I.C.M.S.C.

INSTITUTO DE CIÊNCIAS MATEMÁTICAS DE SÃO CARLOS

DESCRIÇÃO, ANÁLISE E CONSIDERAÇÕES SOBRE A IMPLEMEN
TAÇÃO DE UM DUTO PARALELO DE COMUNICAÇÃO DE ALTO
NÍVEL PARA UM SISTEMA DISTRIBUÍDO FUNCIONAL

Marcos José Santana

UNIVERSIDADE DE SÃO PAULO

SÃO CARLOS - SÃO PAULO
BRASIL

DESCRIÇÃO, ANÁLISE E CONSIDERAÇÕES SOBRE A IMPLEMEN
TAÇÃO DE UM DUTO PARALELO DE COMUNICAÇÃO DE ALTO
NÍVEL PARA UM SISTEMA DISTRIBUÍDO FUNCIONAL

Marcos José Santana

Orientador:

Prof. Dr. Jan F. W. Slaets

Dissertação apresentada ao Instituto
de Ciências Matemáticas de São Car-
los, da Universidade de São Paulo, pa-
ra obtenção do título de Mestre em
Ciências de Computação e Estatística

COMISSÃO EXAMINADORA

Prof. Dr. Jan F. W. Salaets (IFQSC-USP)

Prof. Dr. Fernão S. R. Germano (ICMSC-USP)

Prof. Dr. Valentin O. Roda (IFQSC-USP)

Dedico este trabalho

À Regina pelo incentivo, apoio e
compreensão em todas as etapas de
nossas vidas.

Ao André pelo novo sentido de
vida que nos trouxe.

AGRADECIMENTOS

Ao Professor Dr. Jan F. W. Slaets pela orientação e amizade.

Ao Professor Dr. Odelar Leite Linhares e Dr. Maximilian Emil Hehl pelo apoio que têm dado ao nosso grupo.

Ao Professor Dr. Fernão S. R. Germano pelo apoio, colaboração e incentivo em todas as nossas atividades.

Ao Departamento de Ciências de Computação e Estatística pelo incentivo no desenvolvimento científico de seu corpo docente.

Aos amigos que direta ou indiretamente colaboraram para a realização deste trabalho, em especial ao Onofre Trindade Junior e a Mônica M. R. Germano que nunca mediram esforços para prestar colaboração.

A secretária Sueli Ap. H. Ferreira pela datilografia do trabalho.

ÍNDICE

CAPÍTULO I - INTRODUÇÃO

1 - <i>CONSIDERAÇÕES GERAIS</i>	01
2 - <i>ORGANIZAÇÃO DA DISSERTAÇÃO</i>	03

CAPÍTULO II - REVISÃO BIBLIOGRÁFICA

1 - <i>SISTEMAS DE PROCESSAMENTO DE DADOS DISTRIBUÍDOS (SPDD)</i>	05
2 - <i>DUTOS DE SISTEMAS COMPUTACIONAIS</i>	21

CAPÍTULO III - SOBRE O SISTEMA DISTRIBUÍDO FUNCIONAL (SDF) ASPECTOS FÍSICOS E LÓGICOS DA ESTRUTURA DO SISTEMA

1 - <i>DESCRIÇÃO DO SISTEMA</i>	42
2 - <i>ARQUITETURA DO SDF</i>	45
3 - <i>ALGUNS MÓDULOS PARA O SDF</i>	46
4 - <i>DUTOS DE INTERLIGAÇÃO DO SDF</i>	51
5 - <i>SOBRE O SOFTWARE DE CONTROLE DO SDF</i>	52

CAPÍTULO IV - SOBRE A COMUNICAÇÃO NO SDF

1 - <i>ASPECTOS DO MECANISMO DE COMUNICAÇÃO ENTRE MÓDULOS DO SDF</i>	55
2 - <i>O DUTO DE COMUNICAÇÃO DE ALTO NÍVEL (DCAN)</i>	64
3 - <i>ASPECTOS FÍSICOS DA ARQUITETURA DE CONTROLE DO DCAN..</i>	78
4 - <i>CONSIDERAÇÕES SOBRE OS COMANDOS DE ALTO NÍVEL DO DCAN</i>	88
5 - <i>ASPECTOS DA ESTRUTURAÇÃO LÓGICA DO SDF</i>	102
6 - <i>O DUTO DE DADOS E ENDEREÇOS (DDE)</i>	110

