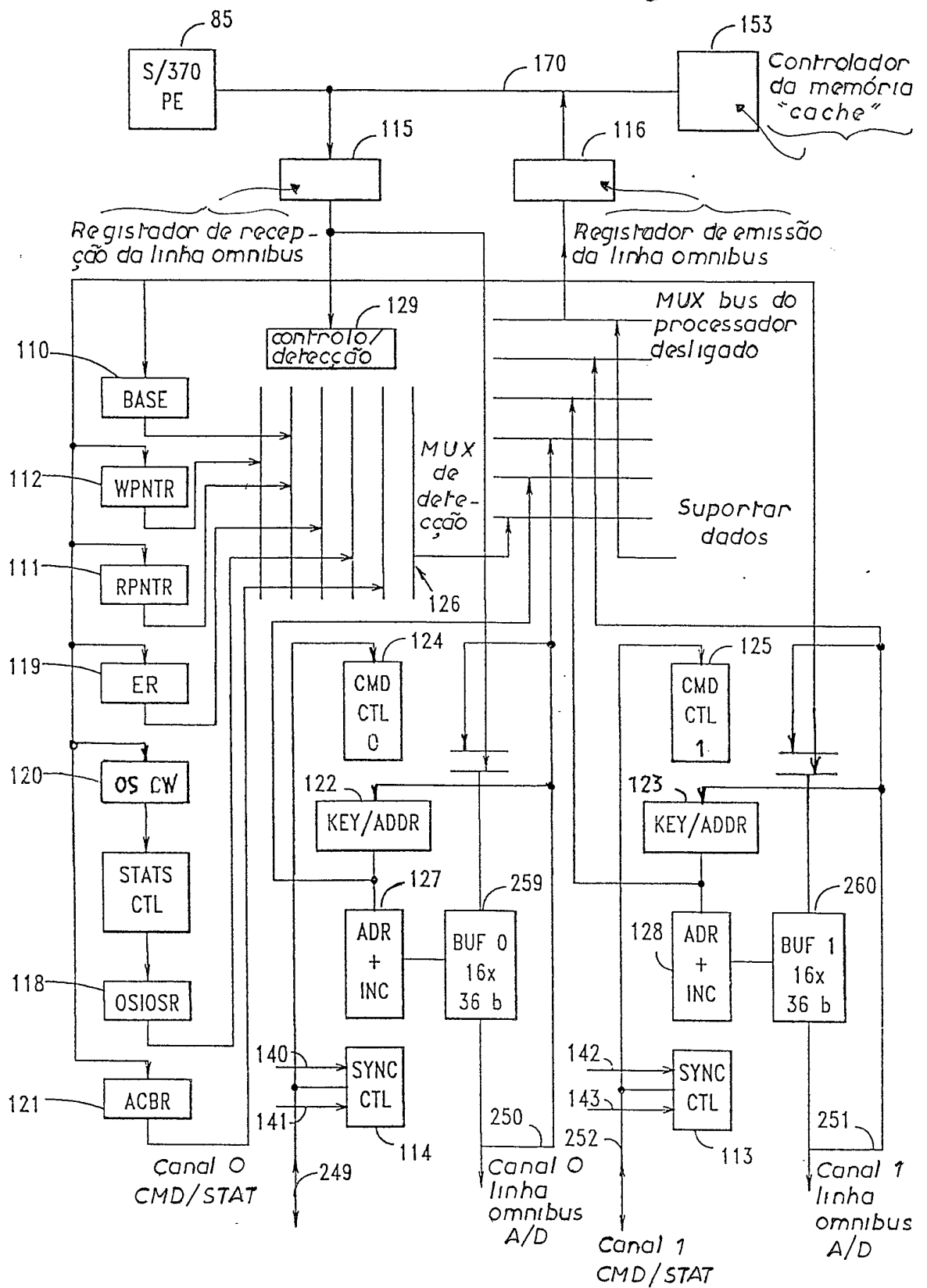


FIG. 13 - Adaptador das linhas omnibus 154

CANAL			
SUBCICLO		1	2
BIT	0123	4567	
CICLO DE SELECÇÃO BSM selecciona em Caixa de correio selecciona em	11ff 110f	ffff ffff	
CICLO DE DADOS saida de dados	0000	0000	
SAIDA DE DADOS/ESTADOS	KAB1	KAB1	

Dados na linha omnibus 249

1		2		
		1	1	2 2 3
0.....7	8.....5	6.....3	4.....1	
kkkk1ppc	aaaaaaaa	aaaaaaaa	aaaaaaaa	
kkkk001c			00000	
dddddddd	dddddddd	dddddddd	dddddddd	
dddddddd	dddddddd	dddddddd	dddddddd	

Dados na linha omnibus 250

FIG. 14A



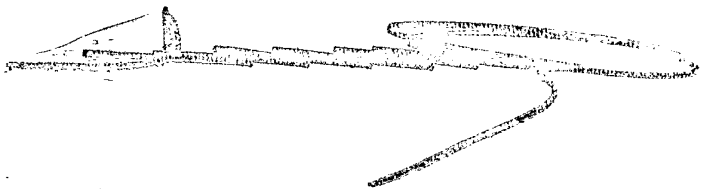
CANAL SUBCICLO	1	2
BIT	0123	4567
CICLO DE SELECÇÃO :		
BSM selecciona em	01ff	ffff
Caixa de correio selecciona em	110f	ffff
Fila de espera selecciona	110f	ffff
CICLO DE DADOS :		
Entrada de dados	////	////
Saída de dados	KAB1	KAB1

Dados na linha omnibus 252

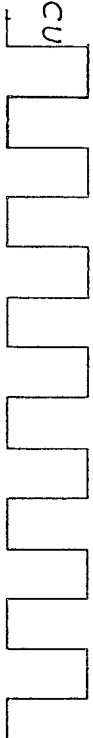
1		2	
0.....7	8.....5	11 6.....3	22 4.....1
kkkk1ppc	aaaaaaaa	aaaaaaaa	aaaaaaaa
kkkk001c			...00000
kkkk001c			...00000
aaaaaaaa	aaaaaaaa	aaaaaaaa	aaaaaaaa
////////	////////	////////	////////

Dados na linha omnibus 251

FIG. 14B



RELÓGIO DE BCU



DADOS DE SAÍDA DA BCU



ETIQUETA NO SENTIDO
ASCENDENTE
(SAÍDA DA BCU (2620))



FIG. 15A — Armazenamento

[Handwritten signature]

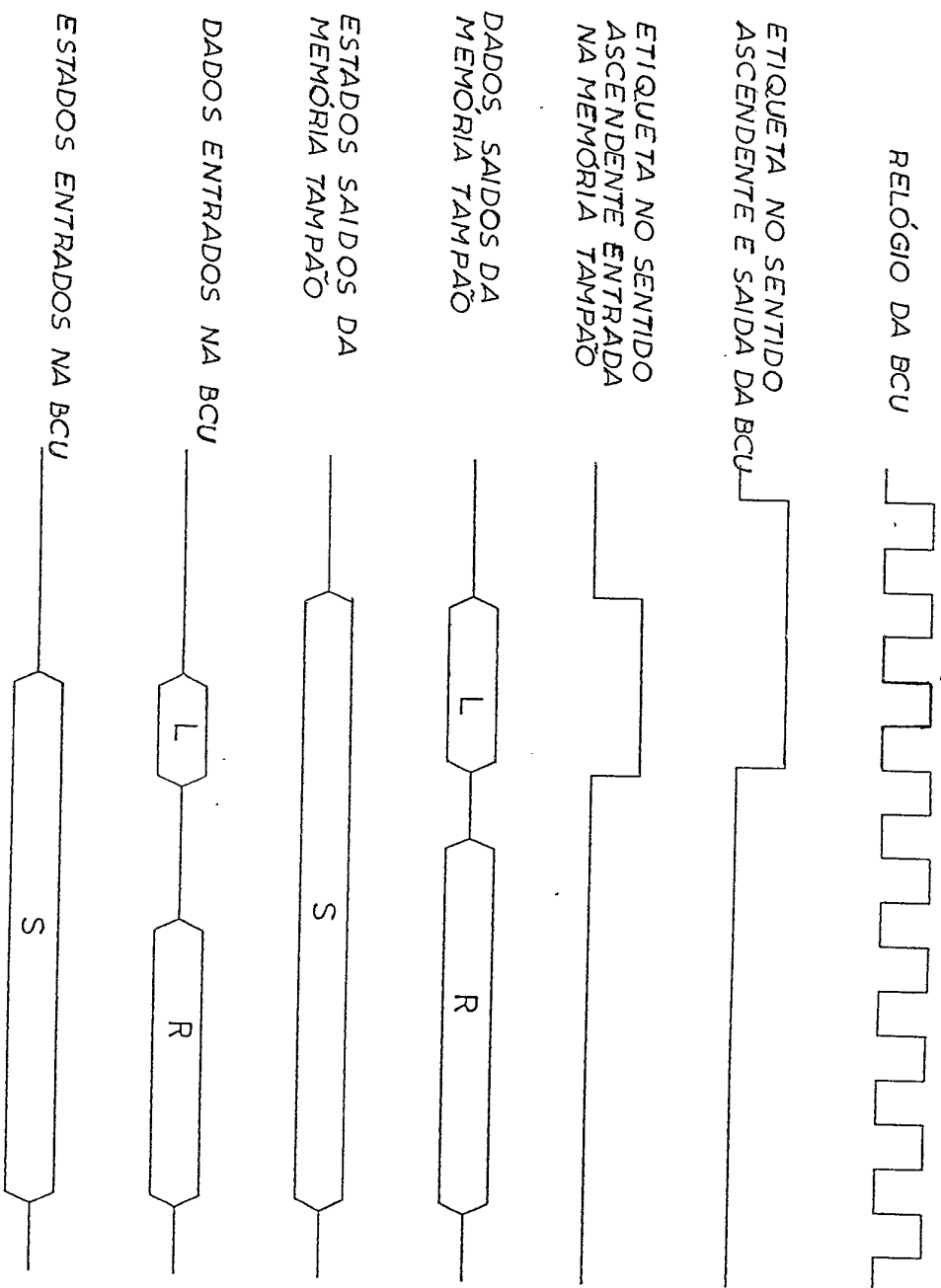


FIG. 15B - Busca

[Handwritten signature]

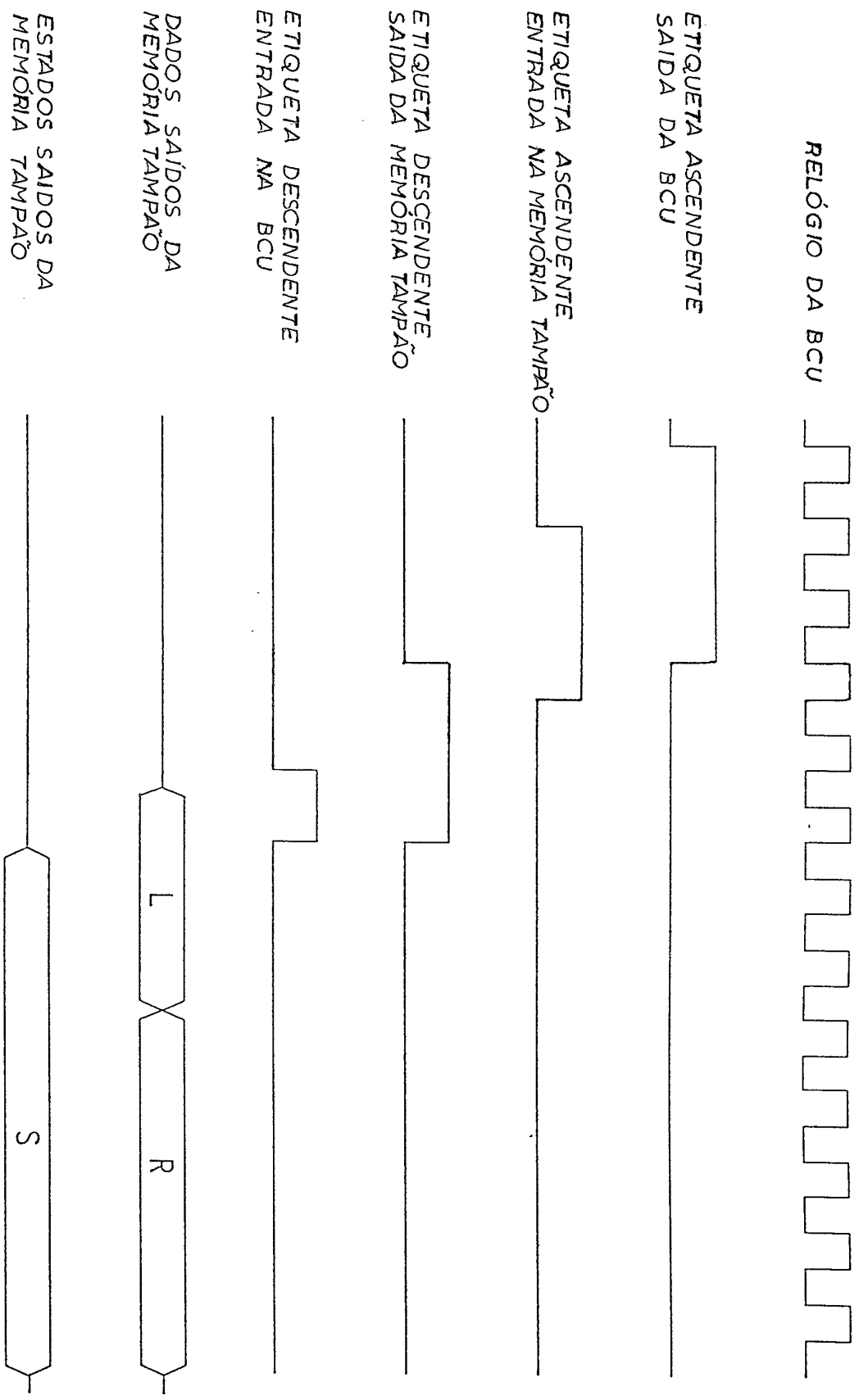


FIG. 15C -Busca (dados retardados)

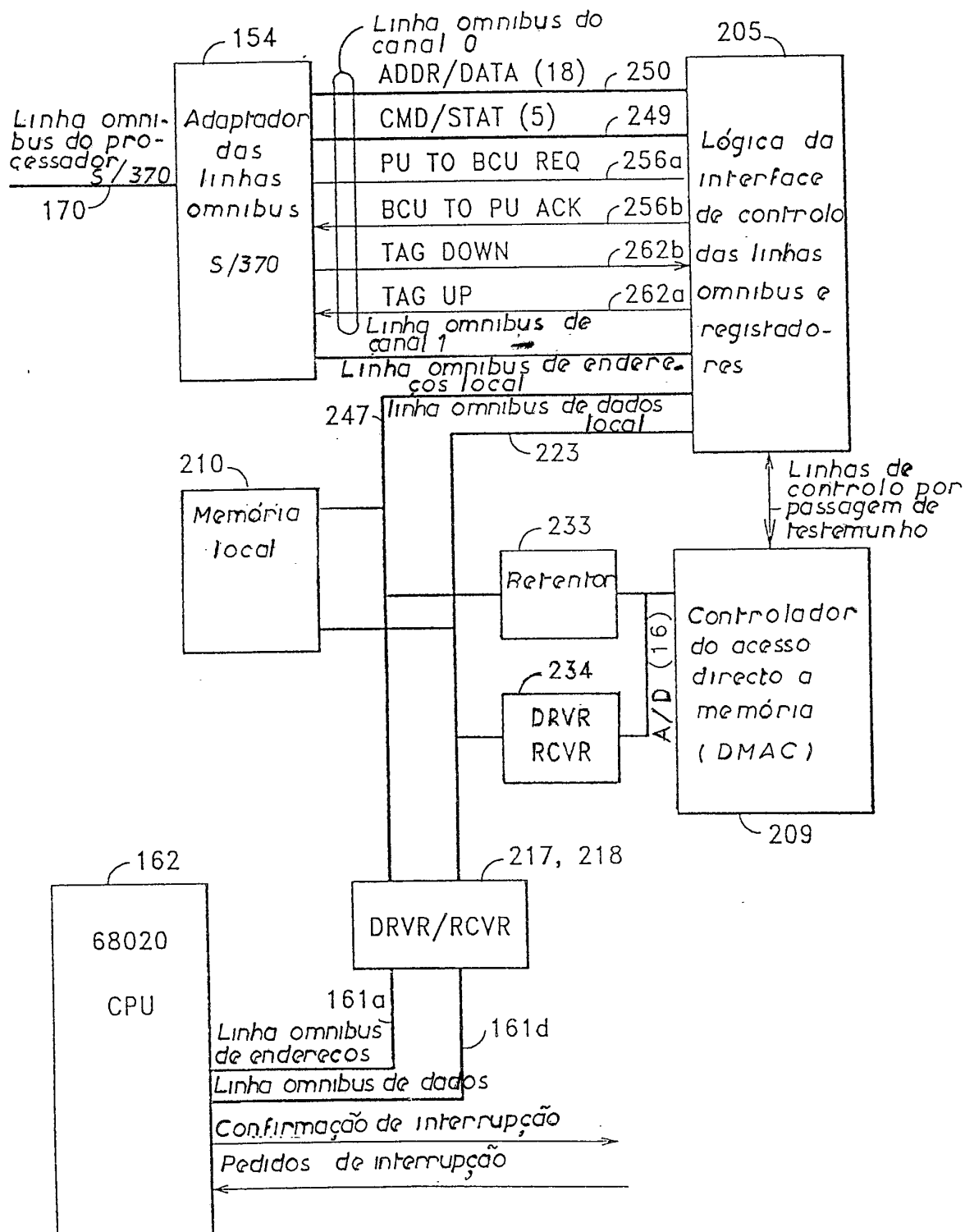


FIG. 16

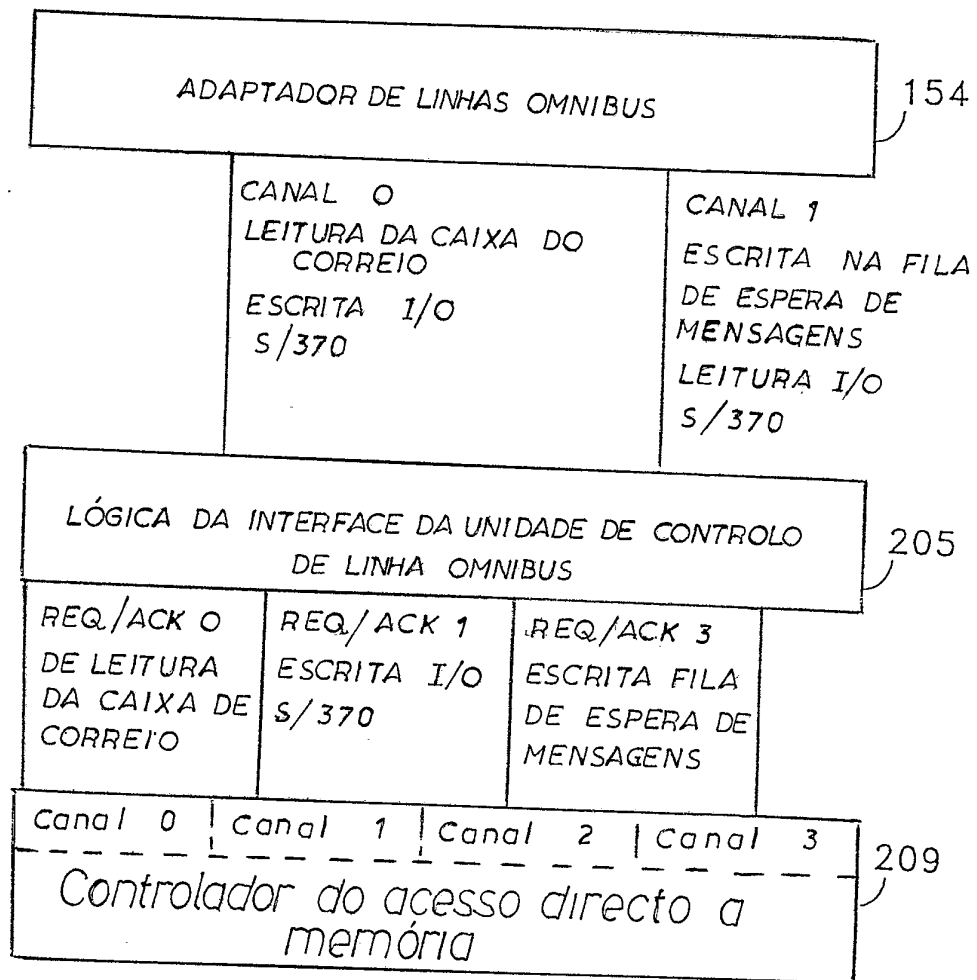
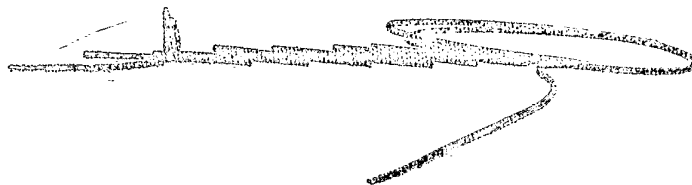


FIG. 17.