CAPÍTULO V - SOBRE ALGUMAS CONSIDERAÇÕES PARA IMPLEMENTAÇÃO DO SDF

1 - <i>UMA CONFIGURAÇÃO BÁSICA PARA O SDF</i>	113
---	-----

2 - DIRETRIZES PARA IMPLEMENTAÇÃO DA INTERFACE DO DCAN ..	121
CAPÍTULO VI - CONCLUSÕES	
1 - CONCLUSÕES FINAIS	132
2 - A CONTINUIDADE DO TRABALHO	136
BIBLIOGRAFIA	137
APÊNDICE	
A - ABREVIATURAS EMPREGADAS	140
B - GLOSSÁRIO	141
C - ROTINAS DESENVOLVIDAS PARA A IMPLEMENTAÇÃO DESCRITA EM V-2	143

DESCRIÇÃO, ANÁLISE E CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO
DE UM DUTO PARALELO DE COMUNICAÇÃO DE ALTO NÍVEL PARA
UM SISTEMA DISTRIBUÍDO FUNCIONAL

RESUMO

Este trabalho apresenta a conceituação básica sobre sistemas de processamento de dados distribuídos e dutos de sistemas computacionais, destacando-se as características de alguns dutos padronizados.

É apresentada uma proposta de arquitetura distribuída para um sistema composto por módulos funcionais (o SDF). A arquitetura proposta difere de diversas outras arquiteturas distribuídas, pelo seu aspecto funcional, onde a distribuição do processamento ocorre através de tarefas completas.

Como meio de interligação dos módulos do SDF, propõem-se a utilização de dois dutos paralelos: um para a comunicação a nível de dados (DDE) e outro para a comunicação através de comandos de alto nível (DCAN), que acionam diretamente o processamento distribuído. Tem-se a descrição e a análise do funcionamento do DCAN, com a apresentação das diretrizes básicas para implementação da interface e um conjunto de comandos de alto nível.

DESCRIPTION, ANALYSIS AND CONSIDERATIONS ON THE IMPLEMENTATION OF A HIGH LEVEL PARALLEL COMMUNICATION BUS FOR A FUNCTIONAL DISTRIBUTED SYSTEM

ABSTRACT

This work presents the basic concepts of distributed data processing systems and computing systems buses, bringing out the features of some standard buses.

A distributed architecture suggestion for a system composed by functional modules (the SDF), is presented. The suggested architecture differs from many other distributed architectures in its functional aspect, where the distribution of processing occurs by complete tasks.

For the interconnection of the SDF modules two parallel buses are suggested: one for data level communication (DDE) and another for high level communication commands (DCAN), used to operate the distributed processing. The description and analysis of DCAN operations are treated together with the basic lines for interface implementation and a high level commands group.

CAPÍTULO I

INTRODUÇÃO

1 - CONSIDERAÇÕES GERAIS

O desenvolvimento de sistemas baseados em microprocessadores tem sido bastante explorado nos últimos anos, constituindo uma área de trabalho em constante evolução.

Os grupos de "Instrumentação Eletrônica" do Instituto de Física e Química de São Carlos - USP e de "Hardware e Software Básico" do Instituto de Ciências Matemáticas de São Carlos - USP, tem procurado unir esforços no sentido de trabalhar no desenvolvimento de sistemas computacionais baseados em microprocessadores, buscando acompanhar a evolução da área.

Uma tendência fortemente manifestada nos sistemas computacionais modernos é a descentralização do processamento, procurando não concentrar as diversas tarefas em um único elemento processador, de modo a obter um sistema com maior flexibilidade e melhor desempenho, tanto do ponto de vista computacional, como do ponto de vista físico e econômico.

A tecnologia atual favorece o desenvolvimento de sistemas com arquitetura distribuída [Fa83] e, em particular, os sistemas distribuídos baseados em microprocessadores.

Os sistemas modernos tem sido desenvolvidos procurando obter certa independência quanto ao processador e facilitar a assimilação dos avanços tecnológicos [So81].