ENDEREÇOS		CHO	CH1	CH2	CH3
CSR	00 estado do canal				
CER	01 erro no canal				
DCR	04 controlo de dispositivos	A9	A8	A8	A8
OCR	05 controlo de operação	9E	92	12	12
SCR	06 controlo de sequência	04	04	04	04
CCR	07 controlo do canal	08	08	08	08
NIV	25 controlo normal	VV	VV	VV	VV
EIV	27 vector de erro	VV	VV	VV	VV
CPR	2D prioridade dos canais	00	02	02	01
MFC	29 códigos de funcionamento da memória	04	04	04	04
DFC	31 códigos de funcionamento dos dispositivos	00	00	00	00
BFC	39 códigos de funcionamento de base	01	00	00	00
MTC	0A/B contagem de transferências 0A/memória B				
BTC	1A/B contagem de transferências 1A/base B				
MAR	0C-F endereço de memória				
DAR	14-7 endereço de dispositivo				
BAR	1C-F endereço de base				
GCR	FF controlo geral				

FIG. 18-Registadores do DMAC

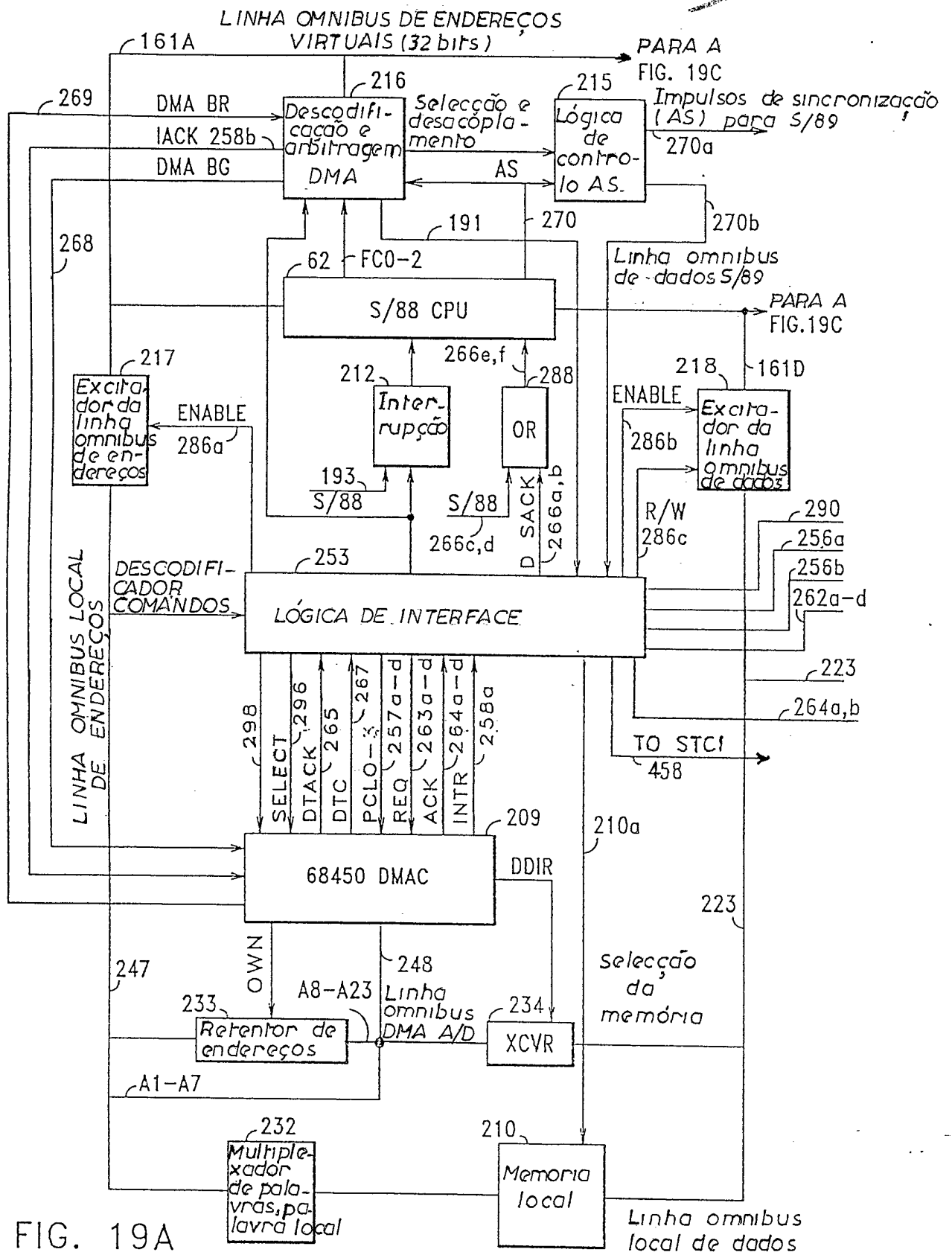


FIG. 19A

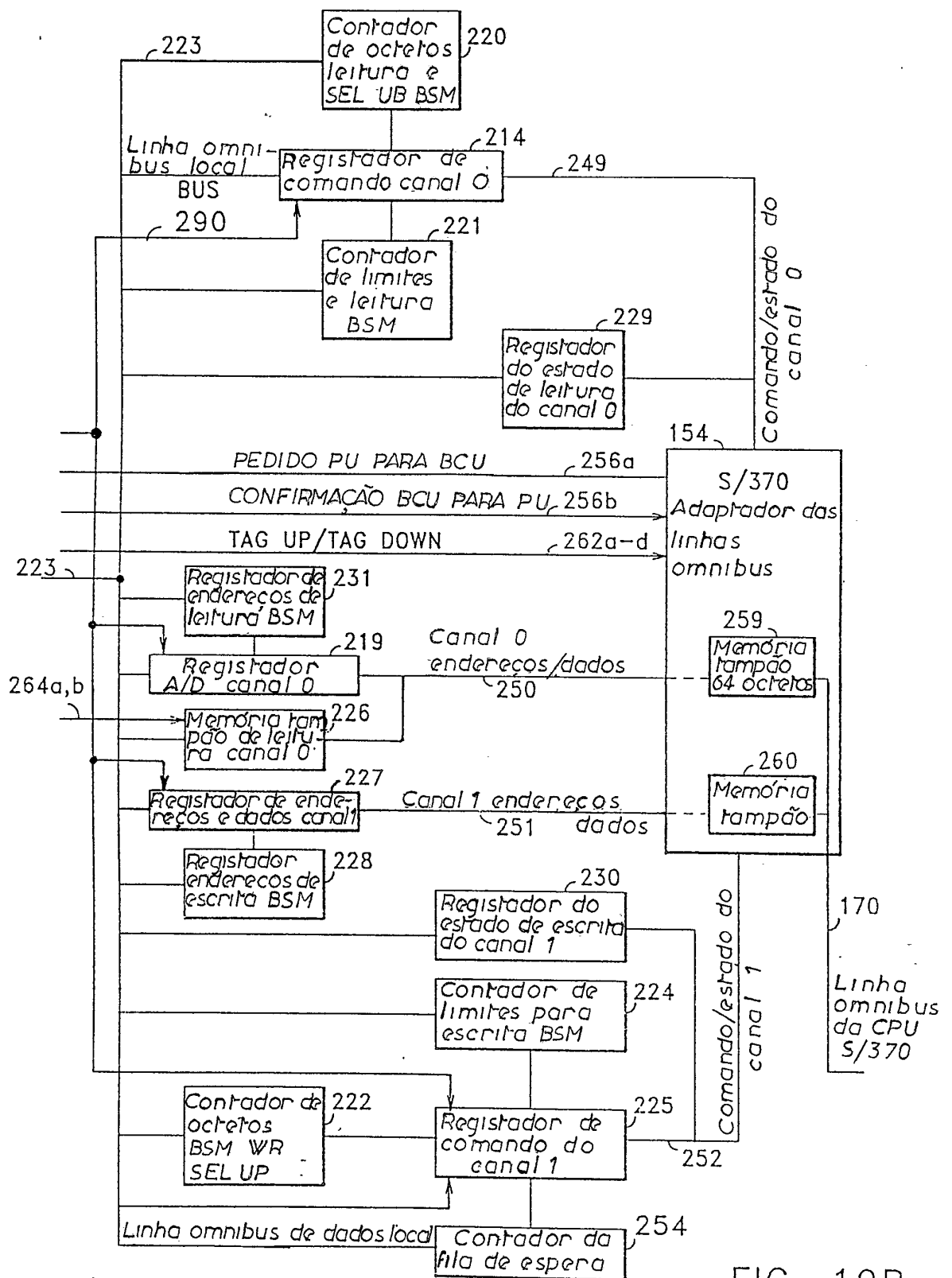


FIG. 19B

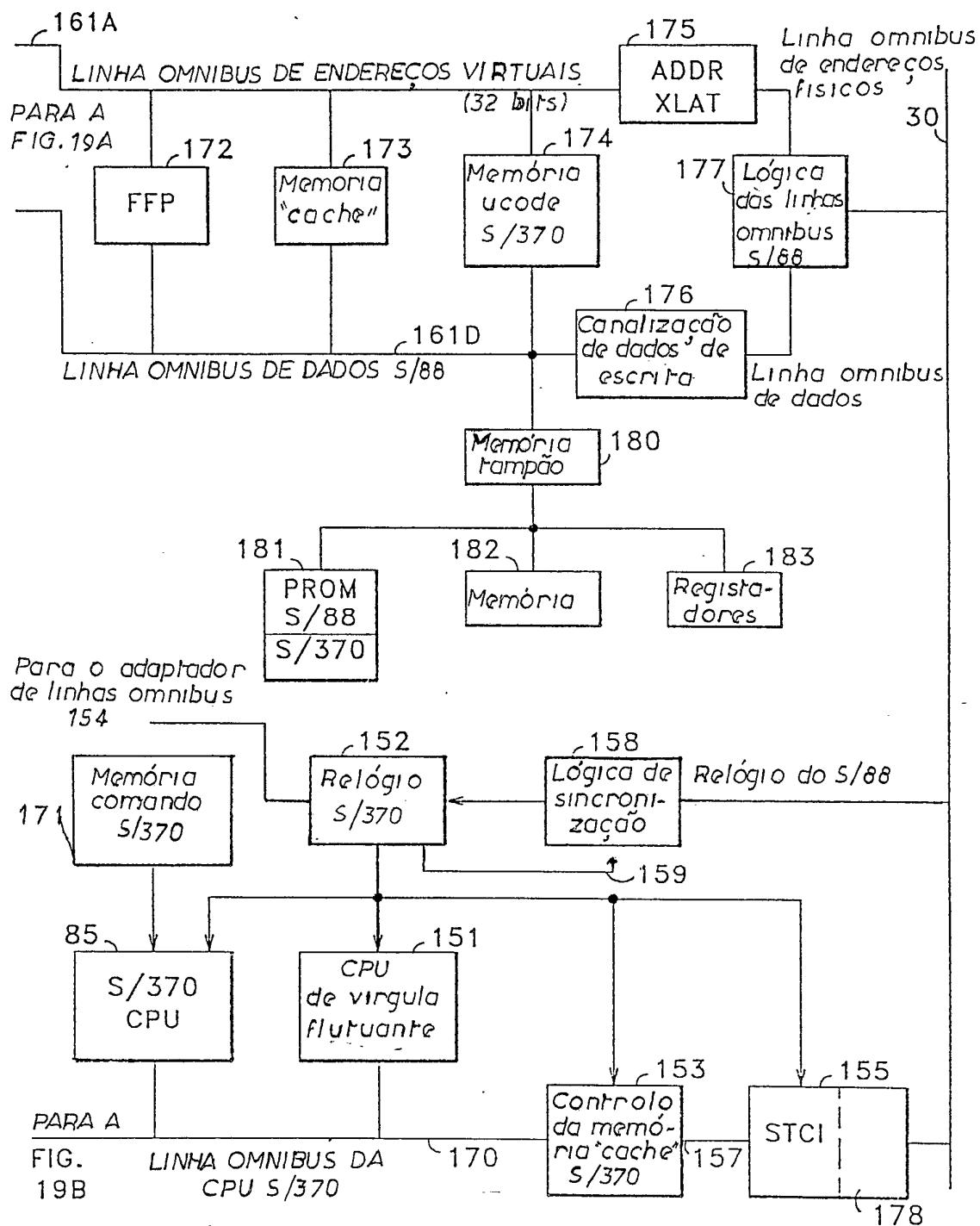


FIG. 19A	FIG. 19B	FIG. 19C

FIG. 19

FIG. 19C

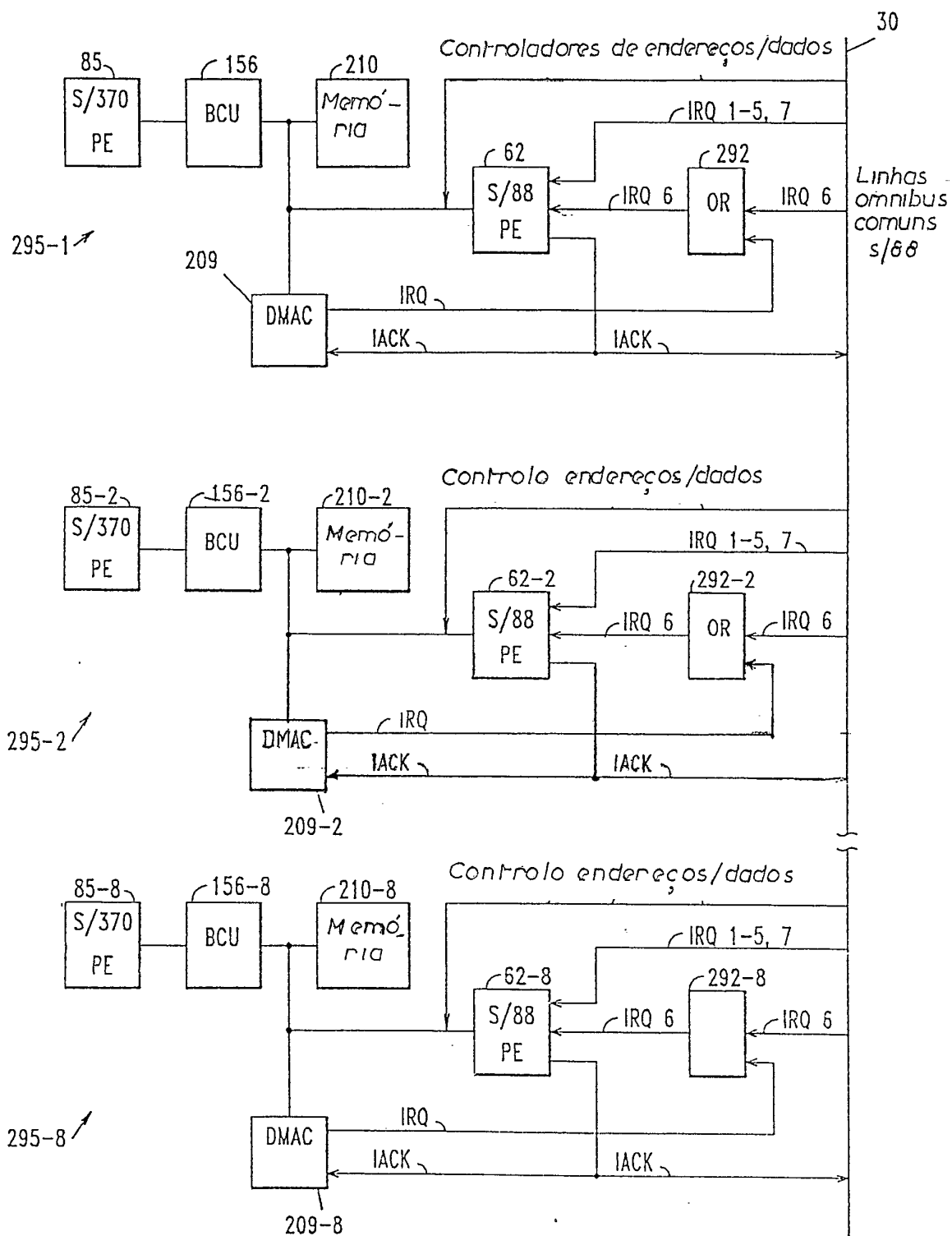
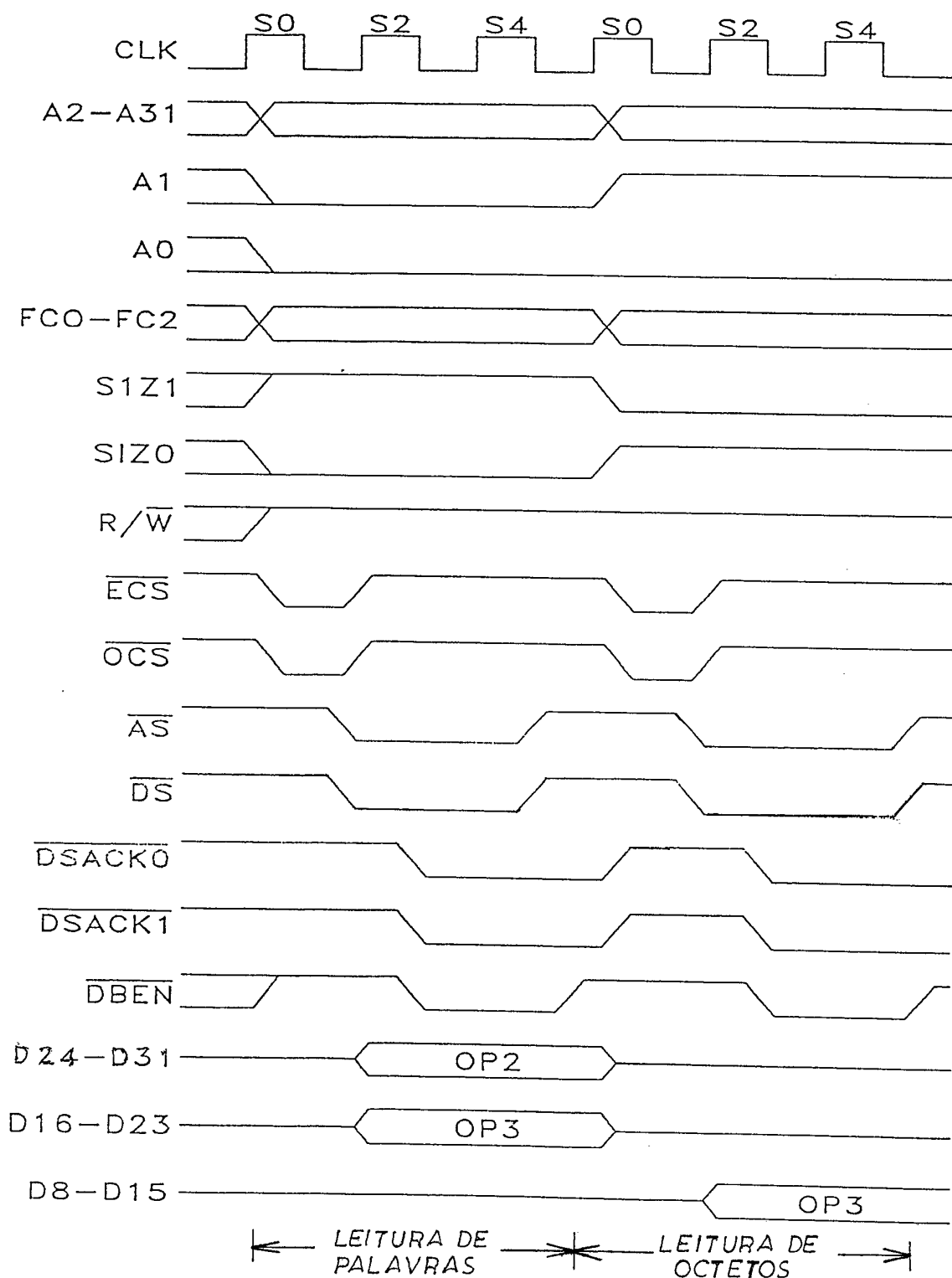
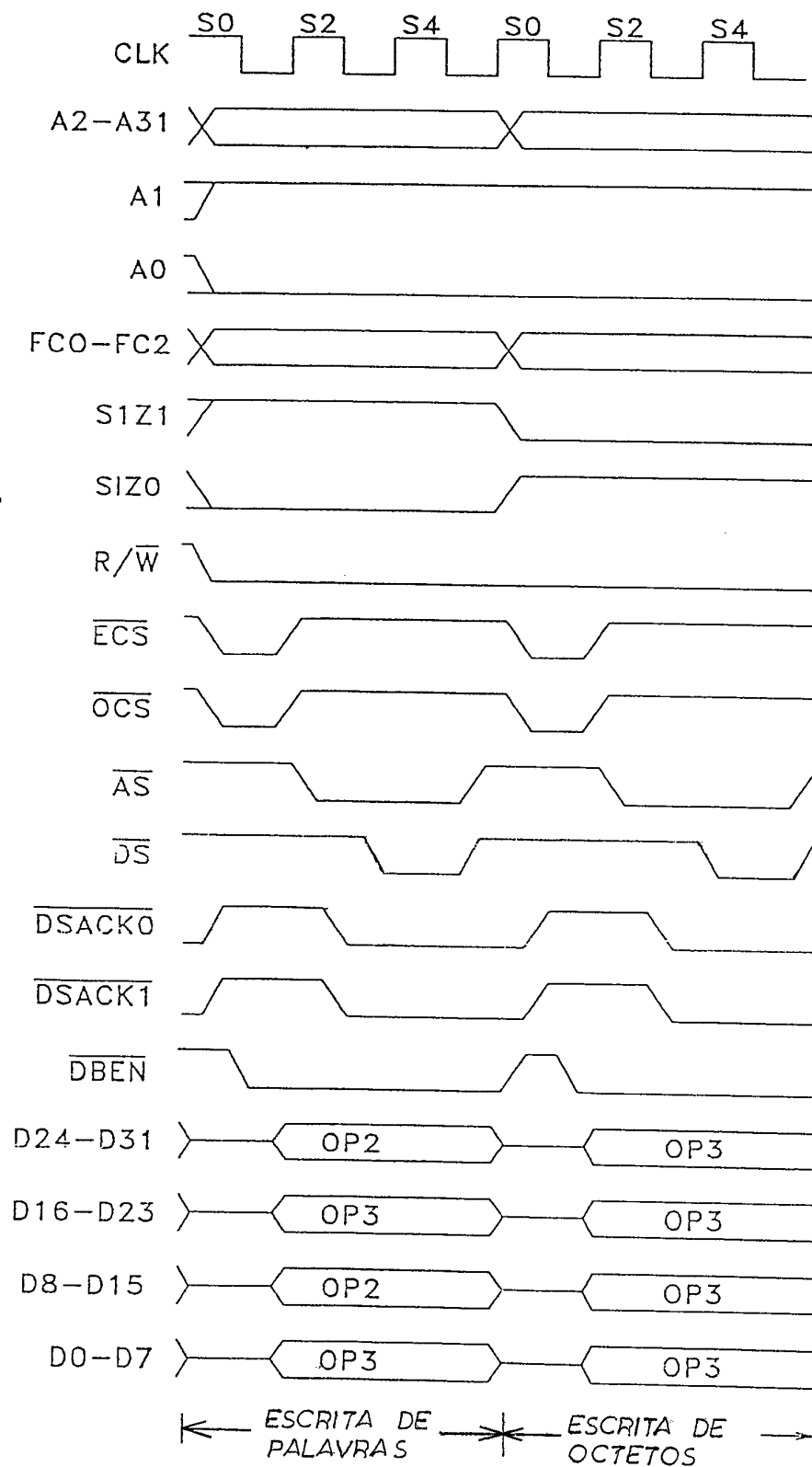


FIG. 21



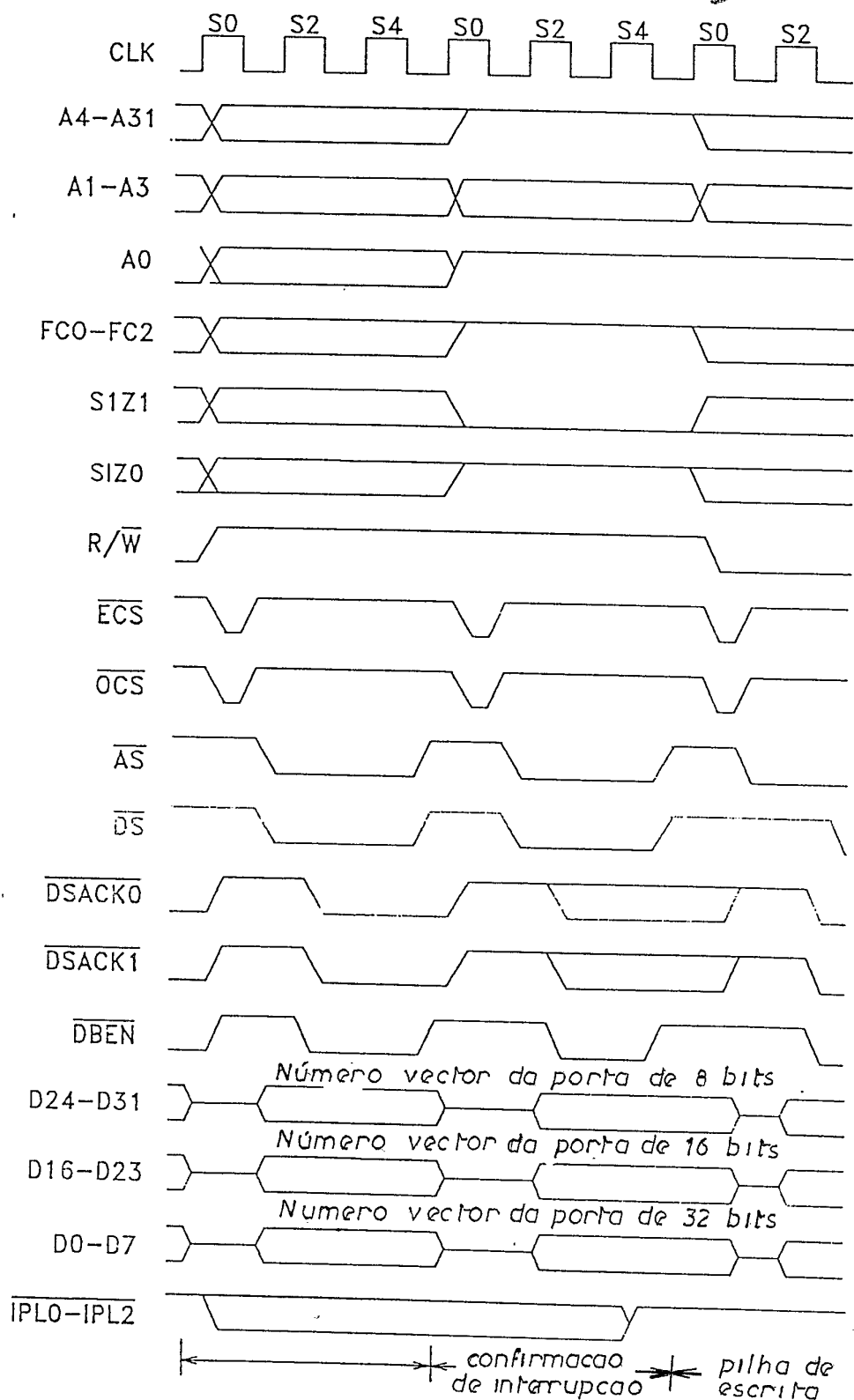
DISTRIBUIÇÃO DE TEMPOS DO CICLO DE LEITURA DE
OCTETOS E PALAVRAS
(PORTA DE DADOS DE 32 bits)

FIG. 22



DISTRIBUIÇÃO DE TEMPOS DO CICLO DE
ESCRITA DE OCTETOS E DE PALAVRAS
(PORTA DE DADOS DE 32 bits)

FIG. 23



DISTRIBUIÇÃO DE TEMPOS DO CICLO DE CONFIRMAÇÃO DE INTERRUPÇÃO

FIG. 24

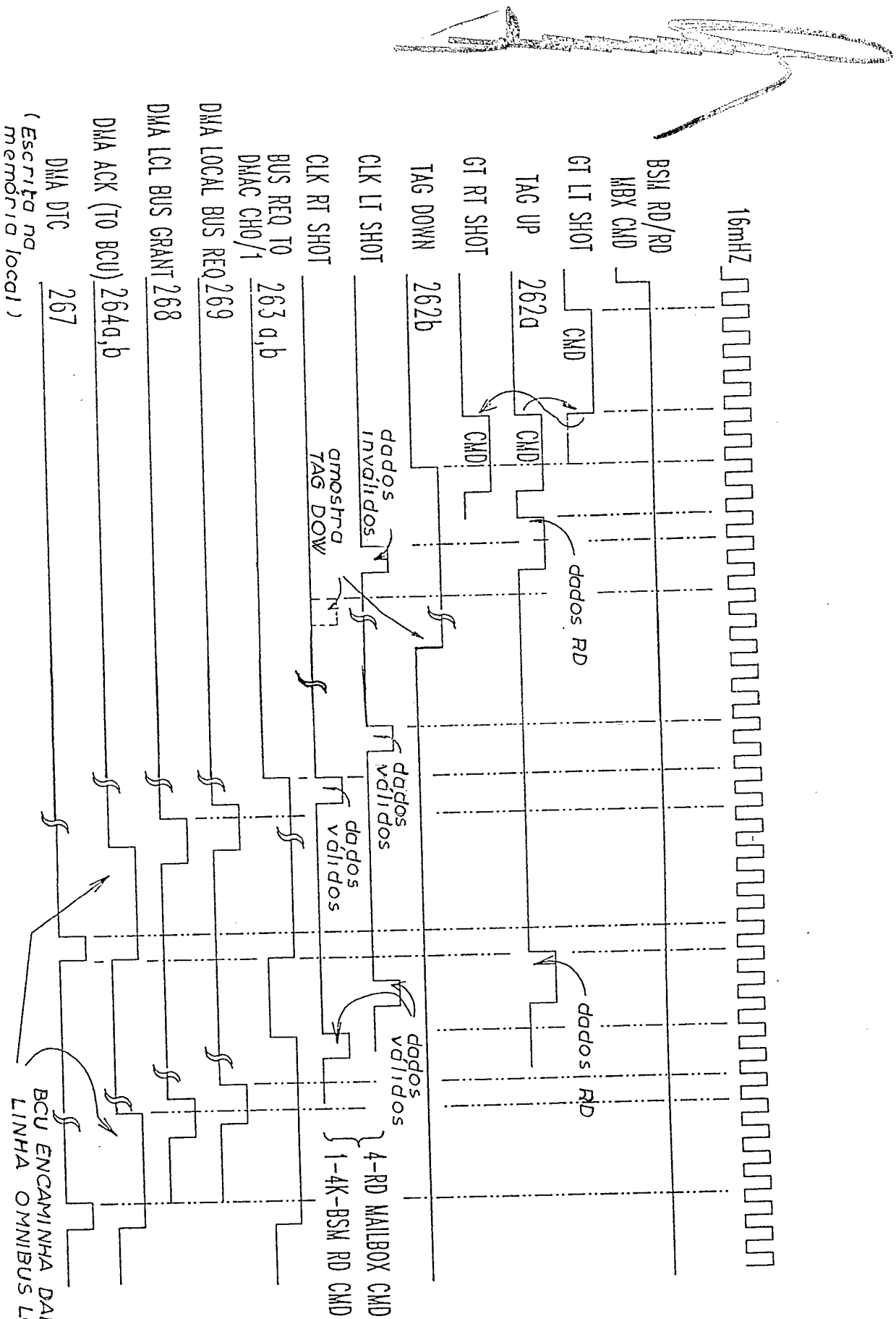


FIG. 25 — Leitura de MBX/BSM. Leitura de CMD

BCU ENCAMINHA DADOS PARA A LINHA OMNIBUS LOCAL

DMA ENCAMINHA CONTROLOS DE ENDEREÇOS

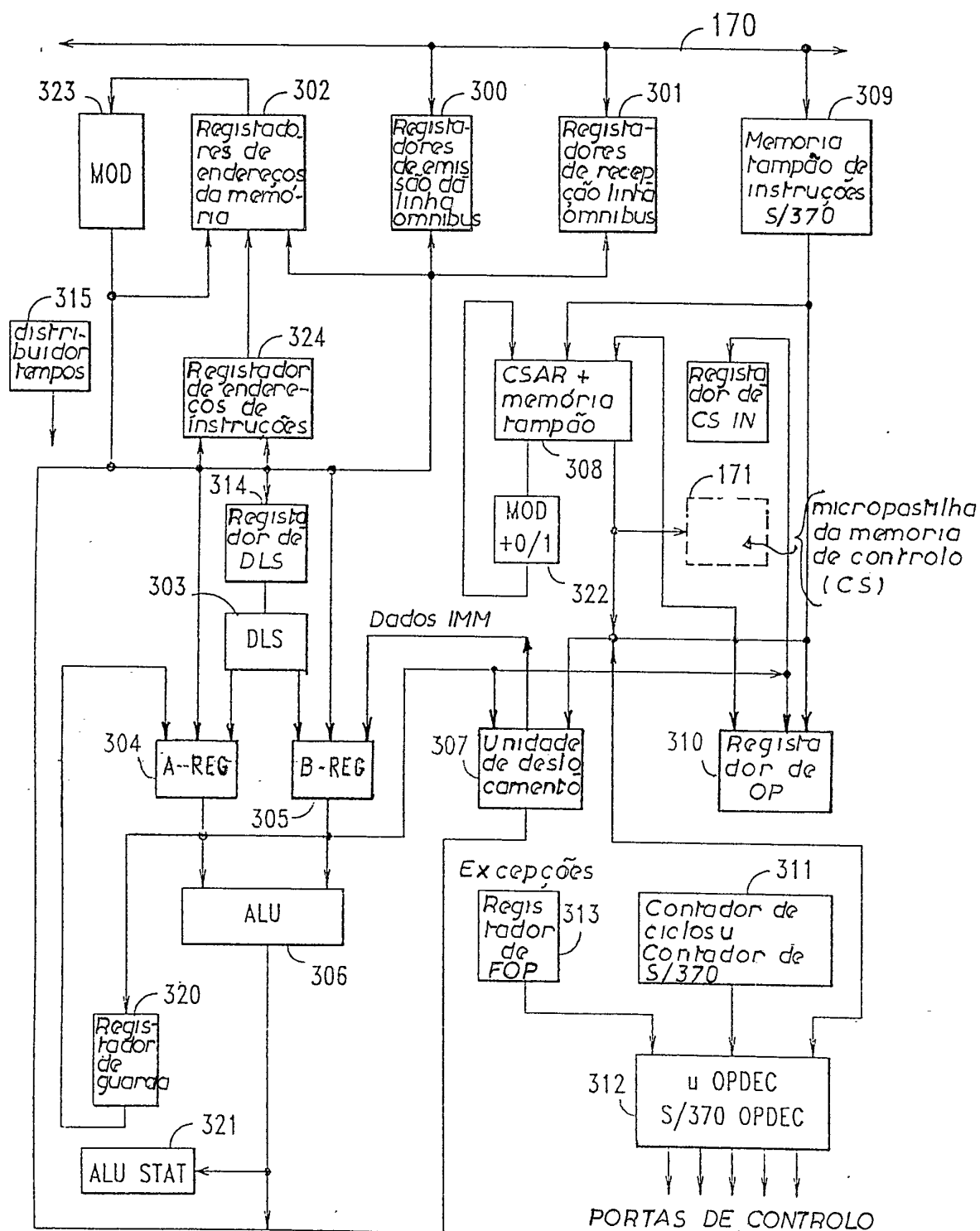


FIG. 27 - S/370 PROCESSADOR (85)

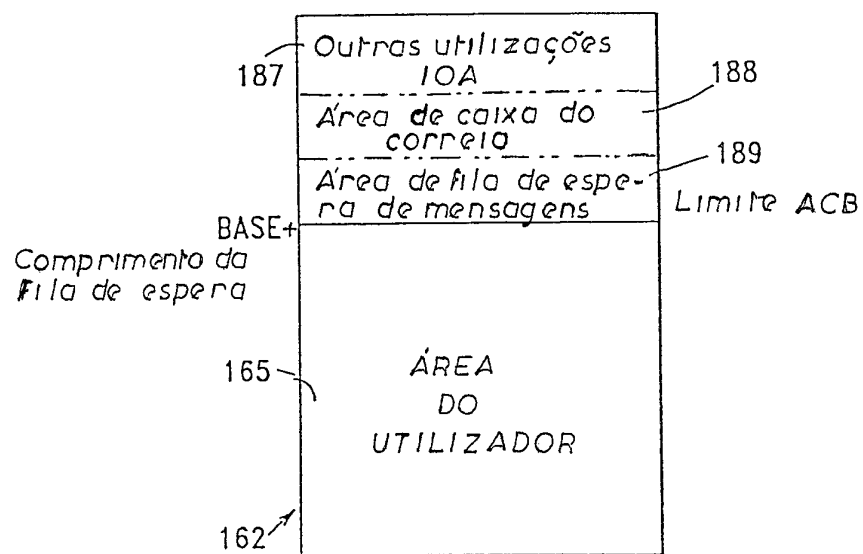


FIG. 28

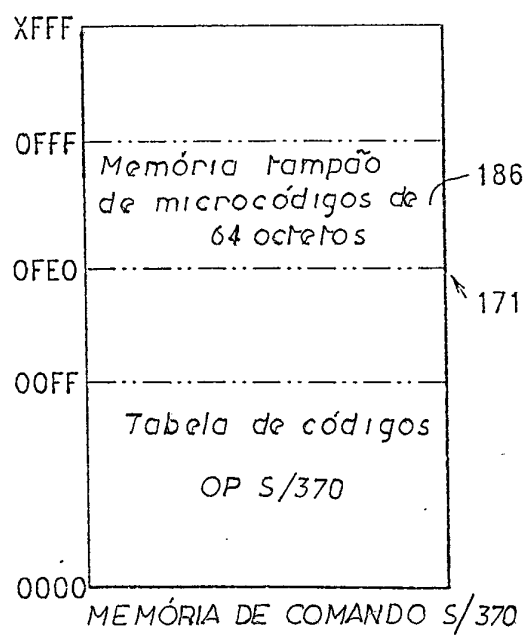


FIG. 29

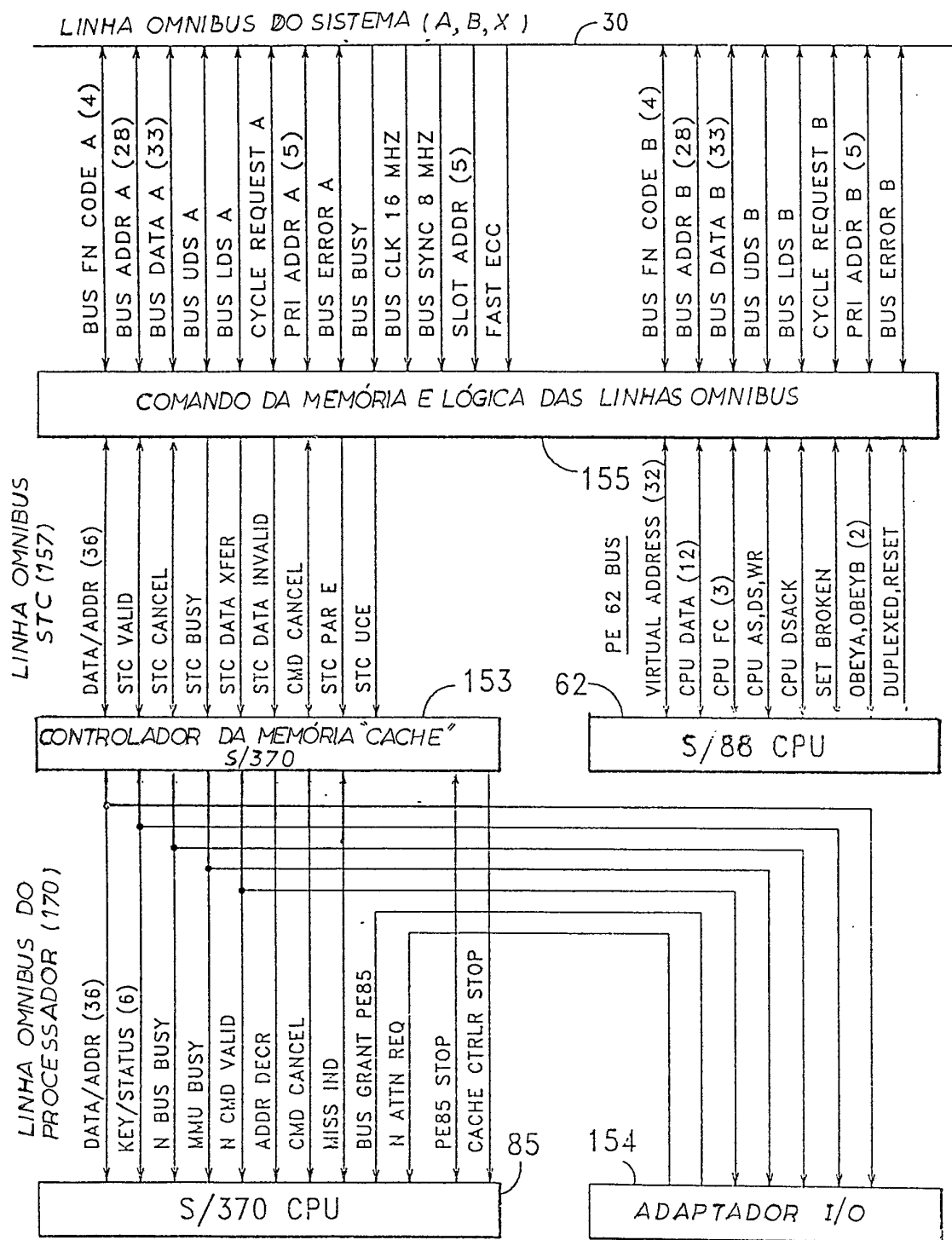


FIG. 30

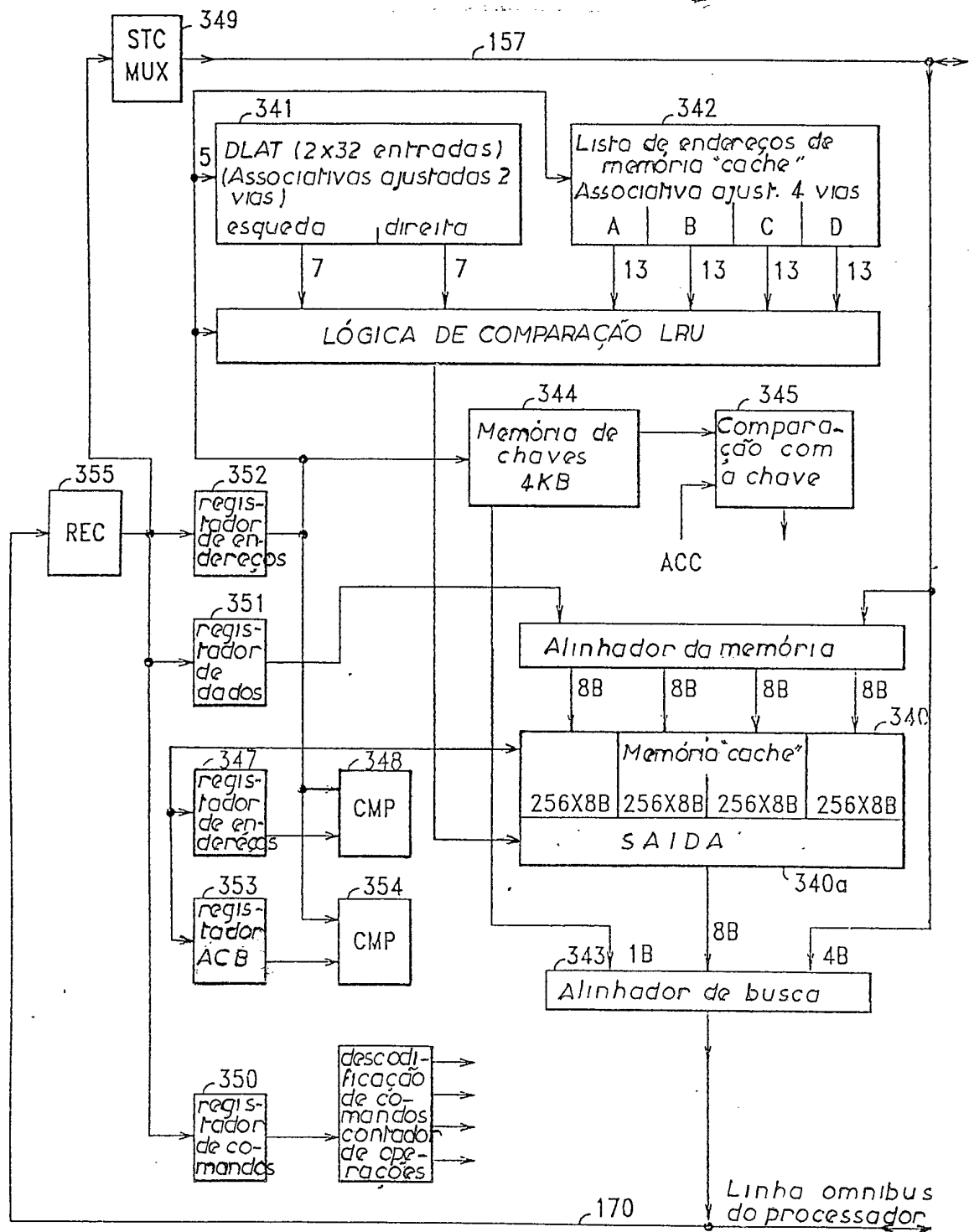


FIG. 31 - controlador da memória "cache" 153

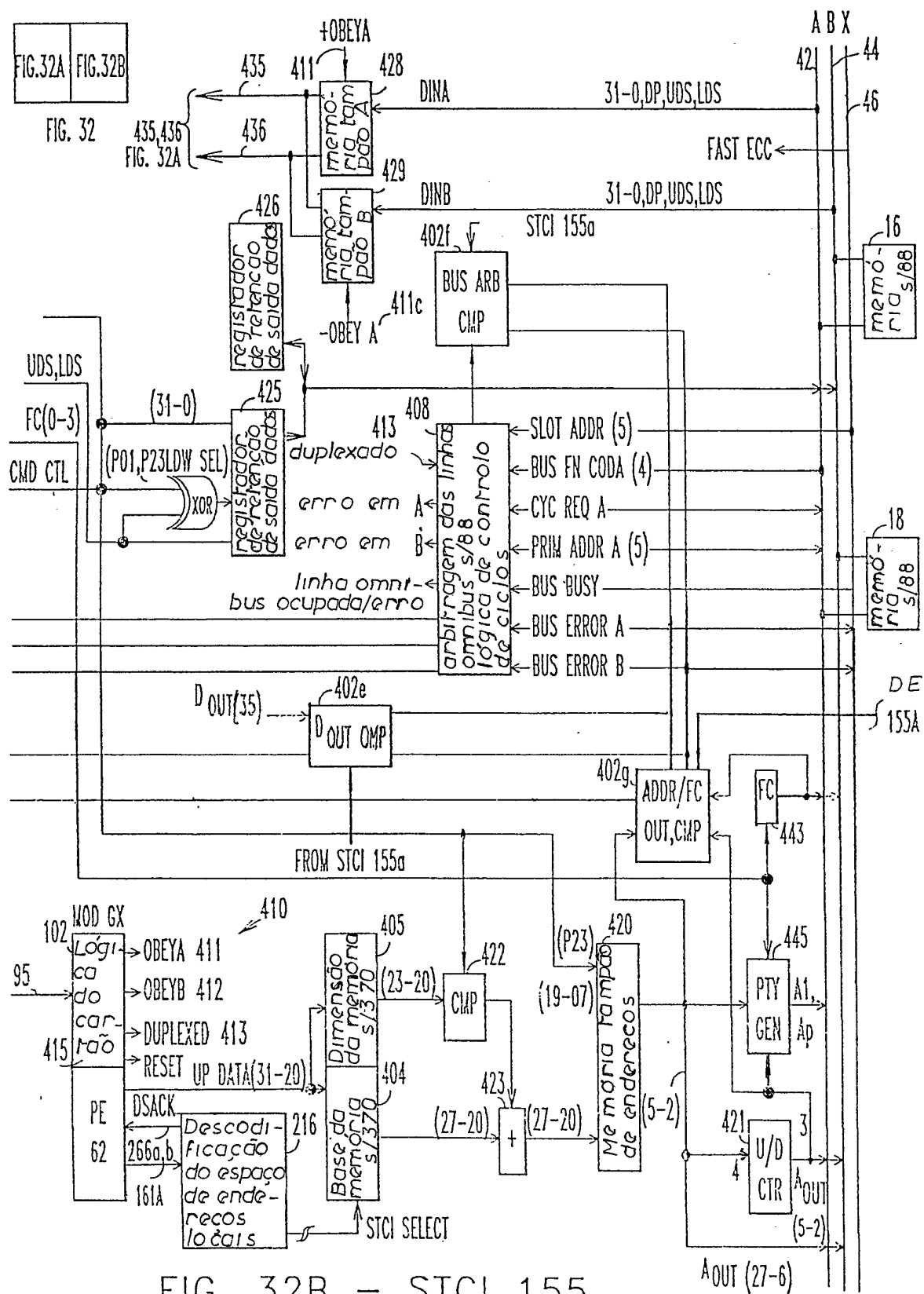


FIG. 32B - STCI 155

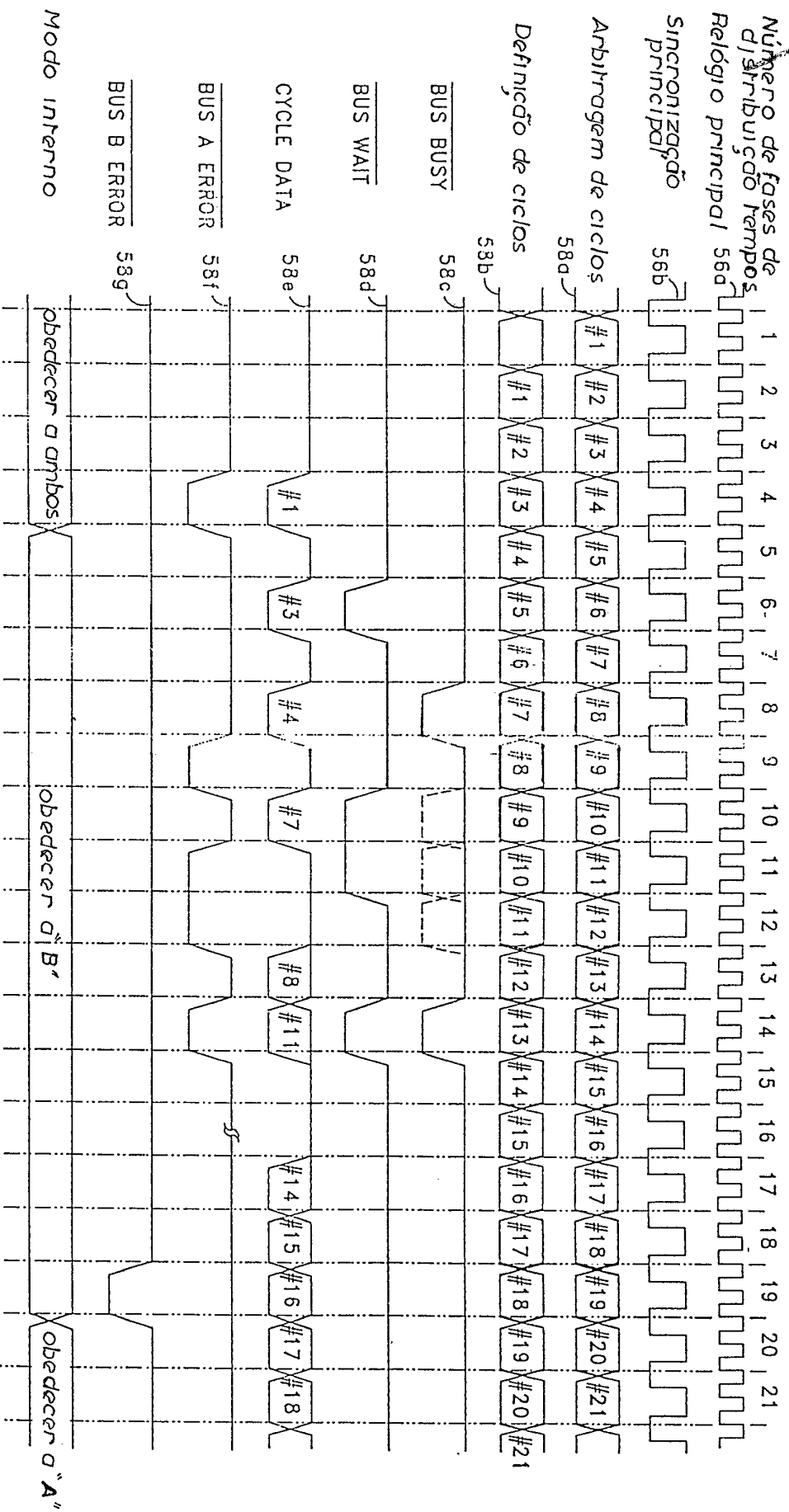


FIG. 33

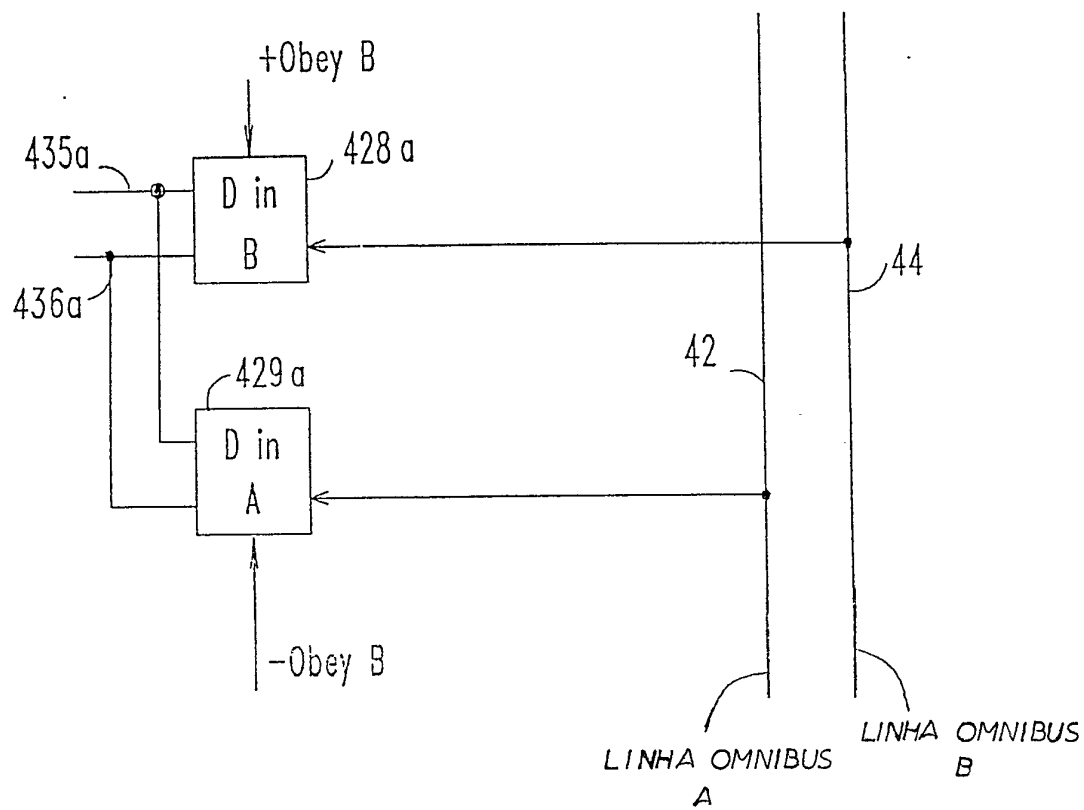
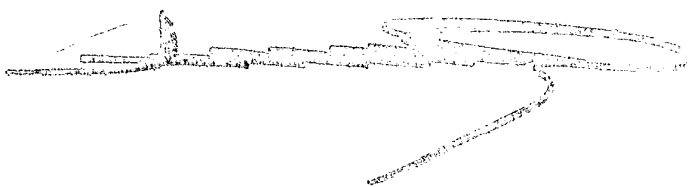