Essas características favorecem, fortemente, diversos setores de aplicação dos sistemas computacionais e de modo especial aos sistemas empregados no controle de processos e aquisição de dados, onde se tem uma grande carga de processamento de dados e a exigência da supervisão do processador para o controle das grandezas físicas envolvidas, por exemplo.

Para o desenvolvimento desses sistemas deve-se ter certo domínio da conceituação envolvendo os sistemas distribuídos de um modo geral, e das técnicas utilizadas na interligação dos seus diversos componentes. O estudo de dutos de sistemas computacionais e dos mecanismos de comunicação entre módulos de

um sistema distribuído, constituem um ponto fundamental para o desenvolvimento dessa classe de sistemas.

Neste trabalho a idéia básica é o estudo de uma arquitetura distribuída, composta por módulos funcionais, independentes, existindo um conjunto de tarefas (ou funções) pré-estabelecidas para cada um. Nesse caso não se procura estabelecer uma arquitetura distribuída que suporte o processamento distribuído a nível de instruções ou operações elementares, mas sim uma distribuição classificada como de "alto nível", definida pela caracterização funcional de cada módulo do sistema. A distribuição do processamento ocorre através de comandos de alto nível (que circulam pelo Duto de Comunicação de Alto Nível - DCAN) que acionam tarefas (ou funções) completas, definidas nos módulos do sistema.

O objetivo do trabalho é apresentar e discutir a arquitetura proposta para o sistema (Sistema Distribuído Funcional - SDF), traçar diretrizes básicas para sua implementação e, em especial, estabelecer um duto de comunicação de alto nível para a interligação dos módulos (o DCAN).

O sistema proposto apresenta características favoráveis tanto à execução de tarefas computacionais normais (desenvolvimento de programas, por exemplo), como para funções mais específicas voltadas à instrumentação e controle de processos.

O sistema não é direcionado à otimização do tempo de uso do processador e à solução de problemas de armazenamento de dados em larga escala. Nesses casos, como enfatizado em {Fa83}, deve-se evitar o uso de sistemas distribuídos, empregando-se processadores de maior potência computacional e arquiteturas adequadas aos problemas considerados.

A modularidade do SDF permite a sua implementação a partir de uma versão inicial, simplificada, evoluindo para versões mais complexas conforme interesse e disponibilidade.

2 - ORGANIZAÇÃO DA DISSERTAÇÃO

No capítulo II é apresentada a revisão bibliográfica efetuada onde se tem, inicialmente, uma síntese do estudo desenvolvido sobre sistemas de processamento de dados distribuídos. Em especial é apresentada uma conceituação para os sistemas distribuídos baseados em microprocessadores. Em continuação são abordados tópicos referentes a dutos de sistemas computacionais, com uma conceituação geral e uma análise dos principais dutos padronizados disponíveis atualmente, com a apresentação de um quadro comparativo das características desses dutos.

No capítulo III tem-se a apresentação geral do sistema distribuído objeto de estudo neste trabalho (o SDF). É apresentada sua arquitetura, possíveis módulos para o sistema e alguns aspectos lógicos do sistema operacional necessário.

No capítulo IV trata-se dos aspectos da comunicação no SDF. São abordados:

- a. uma proposta de mecanismo para troca de mensagens, entre módulos, empregando-se escaninhos de mensagens;
- b. definição, análise e o mecanismo de controle de um duto paralelo e multiplexado para a comunicação de alto nível entre os módulos;
- c. alguns possíveis comandos de alto nível para o DCAN, com uma análise do relacionamento entre as solicitações dos usuários e a comunicação de alto nível do SDF;
- d. a esquematização lógica de um módulo do SDF, mostrando o relacionamento existente entre usuário, software e interligações físicas do sistema;
- e. opções de utilização dos dutos padronizados considerados no capítulo II, como via de comunicação de baixo nível (dados e endereços) para o sistema.

No capítulo V são abordados aspectos relacionados com a implementação do sistema. Inicialmente é apresentada uma proposta de configuração básica para o sistema, detalhando-se a finalidade e importância de cada módulo. Em continuação, são apre-

sentadas algumas diretrizes para o desenvolvimento do sistema de interface e uma versão simplificada é apresentada. As rotinas desenvolvidas para a implementação dessa versão simplificada do sistema de interface são apresentadas no apêndice.

No capítulo VI são apresentadas algumas considerações relevantes referentes ao trabalho, as conclusões finais e as diretrizes para sua continuidade.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

1 - SISTEMAS DE PROCESSAMENTO DE DADOS DISTRIBUÍDOS (SPDD)

1.1 - Considerações Iniciais

O projeto de sistemas computacionais tem sido influenciado fortemente, pelo desenvolvimento tecnológico dos últimos anos.

O rápido avanço atingido na tecnologia de memórias e dispositivos lógicos, juntamente com a redução de tamanho e de custo dos dispositivos LSI, tem favorecido projetos de novos sistemas, cada vez mais sofisticados.

Os ambientes de processamento de dados de um modo geral sofreram nos últimos anos uma mudança radical, imposta pelo avanço tecnológico. Os tradicionais sistemas de processamento em lotes tem sido substituídos por sistemas interativos de multiprogramação centralizados, e por sistemas de processamento de dados distribuídos (redes locais, redes geograficamente distribuídas ou sistemas interligados por duto).

Os microprocessadores deram uma nova perspectiva aos projetos de sistemas computacionais, tornando economicamente viável a busca de novas arquiteturas que reünam características mais adequadas a conjuntos de aplicações.

Soltis, em {So81}, apresenta certas considerações sobre o desenvolvimento de sistemas computacionais na década de 80. A base de seu artigo é o desenvolvimento do IBM/38, mas as considerações apresentadas tem um caráter geral, servindo como orientação para futuros projetos.

É salientado em {So81}, que o projeto de sistemas computacionais, com características de processamento distribuído ou não, deve não apenas se valer dos avanços tecnológicos existentes, mas também manter um canal aberto para a assimilação das novas tendências tecnológicas. Caso contrário, corre-se o risco de se tornar obsoleto em pouco tempo.

A queda no custo dos dispositivos eletrônicos tem favorecido a tendência de incorporação no hardware, de um grande

número de funções, que tradicionalmente fazem parte do software. Como consequência, a implementação dos sistemas operacionais é simplificada e o desempenho dos sistemas melhorado consideravelmente.

1.2 - *Caracterização de Sistemas Distribuídos*

Embora a necessidade e importância dos SPDD ser reconhecida em diversas aplicações computacionais, não existe uma definição que satisfaça aos especialistas da área {St84}.

A falta de um critério uniforme para a definição dos SPDD, é parcialmente devida ao fato da característica de distribuição ocorrer em diversos níveis do sistema, tais como:

- a. arquitetura interna da máquina;
- b. disposição física;
- c. interconexão;
- d. sistema operacional;
- e. controle;
- f. organização da base de dados;
- g. aplicações dedicadas.

Uma definição para ser abrangente, deve levar em consideração todos os níveis possíveis de distribuição ou uma parte considerável desses.

Enslow, em {En78}, faz algumas considerações básicas sobre os SPDD e apresenta diretrizes para a caracterização de sistemas computacionais, como SPDD.

Para que um dado sistema seja considerado distribuído, Enslow considera cinco requisitos, como indispensáveis:

- a. existência de múltiplos recursos de propósito geral (tanto do ponto de vista físico, como lógico);
- b. existência de uma distribuição física desses recursos, que devem interagir por meio de uma interconexão;
- c. existência de um sistema operacional de alto nível, que integre o controle dos componentes distribuídos;
- d. transparência do sistema, permitindo que os usuários utilizem todos os recursos, sem necessidade de conhecimento do tipo de distribuição empregada;
- e. existência de um protocolo de comunicação entre os módulos

do tipo de "cooperação autônoma", sem a existência de qualquer tipo de hierarquia pré-estabelecida.

Certamente, poucos sistemas conhecidos satisfazem todos esses requisitos, dada a forma rígida com que o autor os considera.

Enslow reconhece que a aplicação de seus critérios, não pode ser feita de forma binária e propõem uma aplicação através do que ele denominou "dimensões que caracterizam a distribuição".

São consideradas três dimensões para a classificação dos sistemas:

- a. descentralização do hardware ;
- b. descentralização do controle;
- c. descentralização da base de dados.