FIG. 34



PALAVRA DE ENDEREÇO DE
COMANDO

0	1	2	3	4	5	6	7	8
S	64B OVFL	32B	16B	8B	4B	TRL1	TRLO	C/A (=1)

PALAVRA DE DADO DE
MEMÓRIA

0 (Octeto de dados 0)	7	8
(FIFO EM 0 7)		C/A (=0)

0	7	8
Endereço real (FIFO em 8-15)		0

0	7	8
Octetos de dados 1 (FIFO em 8-15)		P _{0,1}

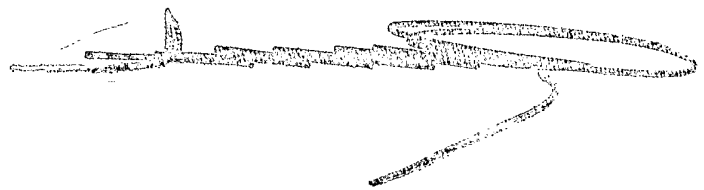
0	7	8
Endereço real (FIFO em 16 23)		0

0	7	8
Octeto de dados 2 (FIFO em 16 23)		LDW SEL

0	7	8
Endereço real (FIFO em 24 31)		P _{2,3}

0	7	8
Octeto de dados 3 (FIFO em 24 31)		P _{2,3}

FIG. 35 — FORMATOS DA MEMÓRIA FIFO



COMANDO	LINHA OMNIBUS STC				OCTETO(S) VÁLIDO(S)				TRANS- FERÊN- CIA ACTUAL	CICLO S/88 ACTUAL
	0	-	7	28-31	UU	UM	LM	LL		

PE85 FETCH 1011 1nnn

1 BYTE	1011	1000	xx00	V					BYTE	MEM32
1 BYTE	1011	1000	xx01		V				BYTE	MEM32
1 BYTE	1011	1000	xx10			V			BYTE	MEM32
1 BYTE	1011	1000	xx11				V		BYTE	MEM32
2 BYTE	1011	1001	xx00	V	V				WORD	MEM32
2 BYTE	1011	1001	xx01		V	V			2-BYTE	MEM32
2 BYTE	1011	1001	xx10			V	V		WORD	MEM32
3 BYTE	1011	1010	xx00	V	V	V			3-BYTE	MEM32
3 BYTE	1011	1010	xx01		V	V	V		3-BYTE	MEM32
4 BYTE	1011	1011	xx00	V	V	V	V		LW	MEM32
8 BYTE	1011	1100	x000	V	V	V	V		2-LW	MEM64
64 BYTE	1011	1101	xx00	V	V	V	V		(8)2-LW	MEM64
64 BYTE	1011	1110	xx00	V	V	V	V		(16) LW	MEM32

Lento

FIG. 36A — INTERFACE STC
(BUSCAS ORIGINADAS NO PE 85)

Comando	Linha omnibus STC			Octeto(s) válido(s)				Trans- ferência actual	Ciclo s/88 actual
	0	-	7 28-31	UU	UM	LM	LL		
I/O 1-64	01nn		nnnn						
<i>Busca de octetos no limite de palavras</i>									
1 BYTE	0100	0000	xxxx	V				BYTE	MEM32
1 BYTE	0100	0000	xxxx		V			BYTE	MEM32
1 BYTE	0100	0000	xxxx			V		BYTE	MEM32
1 BYTE	0100	0000	xxxx				V	BYTE	MEM32
2 BYTE	0100	0001	xxxx	V	V			WORD	MEM32
2 BYTE	0100	0001	xxxx		V	V		2-BYTE	MEM32
2 BYTE	0100	0001	xxxx			V	V	WORD	MEM32
3 BYTE	0100	0010	xxxx	V	V	V		3-BYTE	MEM32
3 BYTE	0100	0010	xxxx		V	V	V	3-BYTE	MEM32
4 BYTE	0100	0011	xxxx	V	V	V	V	LW	MEM32
<i>Para cruzamento de limite de palavras</i>									
<i>parte inicial</i>									
3 BYTE	01nn	nnnn	xx01		V	V	V	3-BYTE	MEM32
2 BYTE	01nn	nnnn	xx10			V	V	WORD	MEM32
1 BYTE	01nn	nnnn	xx11				V	BYTE	MEM32
<i>parte média</i>									
8 BYTE	01nn	nnnn	xxxx	V	V	V	V	2-LW	MEM64
<i>parte final</i>									
1 BYTE	01nn	nnnn	xxxx	V				BYTE	MEM32
1 BYTE	01nn	nnnn	xxxx		V			BYTE	MEM32
1 BYTE	01nn	nnnn	xxxx			V		BYTE	MEM32
1 BYTE	01nn	nnnn	xxxx				V	BYTE	MEM32
2 BYTE	01nn	nnnn	xxxx	V	V			BYTE	MEM32
2 BYTE	01nn	nnnn	xxxx		V	V		BYTE	MEM32
2 BYTE	01nn	nnnn	xxxx			V	V	BYTE	MEM32
3 BYTE	01nn	nnnn	xxxx	V	V	V		3-BYTE	MEM32
3 BYTE	01nn	nnnn	xxxx		V	V	V	3-BYTE	MEM32
4 BYTE	01nn	nnnn	xxxx	V	V	V	V	LW	MEM32

FIG. 36B — INTERFACE STC
(BUSCAS ORIGINADAS NO ADAPTADOR 154)



comando	Linha omnibus STC				octeto(s) válido(s)				Trans-ferência actual	Ciclos 88 actual	UDS LDS	
	0	-	7	28-31	UU	UW	LM	LL				

PE85 MEMÓRIA 1001 1nnn

1 BYTE	1001	1000	xx00	V					BYTE	MEM16	1	0
1 BYTE	1001	1000	xx01		V				BYTE	MEM16	0	1
1 BYTE	1001	1000	xx10	LM		V			BYTE	MEM16	1	0
1 BYTE	1001	1000	xx11		LL		V		BYTE	MEM16	0	1
2 BYTE	1001	1001	xx00	V	V				WORD	MEM16	1	1
2 BYTE	1001	1001	xx01		V	V			2-BYTE	MEM32	0	0
2 BYTE	1001	1001	xx10	LM	LL	V	V		WORD	MEM16	1	1
3 BYTE	1001	1010	xx00	V	V	V			3-BYTE	MEM32	1	0
3 BYTE	1001	1010	xx01		V	V	V		3-BYTE	MEM32	0	1
4 BYTE	1001	1011	xx00	V	V	V	V		LW	MEM32	1	1
8 BYTE	1001	1100	xx00	V	V	V	V		2-LW	MEM32	1	1
64 BYTE	1001	1101	xx00	V	V	V	V		(16) LW	MEM32	1	1

Os octetos válidos multiplexados para octetos superiores estão indicados por "LM" ou "LL"

INTERFACE STC
(ARMAZENAMENTOS ORIGINADOS PELO PE 85)

FIG. 36C



comando	Linha omnibus STC			Octeto(s) válido(s)				Trans- ferência actual	Ciclo s/88		
	0	-	7 28-31	UU	UM	LM	LL		actual	UDS	LDS

I/O 1-64 00nn nnnn

Armazenamento
de octetos

NO LIMETE DE PALAVRAS

1 BYTE	0000	0000	xx00	V				BYTE	MEM16	1	0
1 BYTE	0000	0000	xx01		V			BYTE	MEM16	0	1
1 BYTE	0000	0000	xx10	LM		V		BYTE	MEM16	1	0
1 BYTE	0000	0000	xx11		LL		V	BYTE	MEM16	0	1
2 BYTE	0000	0001	xx00	V	V			WORD	MEM16	1	1
2 BYTE	0000	0001	xx01		V	V		2-BYTE	MEM32	0	0
2 BYTE	0000	0001	xx10	LM	LL	V	V	WORD	MEM16	1	1
3 BYTE	0000	0010	xx00	V	V	V		3-BYTE	MEM32	1	0
3 BYTE	0000	0010	xx01		V	V	V	3-BYTE	MEM32	0	1
4 BYTE	0000	0011	xx00	V	V	V	V	LW	MEM32	1	1

PARA CRUZAMENTO DE LIMITES DE PALAVRAS

Parte inicial

3 BYTE	00nn	nnnn	xx01		V	V	V	3-BYTE	MEM32	0	1
2 BYTE	00nn	nnnn	xx10	LM	LL	V	V	WORD	MEM16	1	1
1 BYTE	00nn	nnnn	xx11		LL		V	BYTE	MEM16	0	1

Parte média

8 BYTE	00nn	nnnn	xxxx	V	V	V	V	2-LW	MEM64	1	1

Parte final

STC (6,7)

+ (30,31)

=00	00nn	nnnn	xxxx	V				BYTE	MEM16	1	0
=01	00nn	nnnn	xxxx	V	V			WORD	MEM16	1	1
=10	00nn	nnnn	xxxx	V	V	V		3-BYTE	MEM32	1	0
=11	00nn	nnnn	xxxx	V	V	V	V	LW	MEM32	1	1

FIG. 36D — INTERFACE STC

(ARMAZENAMENTOS ORIGINADOS
PELO ADAPTADOR 154)

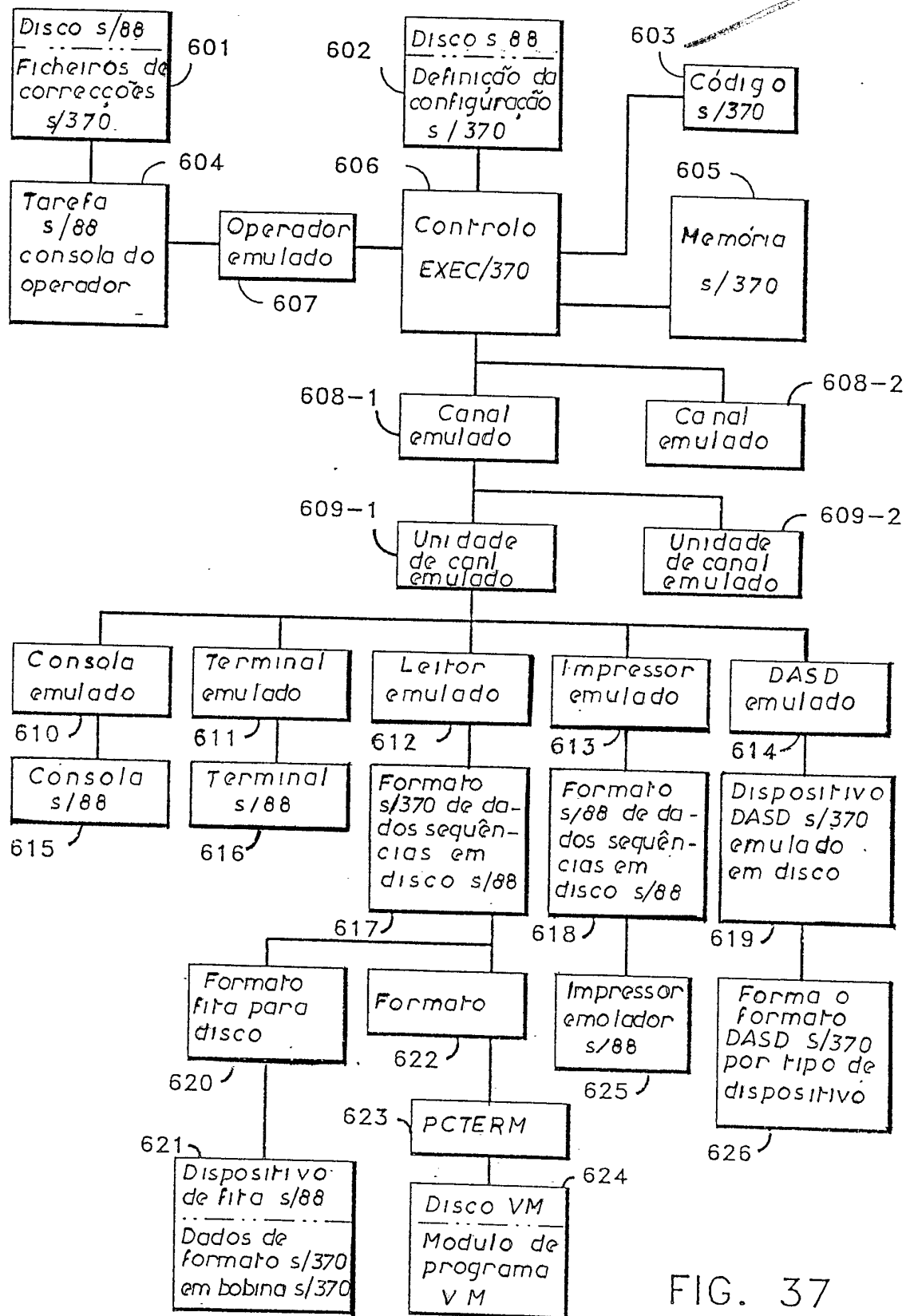


FIG. 37

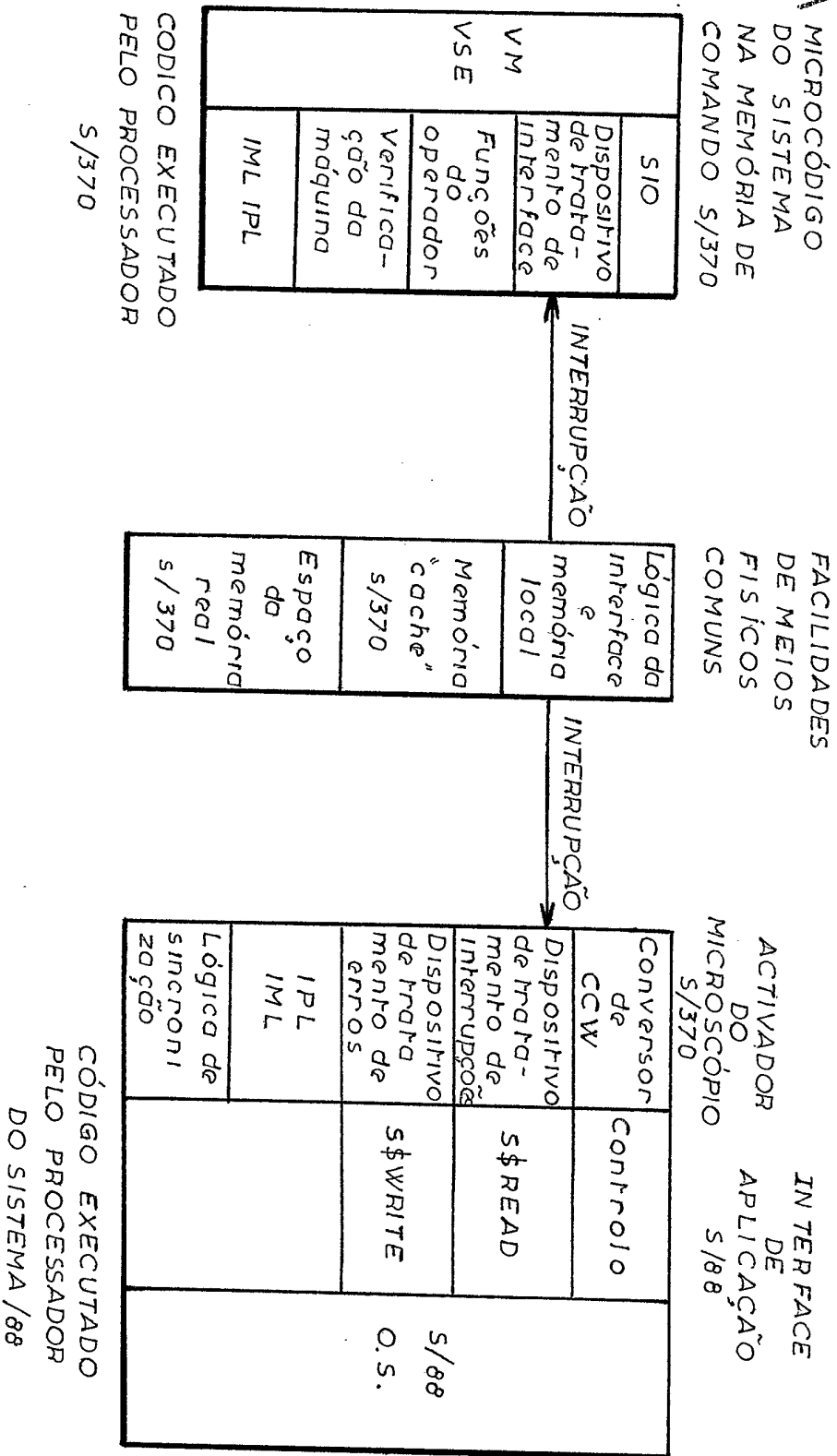


FIG. 38

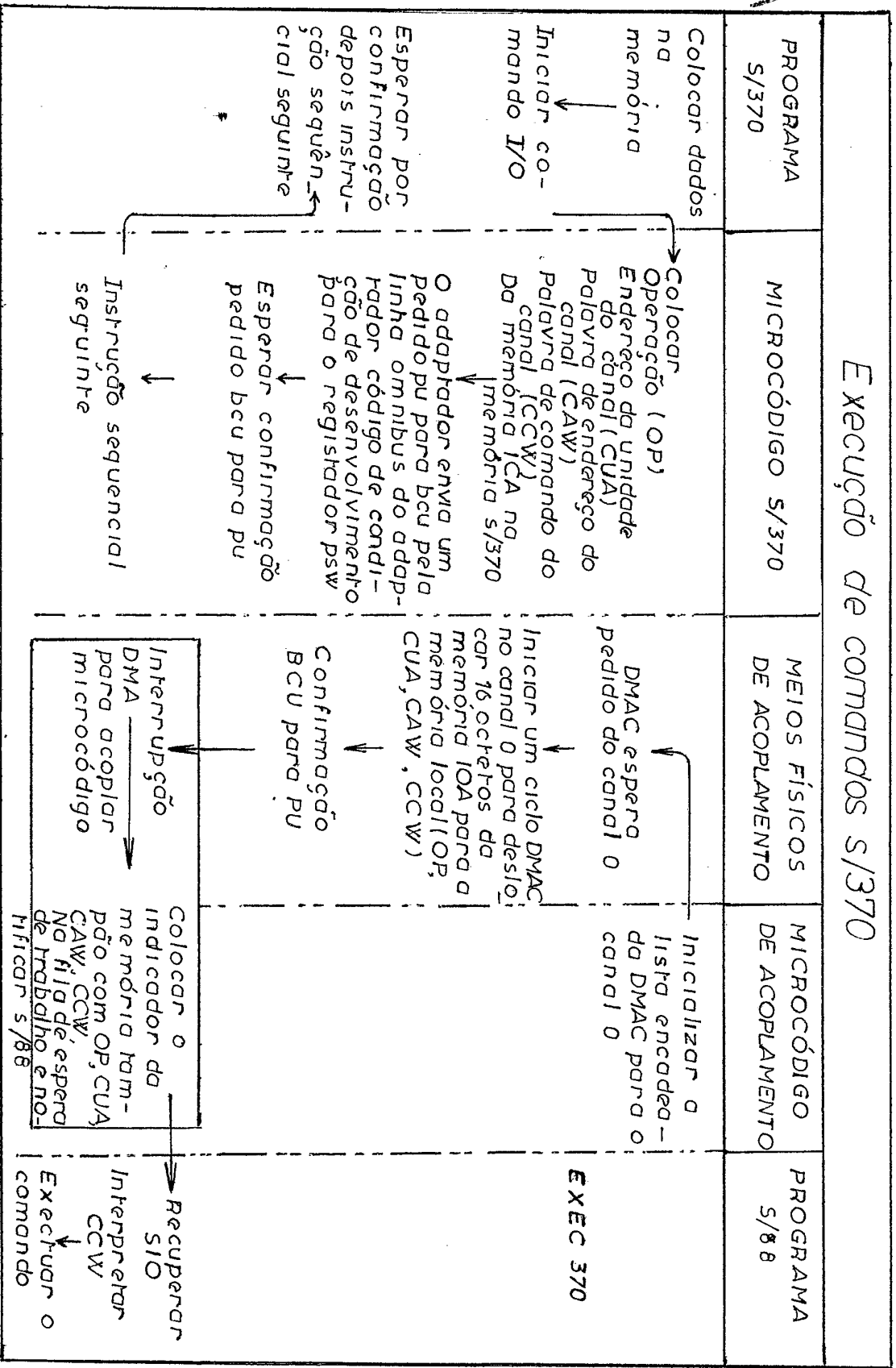
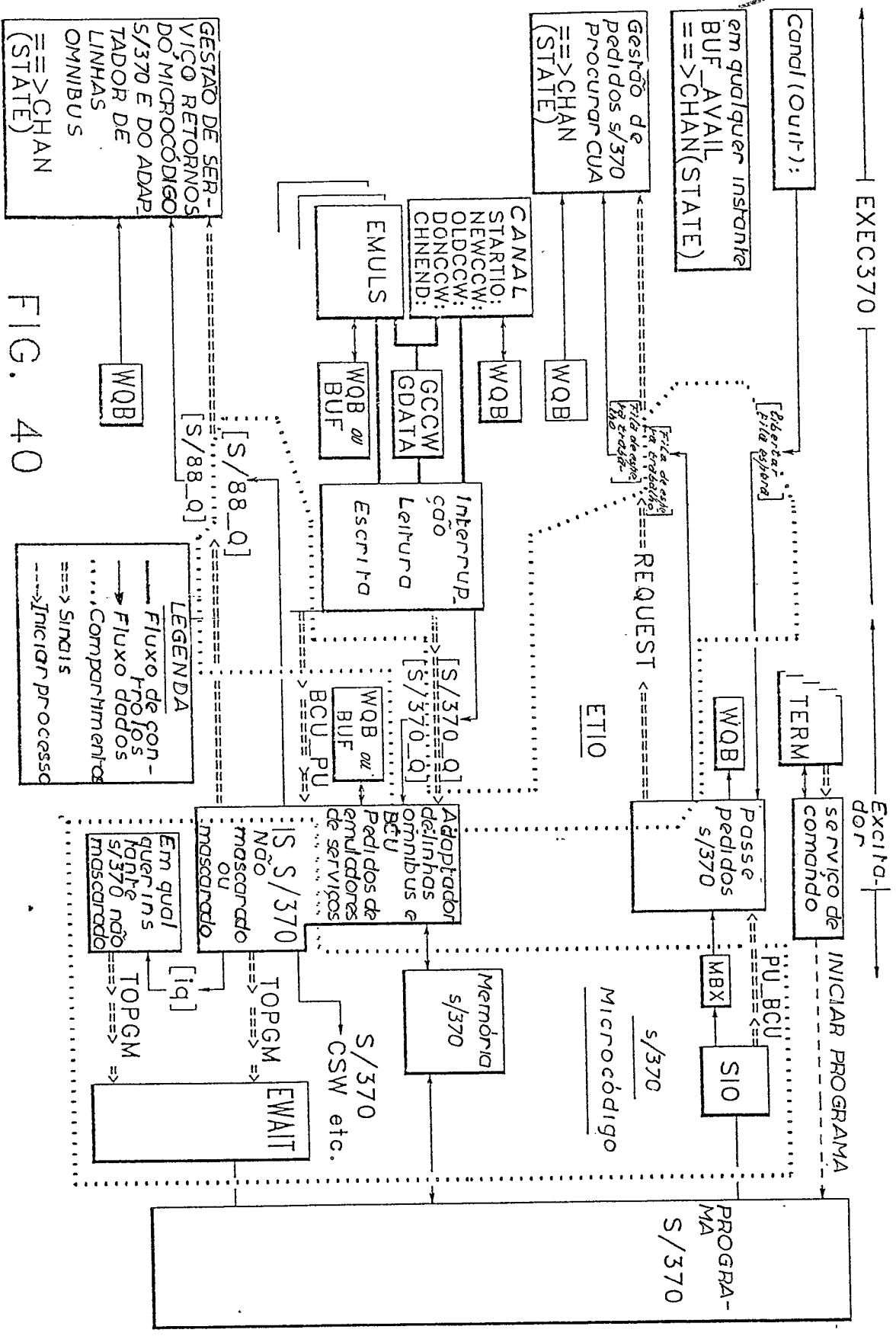


FIG. 39



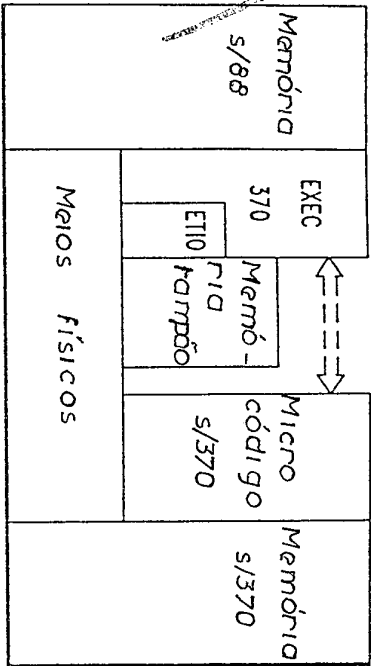


FIG. 41A

FIG. 41B

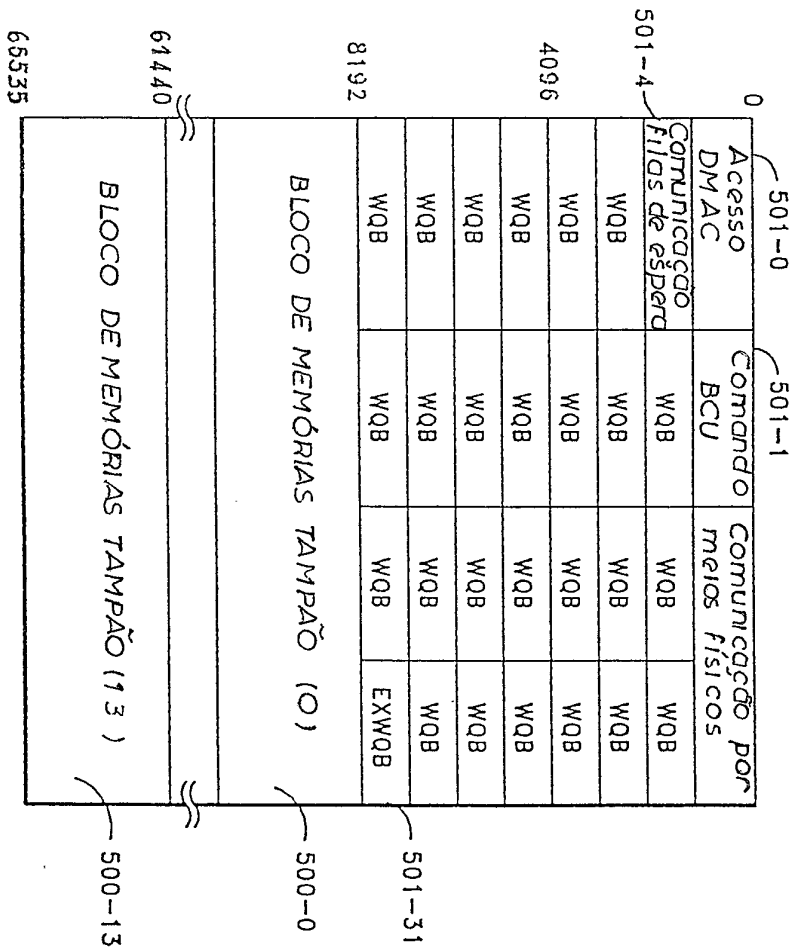
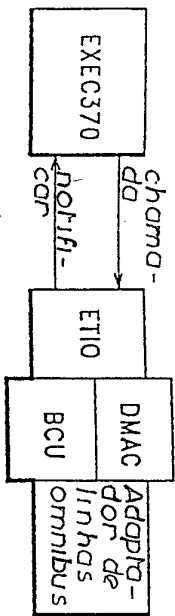


FIG. 41C

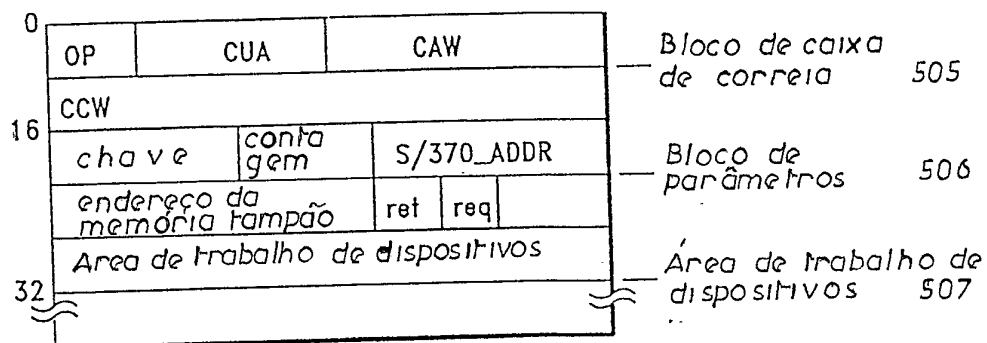


FIG. 41D — WQB BLOCO DA FILA DE ESPERA DE TRABALHO

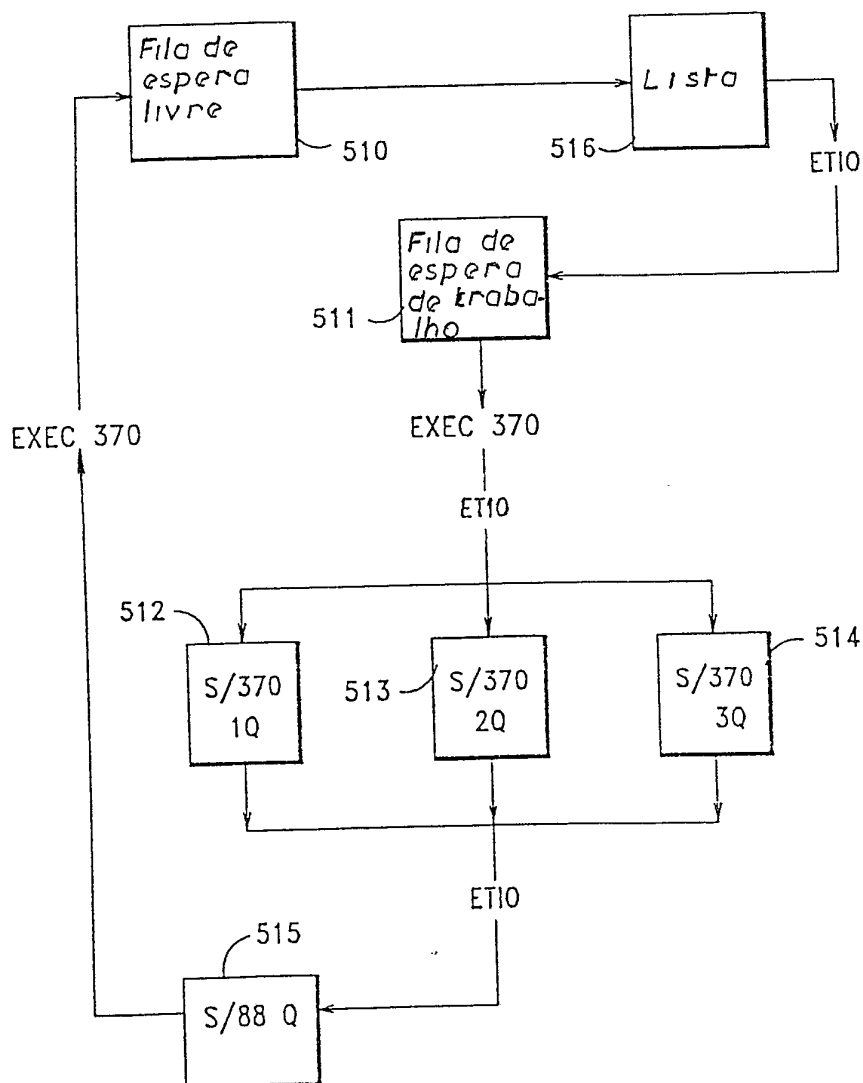


FIG. 41E — PROJECTO DO BLOCO DA FILA DE ESPERA DE TRABALHO (WOB) ATRAVÉS DAS FILAS DE ESPERA

Desvio

0	EXBUSY			501-4
4	EX S/370 EVENT ID			
8	FREE Q	40b	(32 Entradas)	
48	WORK Q	40b	(32 Entradas)	
88	CAP1Q	40b	(32 Entradas)	
128	CAP2Q	40b	(32 Entradas)	
168	CAP3Q	40b	(32 Entradas)	
208	S/88 Q	40b	(32 Entradas)	
248	Reservado	8b	255	

FIG. 41F

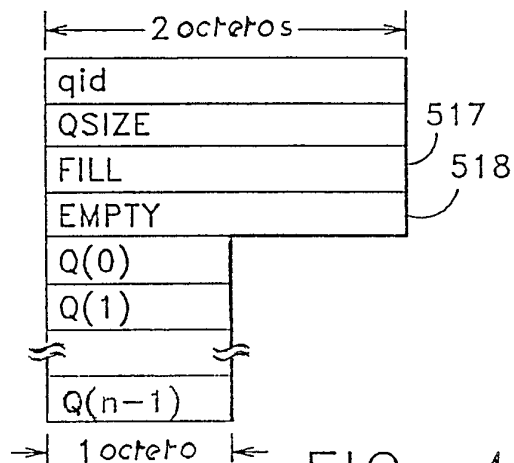


FIG. 41G

Lista de elos de ligação

516

elo de ligacao seguinte	endereço WQB	contagem

FIG. 41H

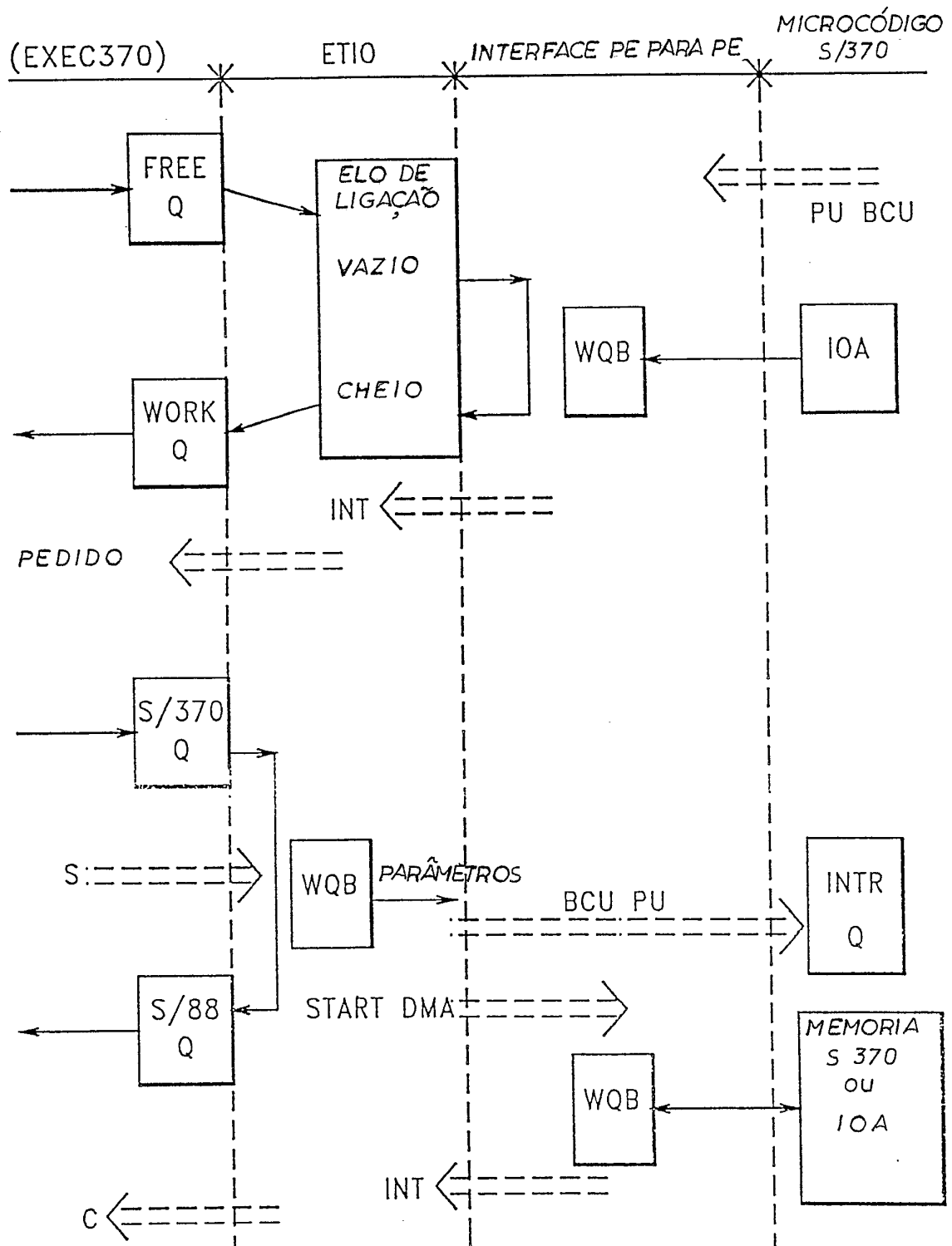


FIG. 42

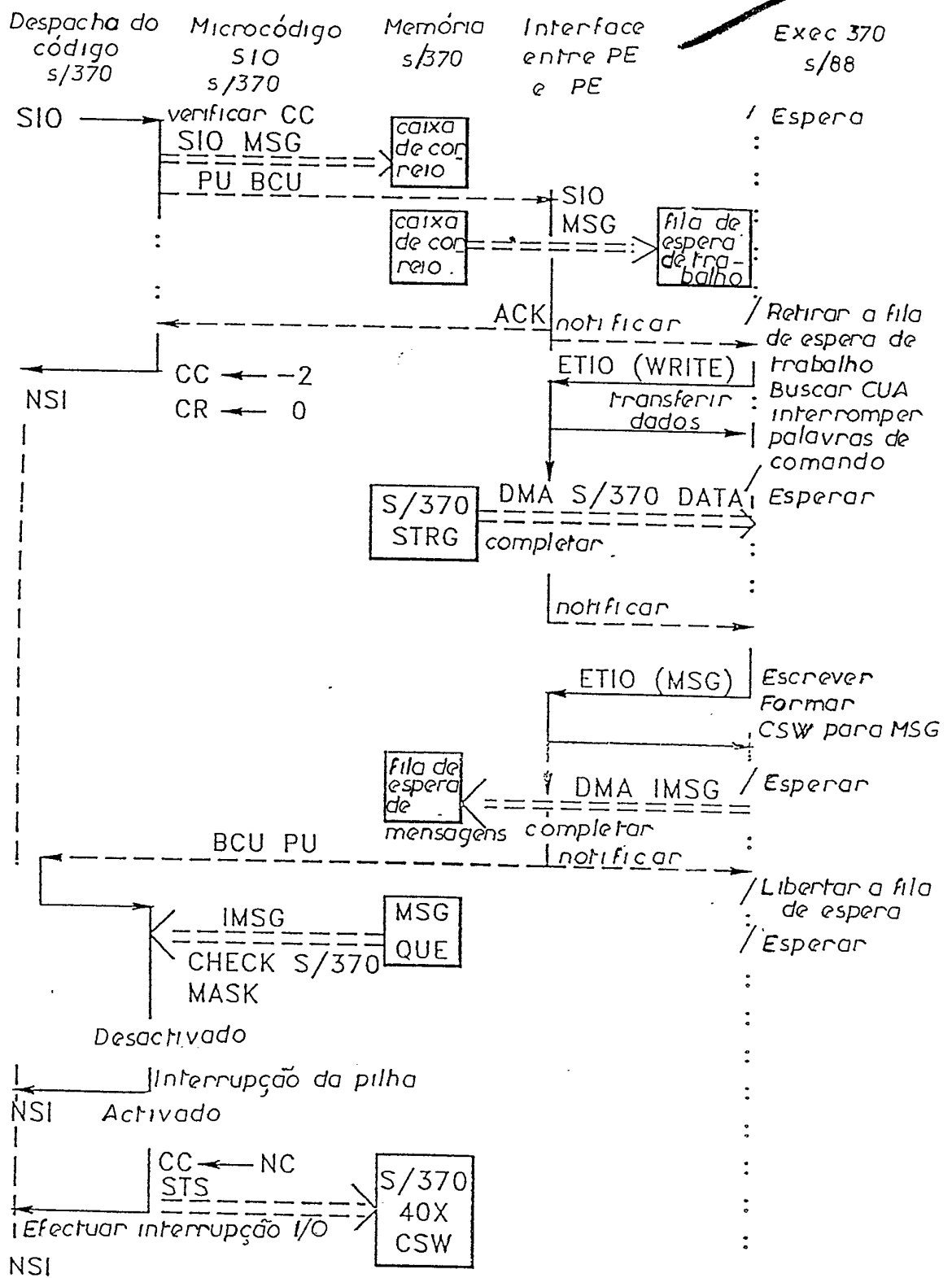
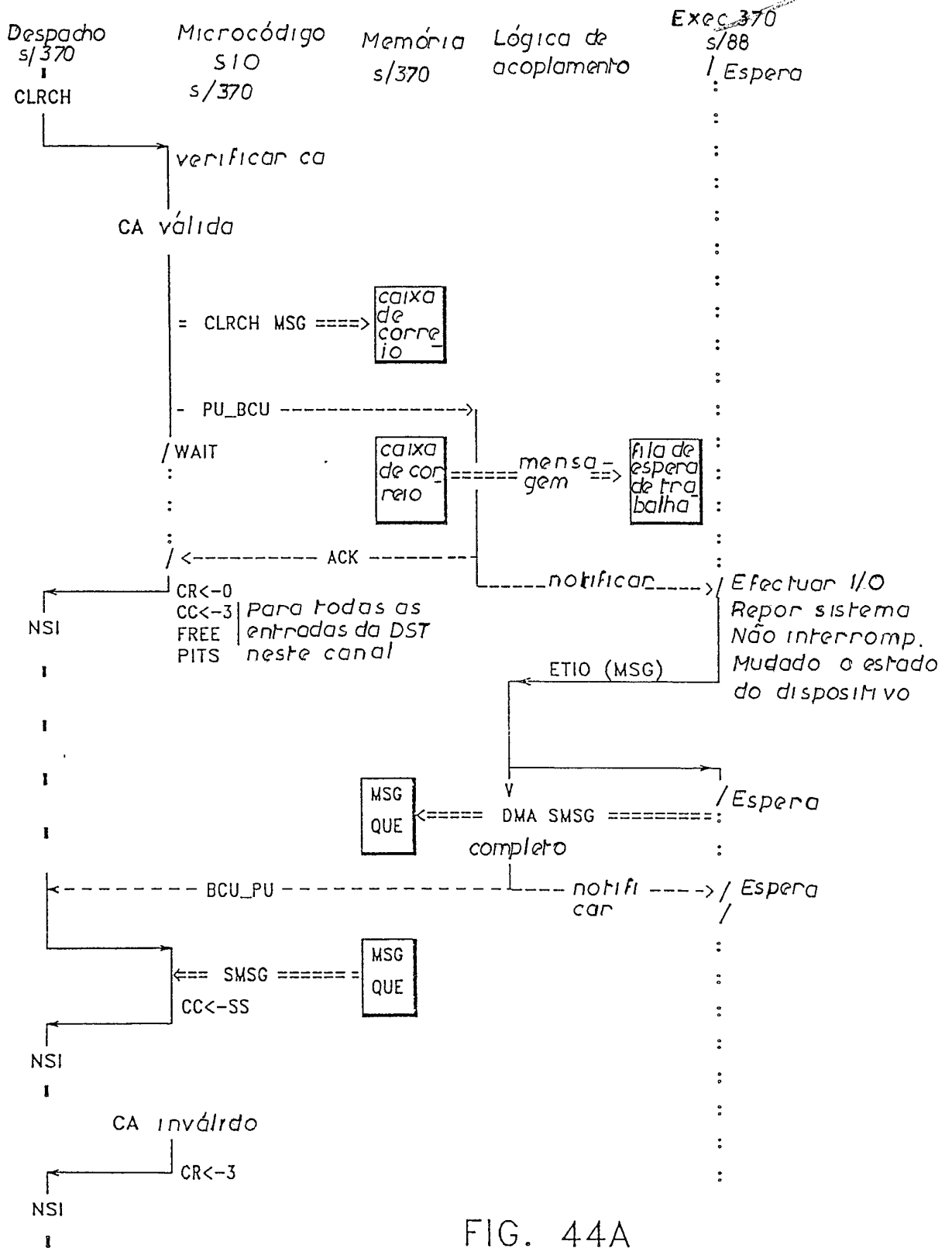