Com isto é possível classificar os sistemas segundo o grau de distribuição encontrada para cada dimensão, estabelecendo-se uma escala indo desde os sistemas totalmente centralizados até os sistemas totalmente distribuídos.

Em {En78}, a aplicação dos critérios sugeridos é rígida, mesmo levando em consideração as três dimensões propostas, porém, dependendo das circunstâncias, certos sistemas que não se enquadram integralmente na caracterização de Enslow podem ser considerados distribuídos, segundo pontos de vista mais flexíveis, como apresentado em {Fa83,Sp77}.

Em {Fa83,Sp77} são considerados, principalmente, os SPDD baseados em múltiplos microprocessadores e a conceituação apresentada é mais flexível do que a de Enslow {En78}.

A caracterização particular para essa classe de SPDD é apresentada no item 1.5 deste capítulo.

1.3 - Considerações sobre Comportamento em Sistemas com Múltiplos Processadores

Os SPDD constituem uma classe de sistemas com múltiplos processadores {En78,Fa83,Sp77,St84}.

A análise de alguns resultados existentes na literatura, sobre medidas de comportamento de sistemas com múltiplos processadores, serve para esclarecer o interesse nesse tipo de arquitetura de sistema.

Considerando a taxa de transferência de dados (ttd) do sistema, idealmente espera-se que haja um acréscimo conforme novos processadores sejam adicionados. Na prática, porém, observa-se que devido ao efeito da saturação, a partir de um certo número de processadores, ocorre exatamente o contrário {Ch80, Fa83}.

A figura II-1.1 apresenta um gráfico típico para o caso, servindo para enfatizar que os SPDD não devem ser pensados como a solução geral para o problema do aumento da potência de processamento de um sistema.

O efeito da saturação é definido em {Fa83} como a degradação na taxa de transferência de dados, devido ao aumento do número de processadores. Pode ser atribuído ao aumento na disputa pelos recursos compartilhados e à sobrecarga nos canais de comunicação.

Chu em {Ch80}, aborda a ocorrência da saturação nos SPDD e alerta para a relação existente com o problema da alocação das tarefas, nos diversos processadores do sistema. Quando a alocação é feita dinamicamente (através de um Sistema Operacional especializado, por exemplo), devem ser considerados algoritmos que procurem amenizar esse fator. Quando a alocação é estática, duran

te a fase de projeto, o projetista do sistema deve estar atento, evitando surpresas futuras.

Cita-se em {Ch80}, que se o projeto for mal direcionado pode ocorrer a saturação a partir de 3 ou 4 processadores a penas.

Com relação a desempenho-custo do processador, espera-se que o desempenho cresça conforme aumenta o custo do procesador.

Em sistemas com um único processador, mostra-se que o desempenho cresce mais rapidamente do que o custo (a lei de Grosch {Fa83} estabelece uma relação quadrática). Nesse caso não é interessante aumentar o número de processadores.

Mas, com o avanço da tecnologia dos microprocessadores, o custo por processador foi reduzido, podendo ser estabelecida uma nova relação desempenho-custo, com um aspecto mais linear, conforme figura II-1.2 .

Assim, torna-se economicamente atrativo o emprego de múltiplos processadores, visando o aumento do desempenho do sis-

tema.

O compartilhamento de recursos dentro do sistema é uma característica da maioria dos sistemas com múltiplos processadores (em particular dos SPDD) que é influenciada, principalmente, por considerações financeiras.

Esse compartilhamento deve ser minuciosamente estudado, para que o sistema não tenha o seu desempenho reduzido, devido a problemas ocasionados por disputa dos recursos.

Normalmente, podem-se compartilhar dispositivos de entrada e saída de dados, unidades de armazenamento secundário, módulos de memória, etc..

O problema da tolerância à falha, pode ser resolvido em sistemas com múltiplos processadores, através da transferência de tarefa de um processador para outro. Isto, certamente, só é possível se a arquitetura implementada tem disponibilidade de recursos para tal.

Também em sistemas com múltiplos processadores, pode-se obter uma maior flexibilidade tanto a nível de software, como de hardware. Em geral, quanto mais flexível o sistema mais complexos se tornam os módulos que o compõem, o que pode reduzir a confiabilidade individual de cada um.

A flexibilidade impõem um maior grau de modularidade e consequente facilidade de expansão e manutenção.