FIG. 43



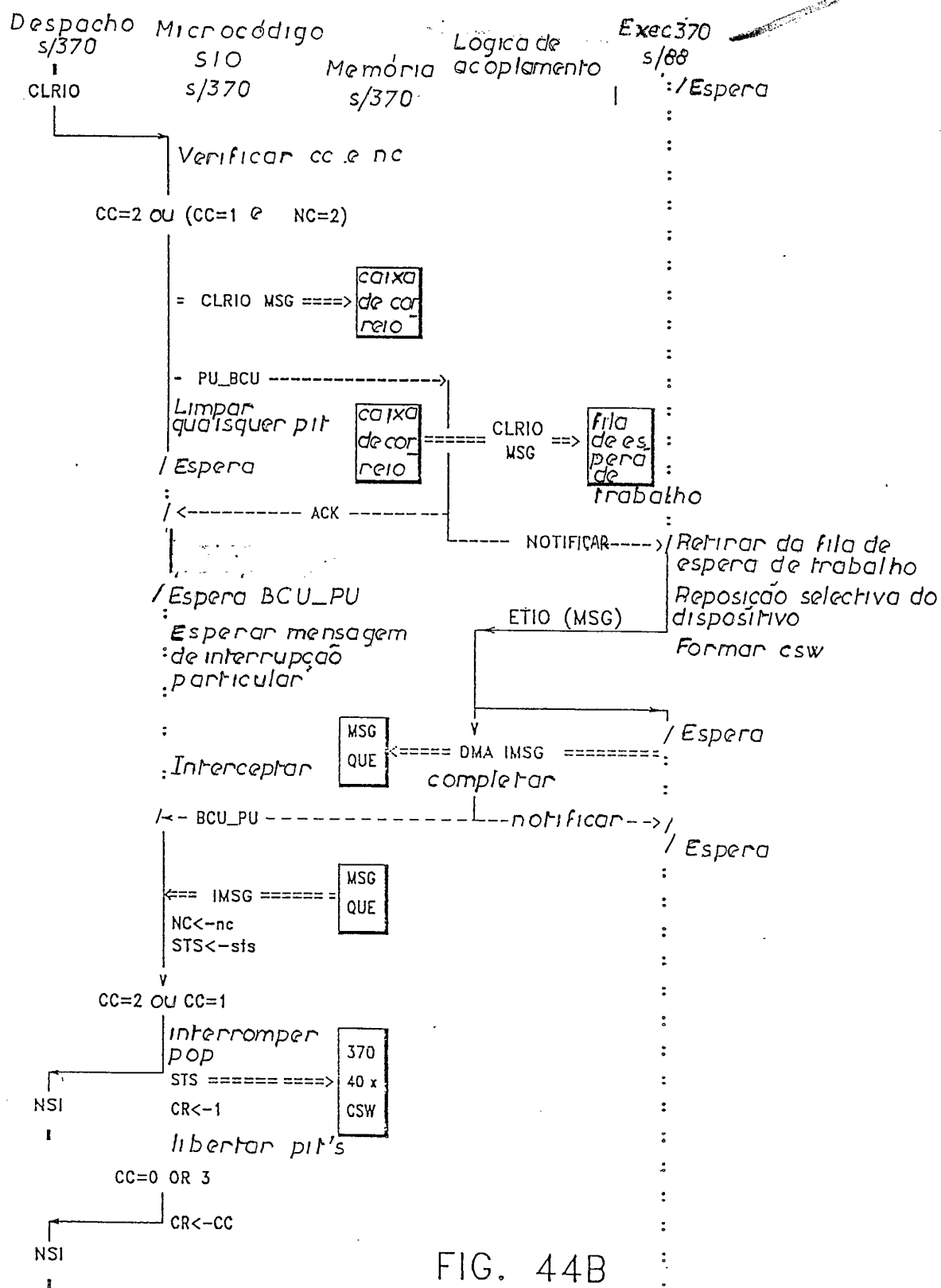


FIG. 44B

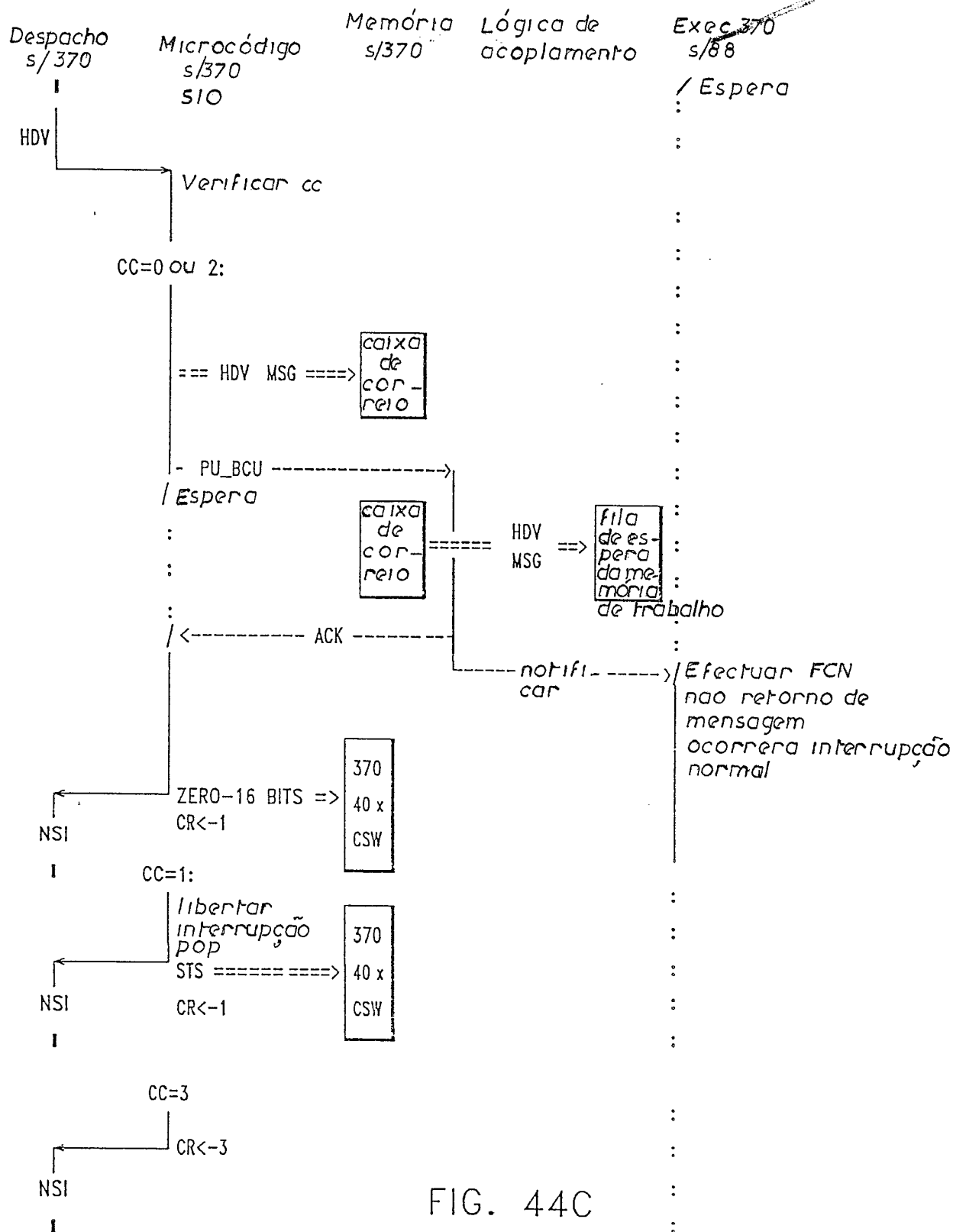


FIG. 44C

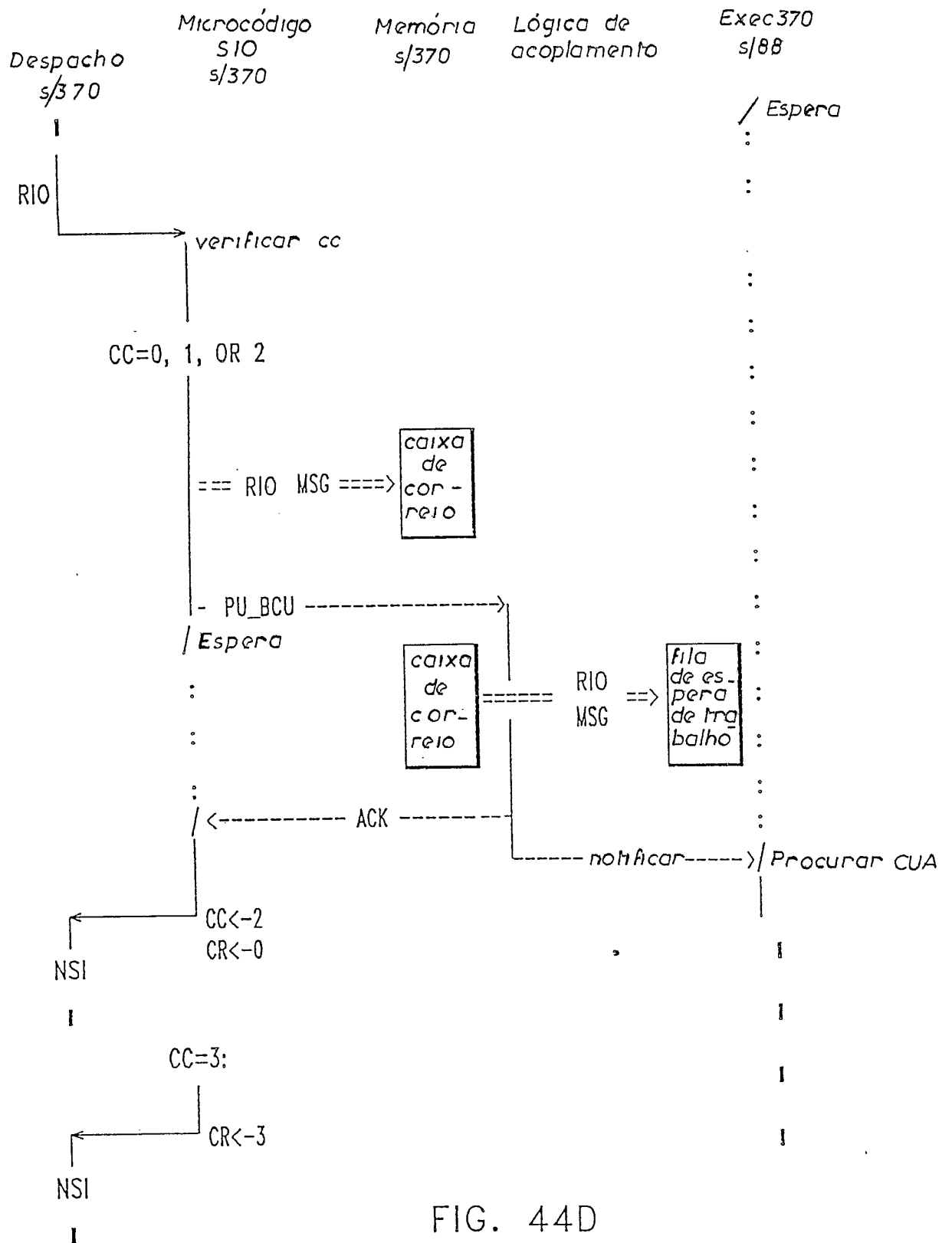


FIG. 44D

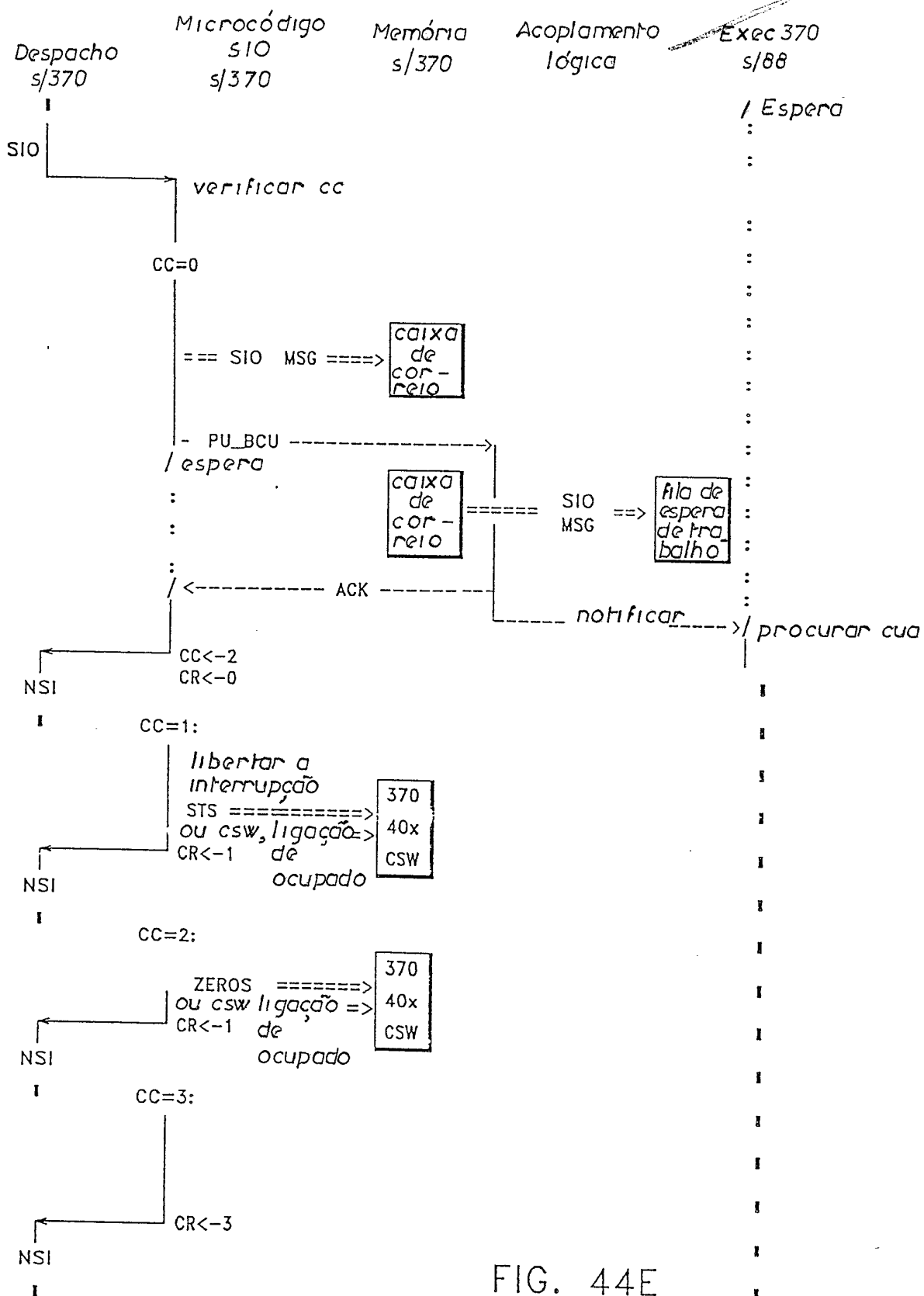


FIG. 44E

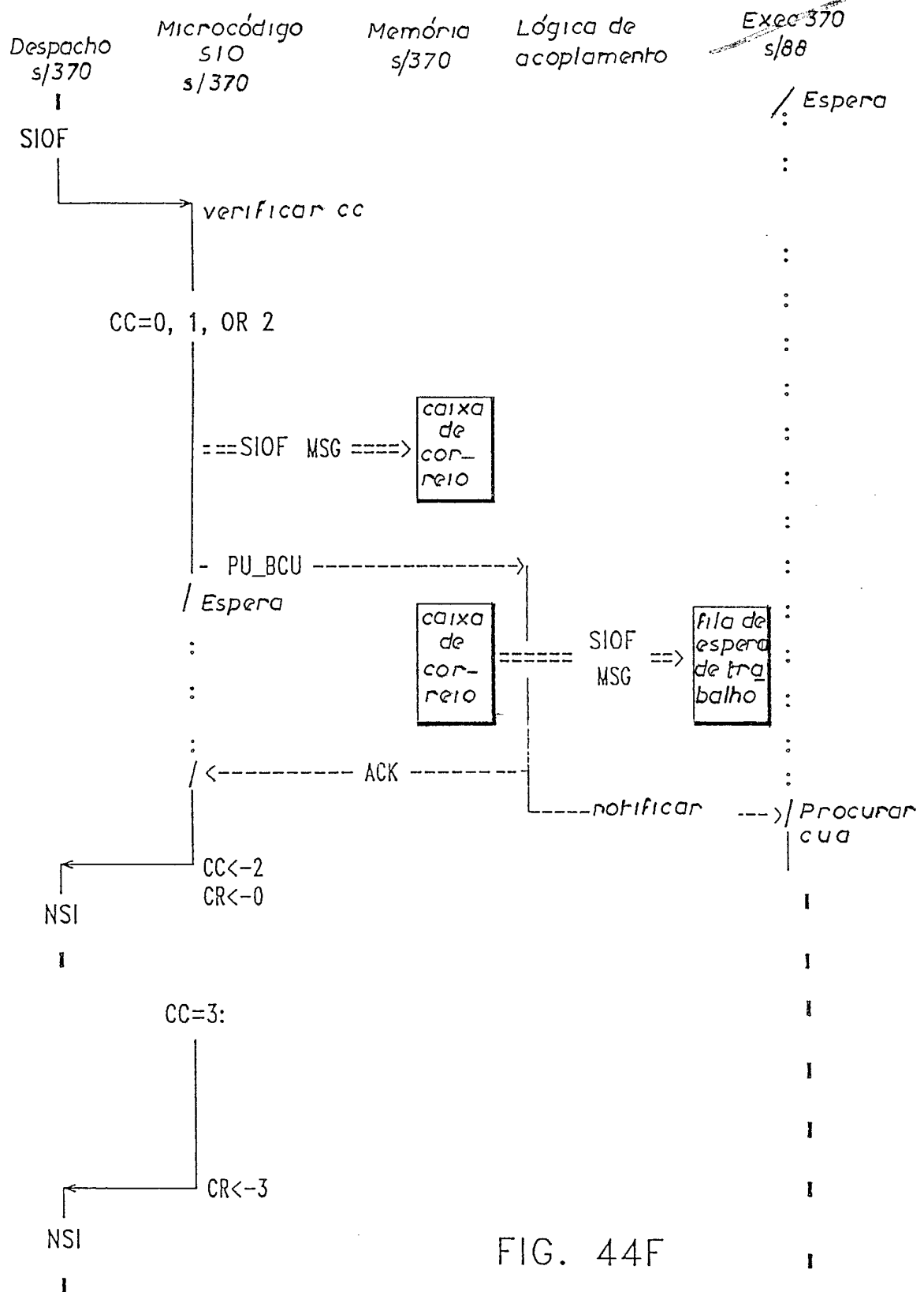


FIG. 44F

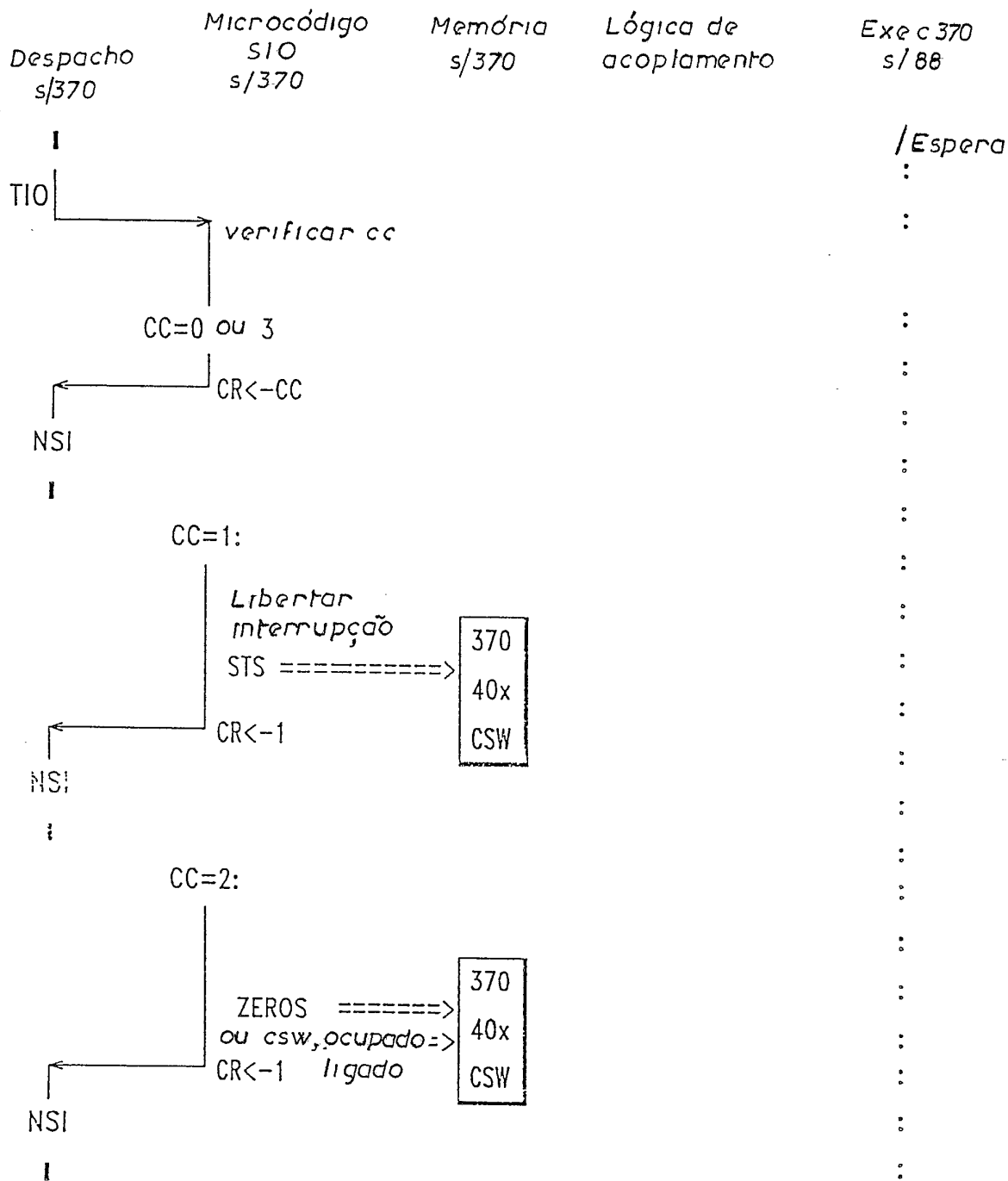
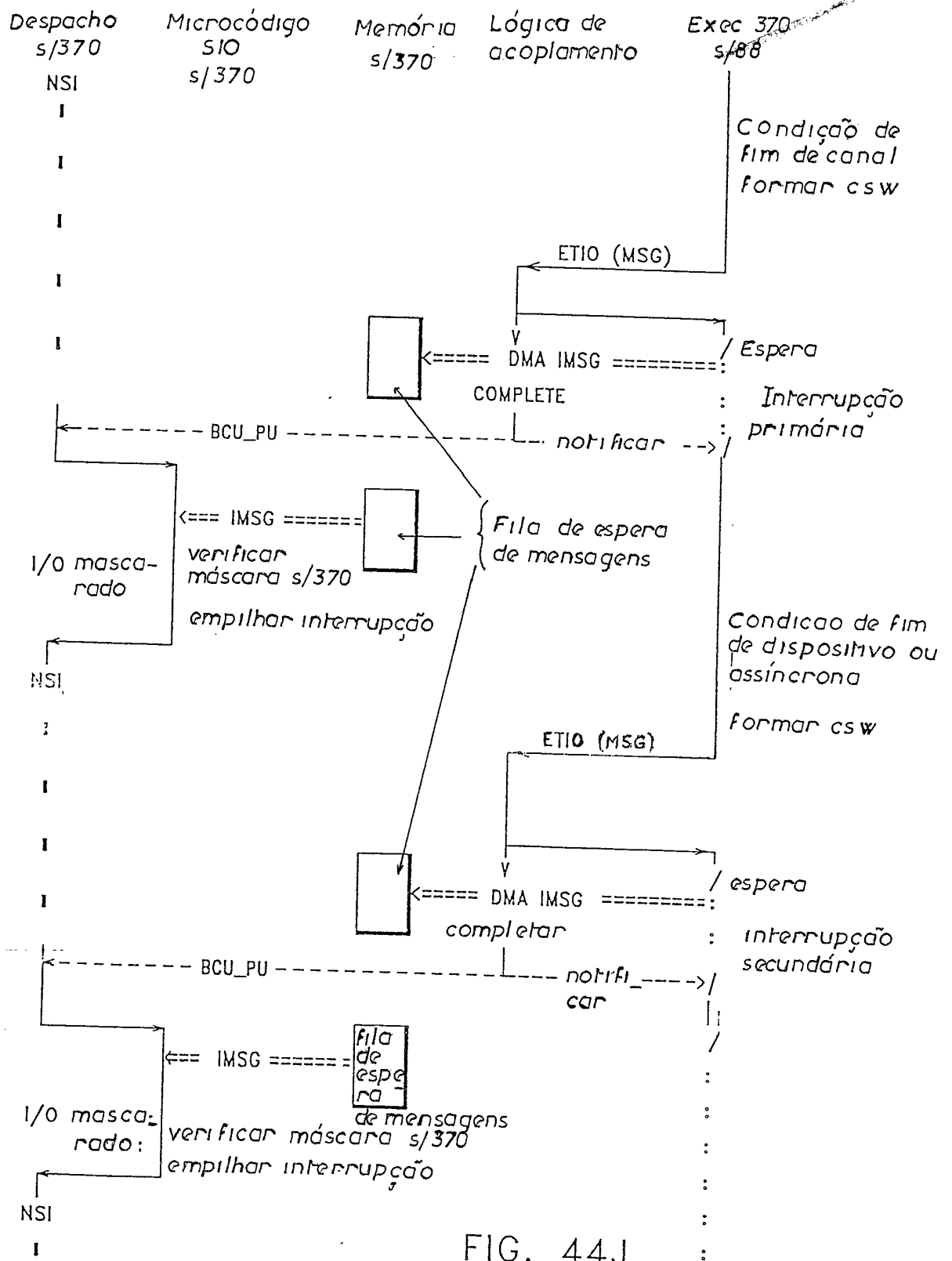


FIG. 44G



COMANDO/ESTADO :

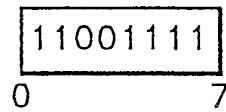


FIG. 45A

LINHA OMNIBUS DE ENDEREÇOS/DADOS

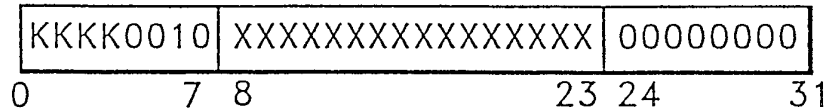


FIG. 45B

LINHA OMNIBUS DE ENDEREÇOS

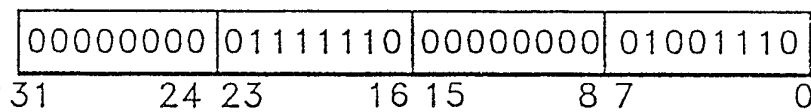


FIG. 45C

LINHA OMNIBUS DE DADOS

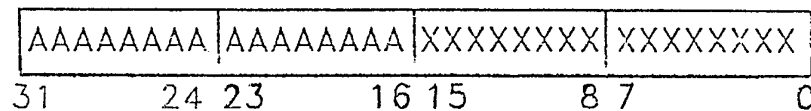


FIG. 45D

LINHA OMNIBUS DE ENDEREÇO

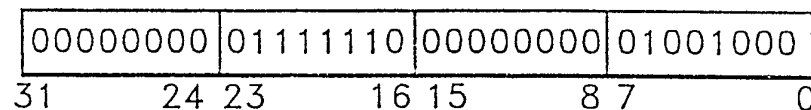


FIG. 45E

LINHA OMNIBUS DE DADOS

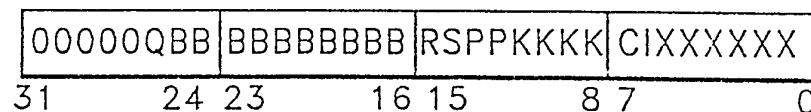


FIG. 45F

LINHA OMNIBUS DE ENDEREÇOS

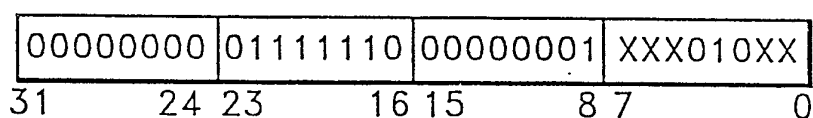


FIG. 45G

LINHA OMNIBUS DE DADOS

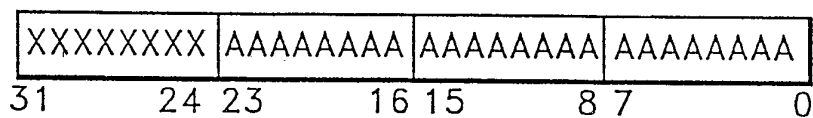


FIG. 45H

COMANDO / DADOS

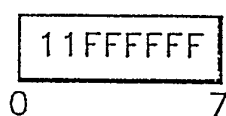


FIG. 45I

LINHA OMNIBUS DE ENDEREÇOS/DADOS

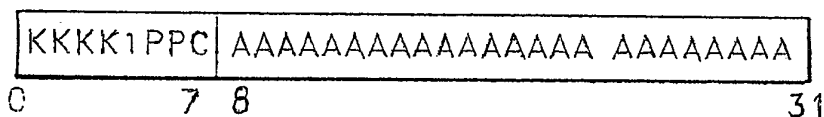


FIG. 45J

LINHA OMNIBUS DO PROCESSADOR



FIG. 45K

LINHA OMNIBUS CHAVE/ESTADO

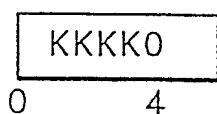
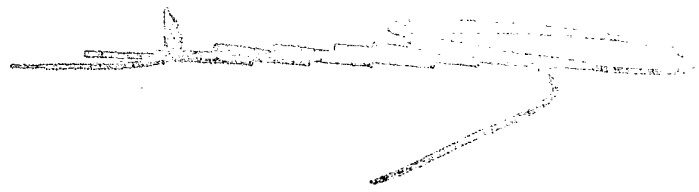


FIG. 45L



LINHA OMNIBUS DE ENDEREÇOS

00000000	01111110	00000000	10001110
31	24 23	16 15	8 7 0

FIG. 45M

LINHA OMNIBUS DE DADOS

AAAAAAAA	AAAAAAAA	XXXXXXXX	XXXXXXXX
31	24 23	16 15	8 7 0

FIG. 45N

LINHA OMNIBUS DE ENDEREÇOS

00000000	01111110	00000000	10001000
31	24 23	16 15	8 7 0

FIG. 45O

LINHA OMNIBUS DE DADOS

00000qbb	bbbbbbbbb	rspkkkkk	cxxxxxxx
31	24 23	16 15	8 7 0

FIG. 45P

LINHA OMNIBUS DE ENDEREÇOS

00000000	01111110	00000001	xxx00100
31	24 23	16 15	8 7 0

FIG. 45Q

LINHA OMNIBUS DE DADOS

xxxxxxx	aaaaaaaa	aaaaaaaa	aaaaaaaa
31	24 23	16 15	8 7 0

FIG. 45R

COMANDO/ESTADO

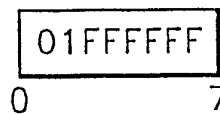


FIG. 45S

LINHA OMNIBUS DE ENDEREÇOS/DADOS

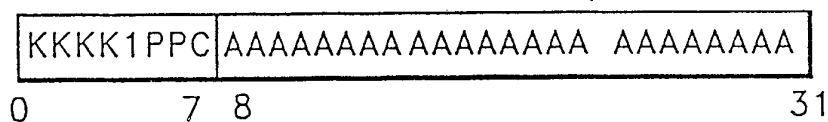


FIG. 45T

LINHA OMNIBUS DO PROCESSADOR

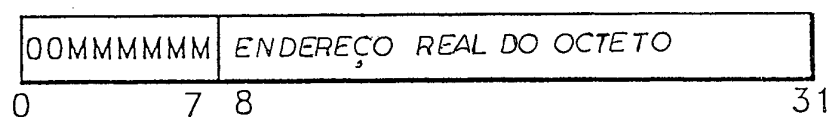


FIG. 45U

LINHA OMNIBUS DE CHAVE/ESTADOS

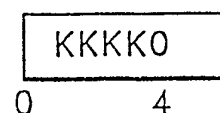


FIG. 45V

LINHA OMNIBUS DE ENDEREÇO

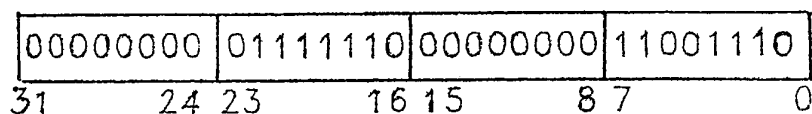


FIG. 45W

LINHA OMNIBUS DE DADOS

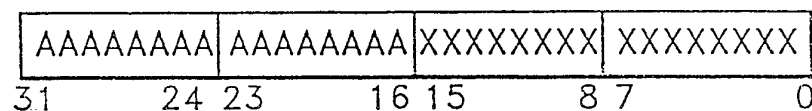


FIG. 45X

LINHA OMNIBUS DE ENDEREÇOS

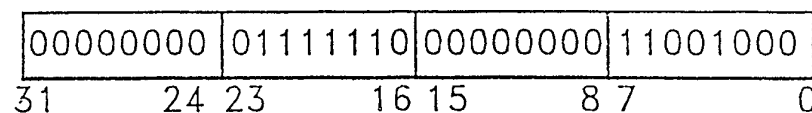


FIG. 45Y

LINHA OMNIBUS DE DADOS

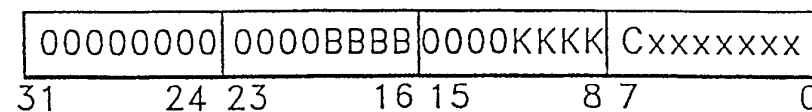


FIG. 45Z

COMANDO/ESTADOS

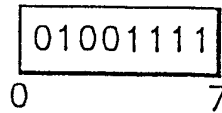


FIG. 45AA

LINHA OMNIBUS DE ENDEREÇOS/DADOS

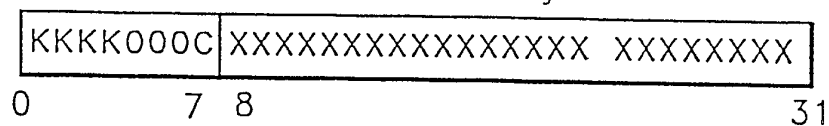


FIG. 45AB

LINHA OMNIBUS DO PROCESSADOR

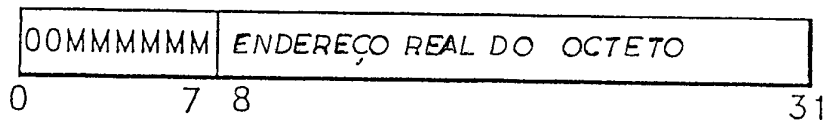


FIG. 45AC

LINHA OMNIBUS DE CHAVE/ESTADOS

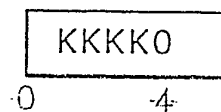


FIG. 45AD

LINHA OMNIBUS DE ENDEREÇOS

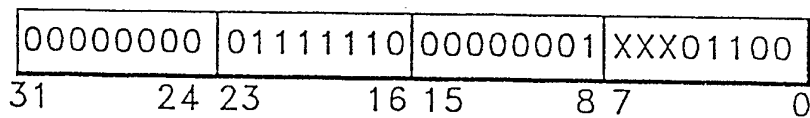


FIG. 45AE

LINHA OMNIBUS DE DADOS

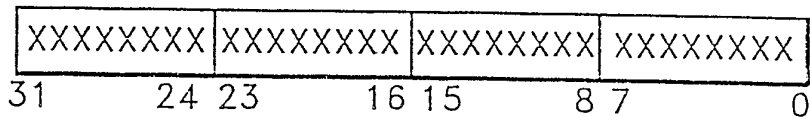


FIG. 45AF

LINHA OMNIBUS DE ENDEREÇOS

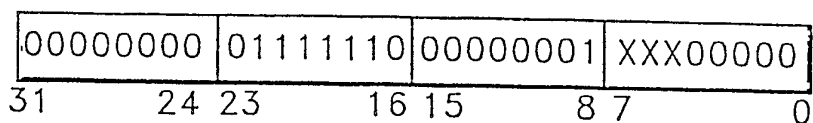


FIG. 45AG

FIG. 46A - EMULAÇÃO DO DISCO S/370

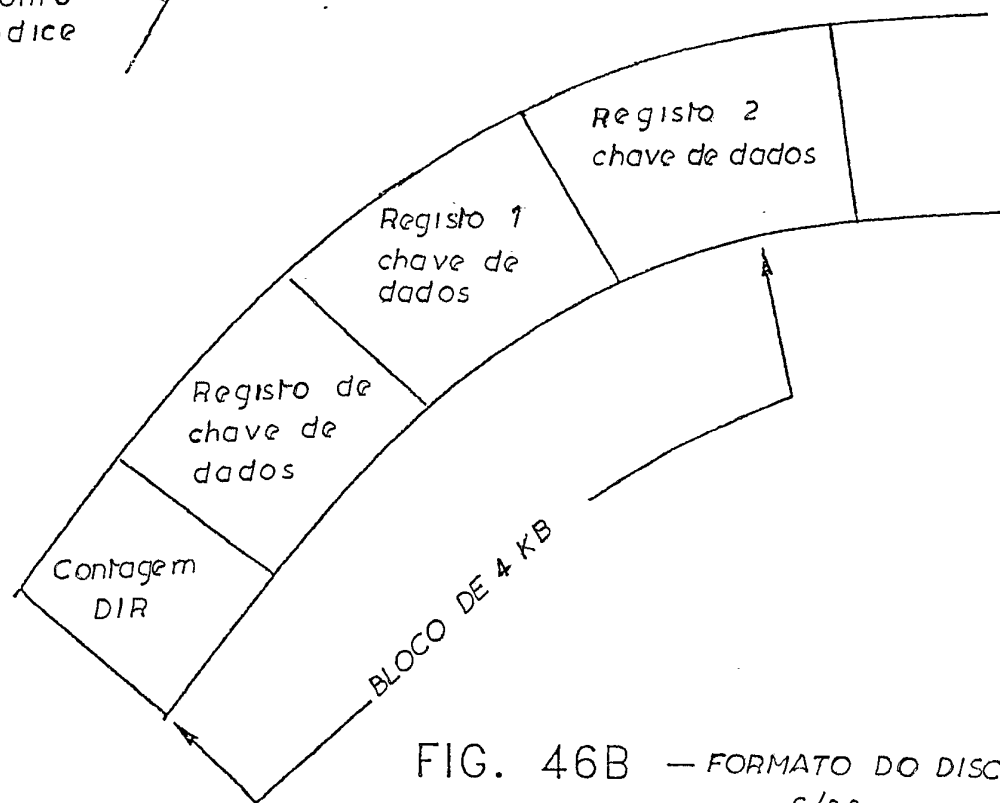
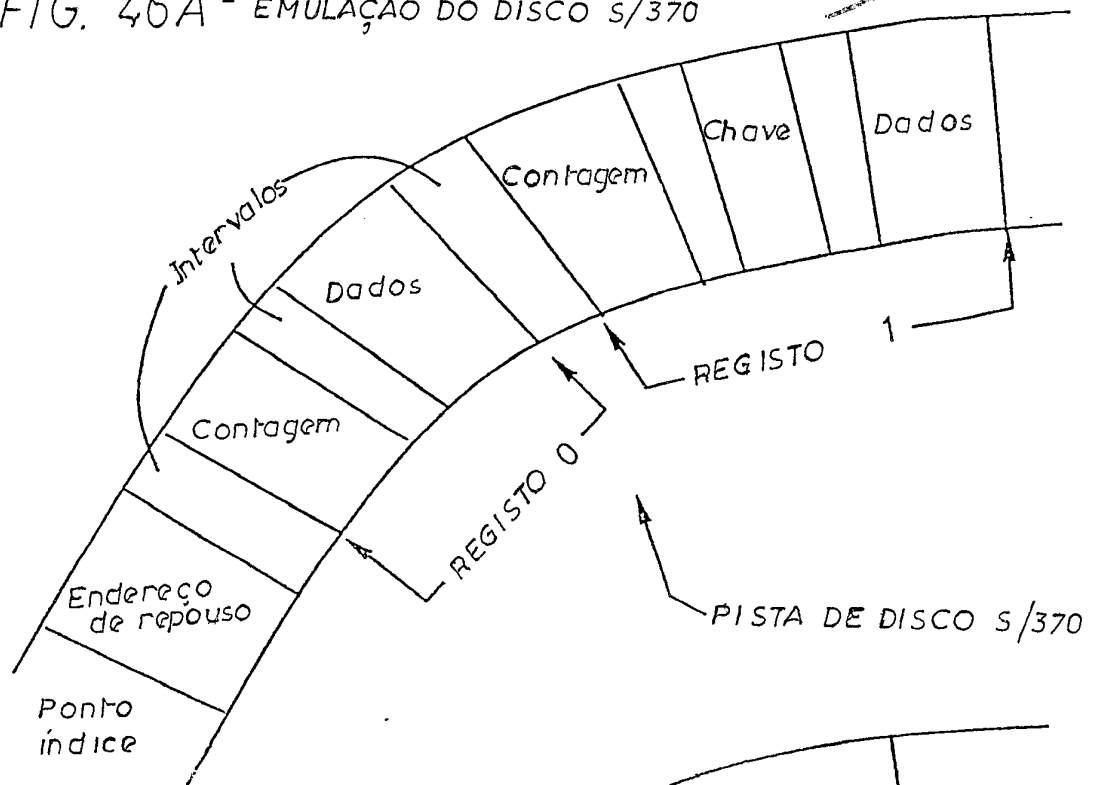