A modularidade, definida como sendo uma medida do grau de compactação e isolamento dos elementos de um sistema, constitui um parâmetro que caracteriza fortemente os SPDD.

Sistemas modulares, geralmente são mais fáceis de se projetar e depurar, pois os módulos tem hardware e software menos complexo. Além disso, a versatilidade do sistema é aumentada e a otimização a nível de tarefas dedicadas é facilitada.

Também o ciclo de vida do sistema é prolongado, pois a modularização permite expansões e substituições de módulos.

As considerações apresentadas aqui, procuram dar uma visão inicial de algumas características dos sistemas com múlti-

plos processadores, em particular dos SPDD, servindo como um parâmetro básico para se justificar o emprego desses sistemas.

1.4 - Organização dos Sistemas Distribuídos

Aspectos Lógicos

Com relação à estrutura lógica dos sistemas com múltiplos processadores, em {St84,Fa83,Sp77} são citadas as organizações vertical e horizontal.

Na primeira, tem-se uma estrutura hierárquica (ou piramidal), com um único mestre e diversos escravos, conforme figura II-1.3 .

Figura II-1.3 - Estrutura Hierárquica

Nesta organização, nem todos os processadores ocupam o mesmo nível lógico. A taxa de transferência de dados obtida é

bastante elevada.

No segundo tipo de organização, tem-se a necessidade de uma coordenação mais sofisticada, pois todos os elementos ocupam o mesmo nível lógico, podendo haver uma comunicação livre entre eles (como por exemplo vide a estrutura da figura II-1.4).

Geralmente este tipo de organização lógica é mais flexível, porém podem haver sérios problemas de sobrecarga de comunicação nos meios de interligação dos módulos.

Figura II-1.4 - Organização horizontal

Aspectos Físicos

Do ponto de vista de estrutura física, consideram-se dois aspectos básicos:

- a. o meio pelo qual ocorrem as comunicações entre os processadores, que em geral pode ser uma memória comum (centralizado) ou uma estrutura de interligação por duto (distribuído);
- b. a topologia de interconexão, que mostra a maneira com que os processadores podem ser interconectados fisicamente.

Existem diversas maneiras para se estabelecer uma interconexão física entre N elementos de um sistema. Os principais fatores levados em consideração para a escolha do tipo de interconexão são:

- a. confiabilidade do sistema (possíveis falhas de alguns módulos, não devem afetar o funcionamento global do sistema);
- b. a expandibilidade do sistema (o esquema adotado deve facilitar a inserção ou retirada de módulos no, ou do sistema original, além de permitir que a implementação inicie com um mínimo e cresça conforme as necessidades ou disponibilidades).

Topologias Básicas

Os tipos básicos de topologia de interconexão para sistemas com múltiplos processadores, citados em {Fa83,Sp77,St84,Ta84} e apresentados na figura II-1.5 são:

- a. duto comum
- b. ligação em estrela
- c. ligação em anel
- d. sistema totalmente interconectado

Outros tipos de interconexão podem ser empregados, porém, basicamente são composições das quatro citadas.

1.5 - Sistemas com Microprocessadores Distribuídos (SMD) Caracterização de um SMD

Como apresentado no ítem 1.3, o desenvolvimento da tecnologia dos microprocessadores tornou economicamente atrativo o uso de múltiplos processadores em sistemas computacionais.

Para um posicionamento mais adequado dos sistemas com microprocessadores distribuídos dentro do contexto geral de sistemas com múltiplos processadores, é considerado o modo de interação entre os sub-sistemas interligados por algum tipo de topologia.

As considerações extremas, apresentadas por Fathi em {Fa83}, são:

- a. as organizações de múltiplos processadores com acoplamento livre, conhecidas principalmente como "Redes de Computado-

res". Nesse caso tem-se computadores autônomos, distribuídos geograficamente, com processamento independente, características de computação local, etc..

- b. as organizações de múltiplos processadores, com acoplamento rígido, conhecidas como sistemas multiprocessadores, onde se tem arquitetura com memória comum, sistema operacional único, recursos compartilhados, igual capacidade nos diversos processadores, sincronismo entre os processadores, etc..