FIG. 46B - FORMATO DO DISCO S/88

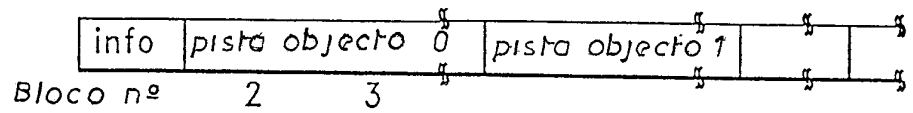


FIG. 46C - BLOCO DOS DISCOS S/88

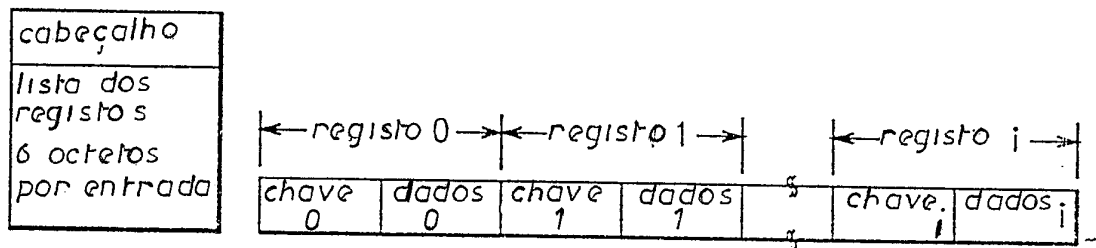


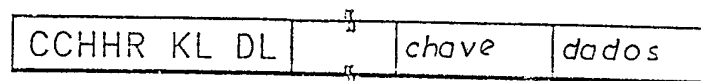
FIG. 46D

Endereço de repouso
de leitura de escrita

FCCHH

FIG. 46F

Registro zero



primeiro registro. na pista

FIG. 46G

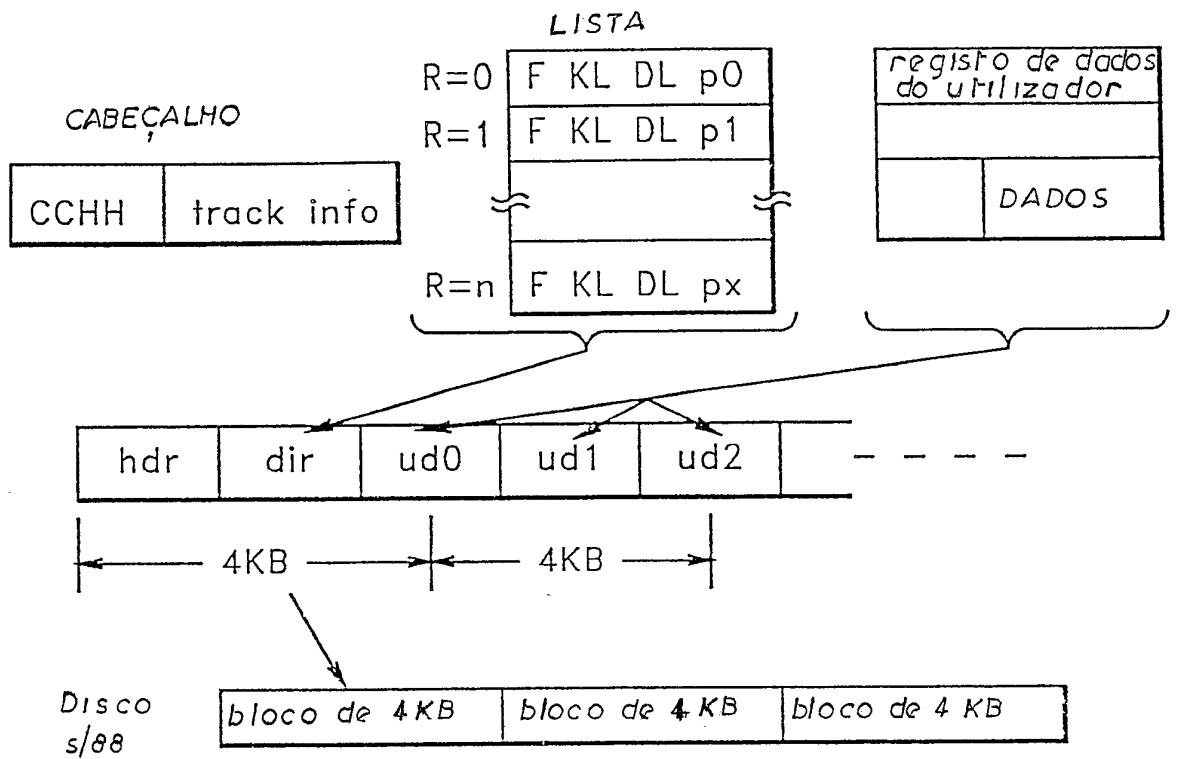


FIG. 46E ← FORMATO DE EMULAÇÃO DE CONTAGENS, CHAVES E DADOS



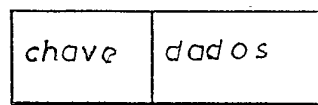
Contagem



registro seguinte na pista

FIG. 46H

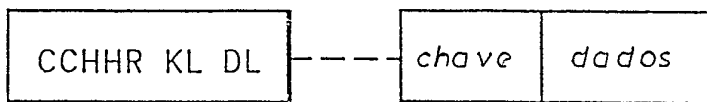
chave & dados



registro seguinte na pista

FIG. 46I

contagem e chave de dados



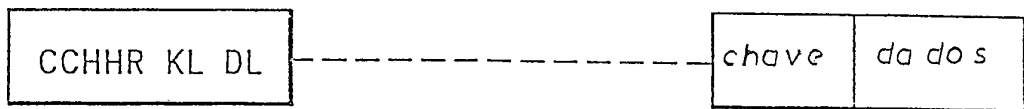
registro seguinte na pista

FIG. 46J

multicontagem chave e dados



registro seguinte na pista



ultimo registro na pista

FIG. 46K

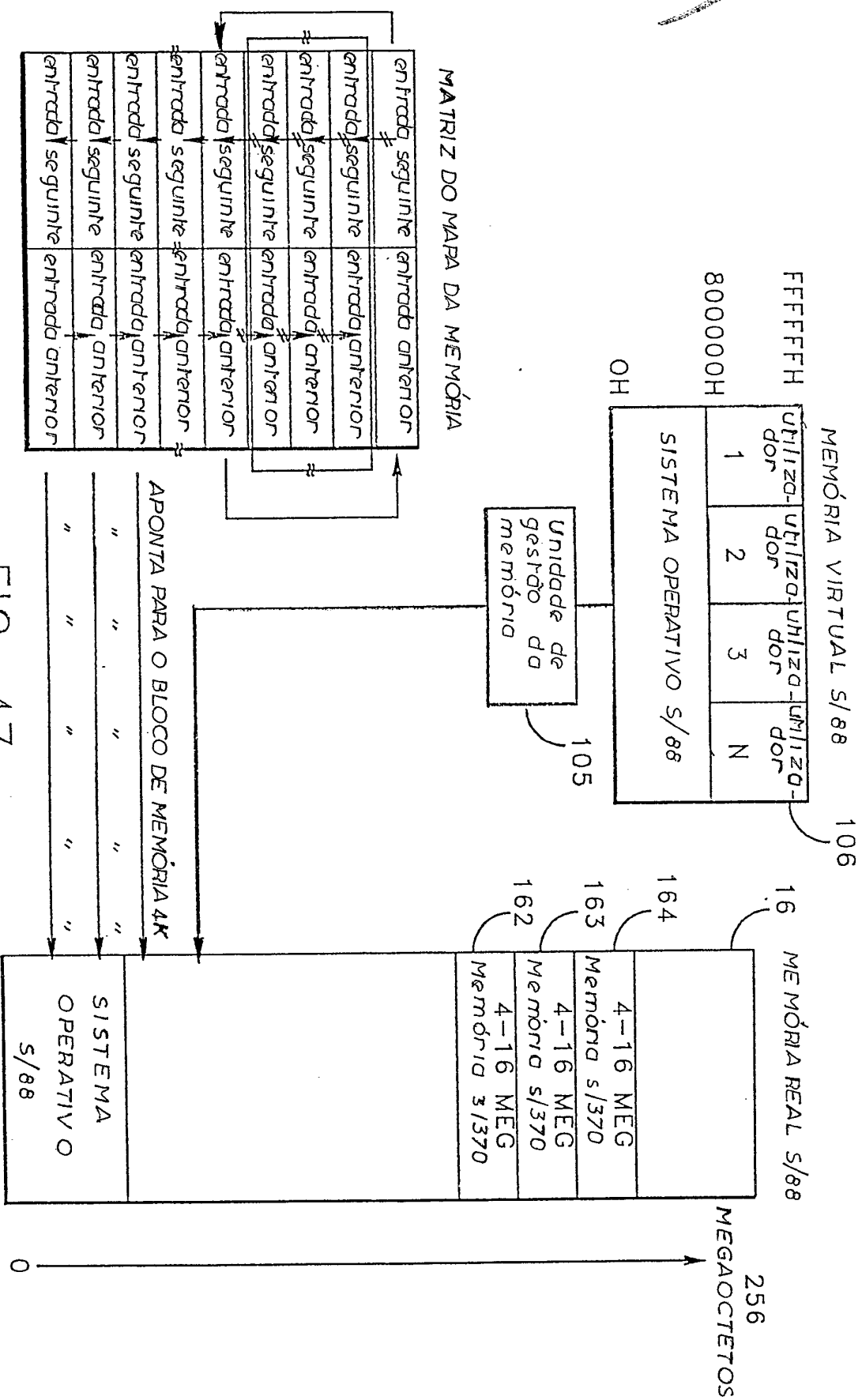


FIG. 47

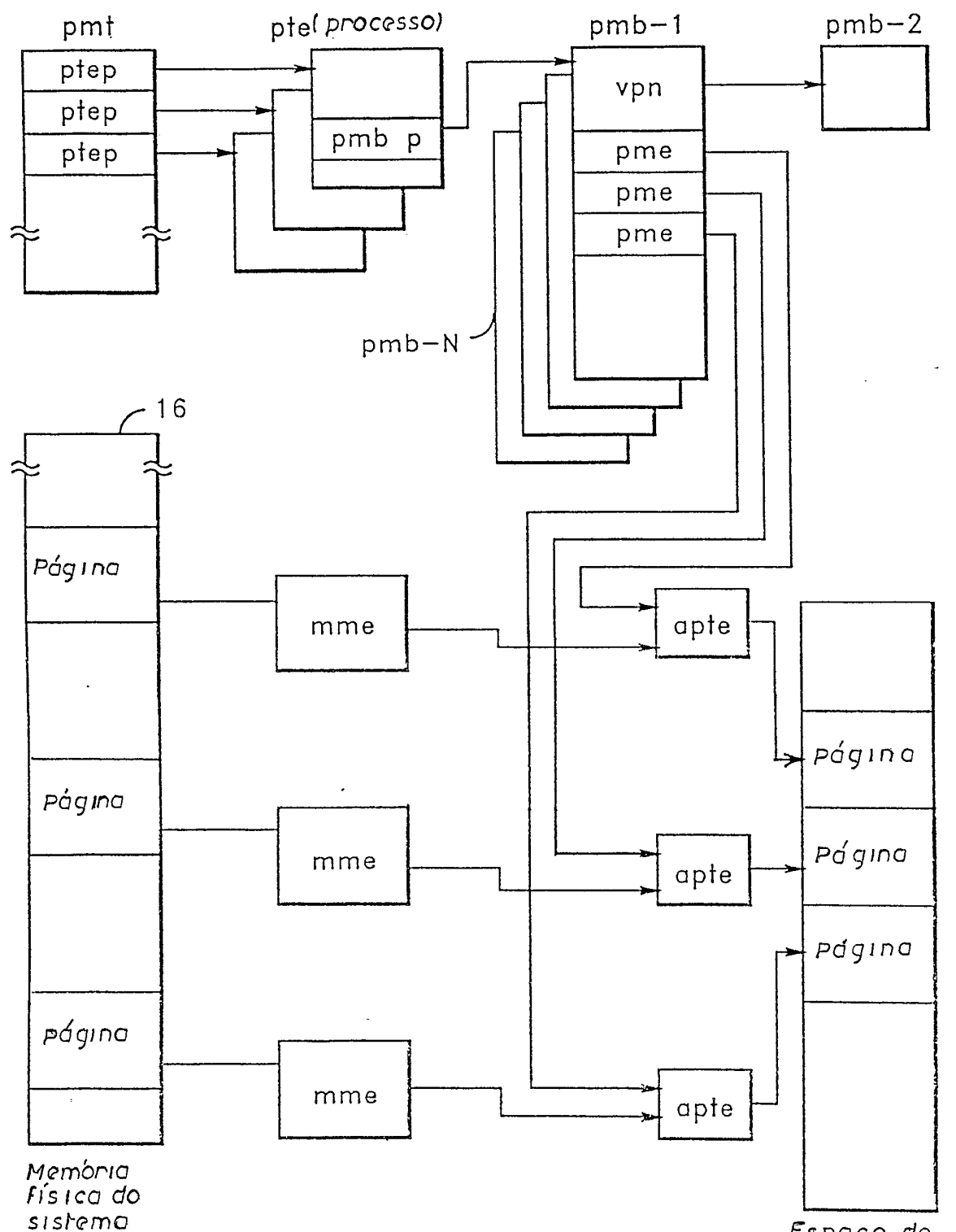


FIG. 48A — DISTRIBUIÇÃO DE POSIÇÕES (MAPA) DOS MEIÓS DE PROGRAMAÇÃO VIRTUAIS, REAIS

mme

indica- dores	mme seguinte	mme anterior	endereço	aptep

FIG. 48B — ENTRADA DO MAPA DA MEMÓRIA

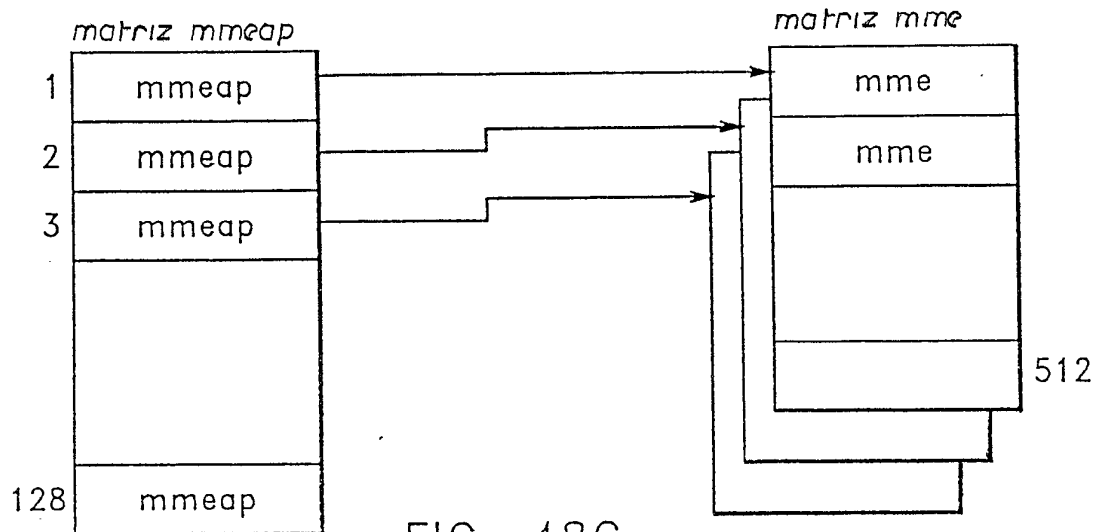


FIG. 48C

Mapa de memória infof-1

mem map infop seguinte 0

Mapa de memória baseada na infor.
(mem map infop)

	mem map infop seguinte	nº de páginas
1		ppn
2		ppn
16		ppn

FIG. 48D

apte

endereço	indica- dores	ref	vpage no.	processo ptr

bytes: 4 1 1 2 4

FIG. 48E — ENTRADA DA TABELA DE
PÁGINA ATIVA

bloco apt

0	apte
1	apte
255	apte

FIG. 48F — BLOCO DA TABELA DE PÁGINAS ACTIVAS

Aposto apt

	n procs	v base	n pages
	users (256 bits)		
0	n ptws	aptep	
1	n ptws	aptep	
npp	n ptws	aptep	
octetos: 2		4	

FIG. 48G — APOSTO DA TABELA DE PÁGINAS ACTIVAS

pte

Informação do cliente ; identificação do processo, nom, etc
Informação do processo ; prioridade, tipo , número , etc
Indicadores: listador do processo, controlo de páginas , informação das listas, etc.
Gestão de pc e mem, PMBp, map slot, protecção do utilizador, preparar mmes, páginas usadas, ficheiro vm parâmetros da tabela ROOT
Outros : contadores, perfil cpu, informações da pilha interrupções do processo, EITEp processo sub, memória "cachê" pedidos, etc.

Explícitos

first pmb ptr
map root tbl phys addr
map root ptr phys
pdr ptr

FIG. 48H — ENTRADA DA TABELA DO PROCESSO (pte)

pmb

	next p	base vpn	map addr
0	Indicador res	aptep	
63	Indicador res	aptep	

FIG. 48I — BLOCO DO MAPA DO PROCESSO

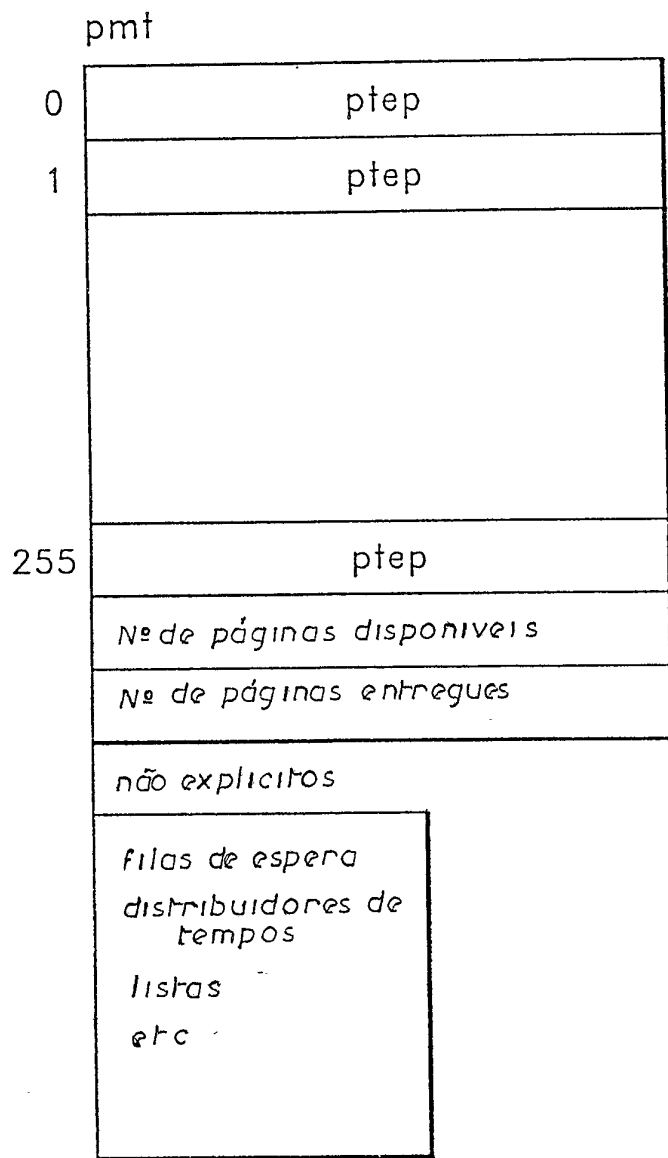


FIG. 48J — TABELA DE GESTÃO DO PROCESSO

ptw

u	ac1	ppn	ac2	
1	3	16	3	10

FIG. 48K — PALAVRA DA TABELA FÍSICA (ptw)

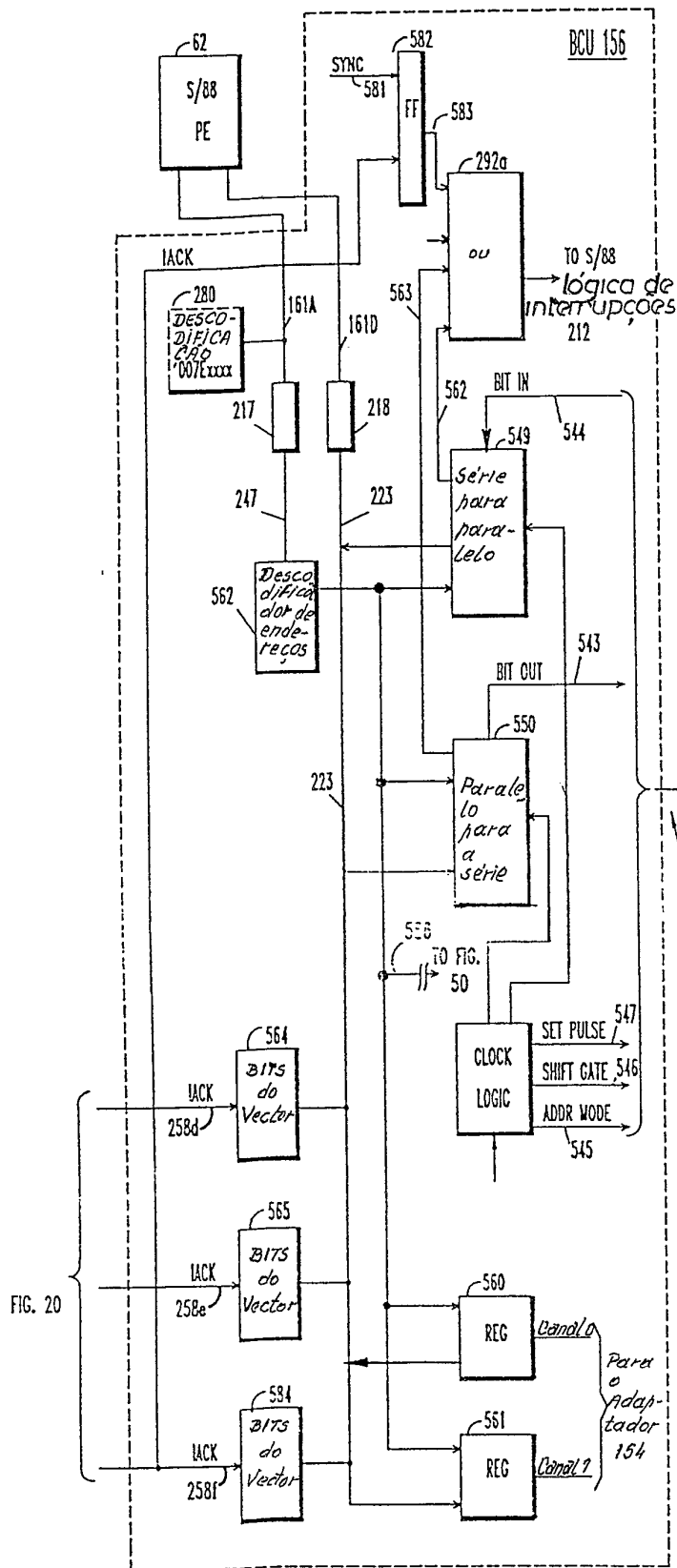
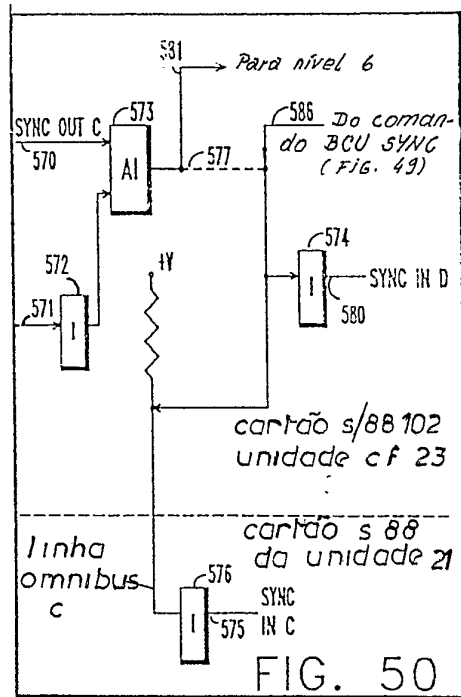
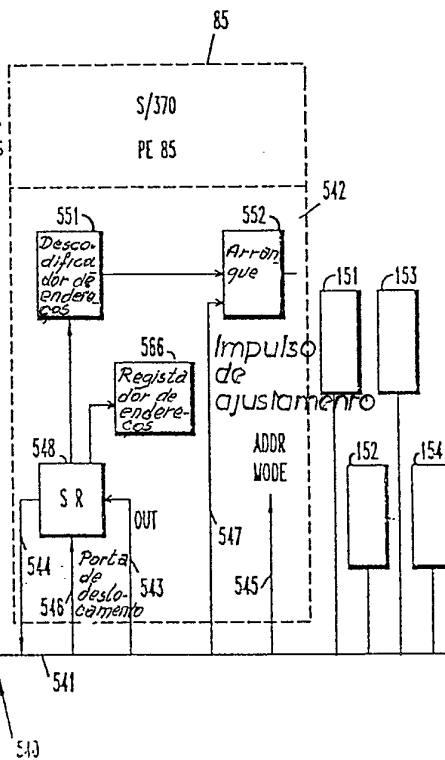
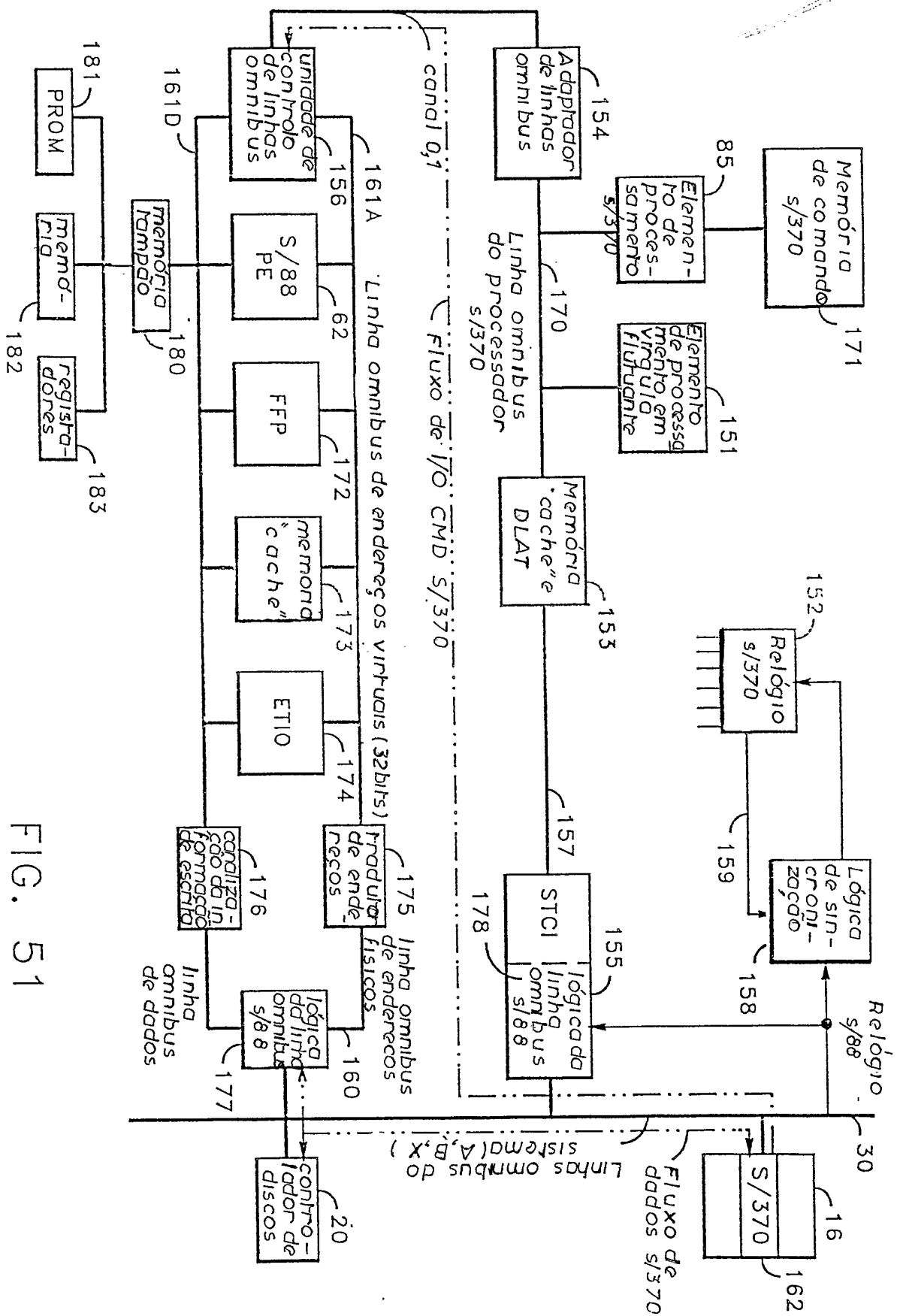


FIG. 49





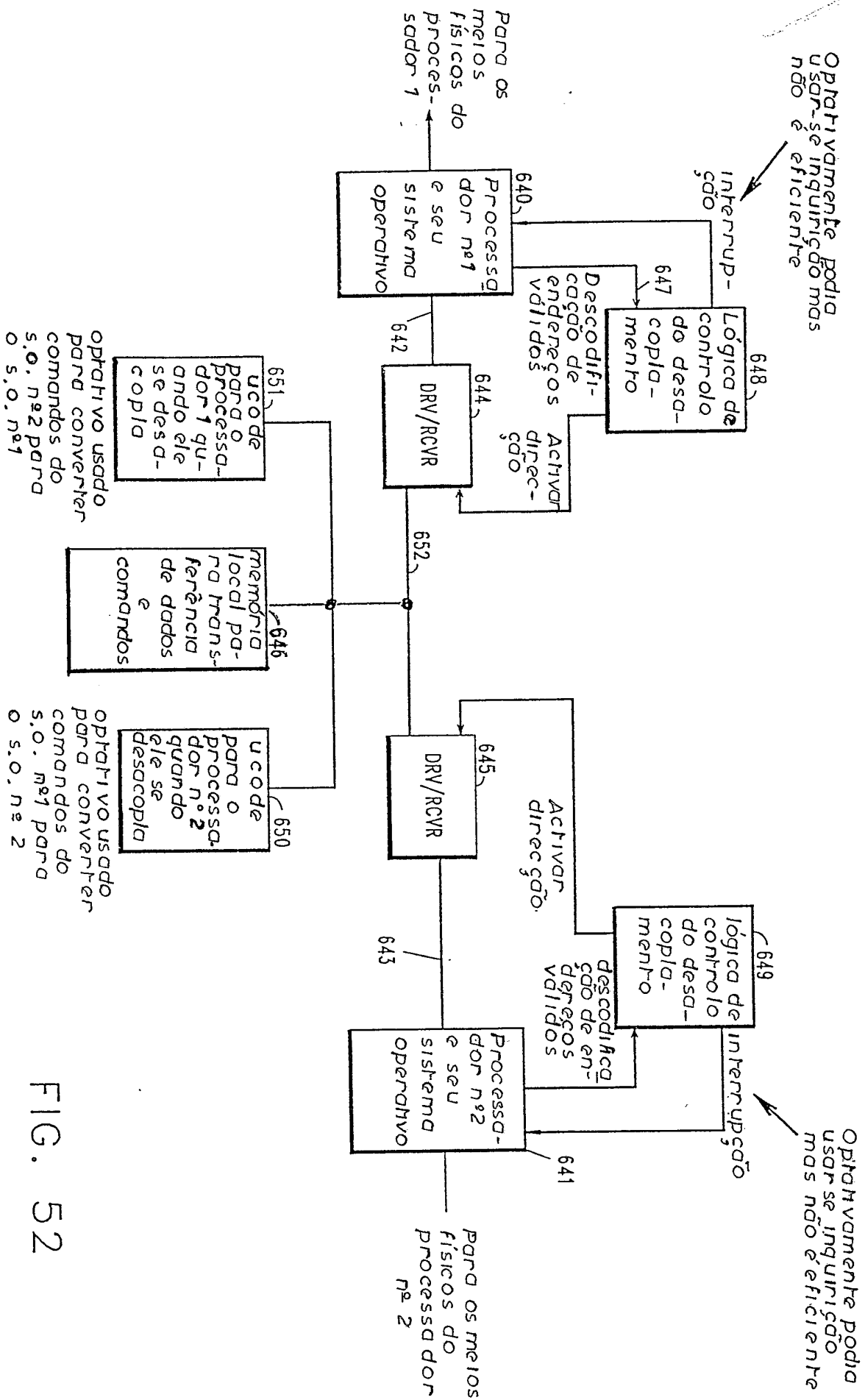


FIG. 52