Os sistemas com microprocessadores distribuídos, geralmente são implementados segundo estruturas que combinam as melhores qualidades de cada um dos extremos citados anteriormente. A organização resultante, apresenta um acoplamento moderado.

As principais características apresentadas por essa classe de sistemas distribuídos são:

- a. autonomia de cada elemento (cada elemento individual do sistema, em geral consiste de uma UCP, programa de controle e memória de dados local, podendo usar ou controlar periféricos);
- b. processadores dedicados a tarefas (idealmente, cada elemento seria dedicado a uma tarefa específica, a qual indicaria o seu grau de complexidade);
- c. processadores com complexidade variada (não existe a restrição de que os processadores sejam iguais);
- d. otimização do software e do hardware (devido às tarefas dedicadas, cada elemento tem um software e um hardware voltados a sua especialização);
- e. comunicação entre processadores, geralmente a nível de dados, embora existam casos de comunicação a nível de comandos;
- f. compartilhamento da carga de processamento estático, pois os processadores são dedicados. O balanceamento da carga de processamento, deve ser feito na fase de projeto do sistema.

Utilização de SMD

Em [Fa83], são apresentadas certas considerações sobre onde utilizar sistemas com múltiplos microprocessadores.

Como conclusão, tem-se que os sistemas com múltiplos microprocessadores devem ser considerados para:

- a. aplicações de controle de processo, de caráter elaborado, com grande e diversificada demanda computacional e processamento em tempo real;
- b. aplicações com excessivo processamento de entrada e saída de dados;
- c. aplicações que necessitam de alta confiabilidade, mas devi

do a fatores financeiros ou de espaço, não permitam uma redundância em grande escala;

Em oposição, tem-se que um sistema distribuído, com múltiplos microprocessadores deve ser evitado quando:

- a. tudo o que se espera é, simplesmente, aumentar a capacidade de processamento do sistema; é preferível, nesse caso, o emprego de um processador mais poderoso ou processadores dedicados (ou especializados);
- b. a tarefa global a ser executada, não pode ser dividida em sub-tarefas, com relativa independência e uma mínima necessidade de intercomunicação.

Diferença entre Redes de Microcomputadores e SMD conectados por Duto

Clark, em [C178], estabelece uma comparação entre os SPDD que constituem redes geograficamente distribuídas, redes locais e os sistemas baseados em dutos.

Da abordagem de Clark, observa-se que uma distinção inicial pode ser feita pelas dimensões das interconexões físicas existentes. Assim, as redes geograficamente distribuídas envolvem ligações que variam desde alguns metros até dezenas de milhares de quilômetros; os sistemas interconectados por duto, abrangem desde os sistemas com um único microprocessador, com ligações na ordem de 1 a 10 cm (ou menor), até os SMD que podem chegar à casa dos 100 m aproximadamente. As redes locais se encaixam entre os dois extremos, indo de alguns metros até vários quilômetros de comprimento das ligações.

A característica apresentada não serve como um critério de fácil aplicação, devido à intersecção existente.

Um outro ponto abordado por Clark, se refere mais especificamente a uma comparação entre redes locais e sistemas conectados por dutos.

Clark observa que um duto é, frequentemente, um conjunto de linhas paralelas, com linhas de dados, controle e endereços separadas (abordado com detalhes em II-1.2), enquanto que as redes tendem a transferir suas informações serialmente, sobre um único par de linhas.

As transferências de dados seriais, como nas redes, trazem a vantagem de economia de linhas de transmissão e eliminação dos problemas de interferência, que ocorrem em linhas paralelas, porém a taxa de transferência de dados é reduzida, uma vez que a informação não pode ser enviada em um único ciclo.

Os sistemas com microprocessadores interconectados por dutos, geralmente tem suas linhas de interconexão limitadas a distâncias pequenas, como será observado nos itens relacionados com os dutos existentes.

1.6 - Exemplo de Um Sistema com Microprocessador Distribuído

Na figura II-1.6, é apresentado um esquema de arquitetura básica de um SMD específico. De acordo com [Fa83], esse sistema apresenta uma arquitetura simples, confiável e possibilita a implementação de processamento concorrente.

Figura II-1.6 - Esquema Simplificado de um SMD

O acionamento das tarefas nesse sistema, é efetuado por um controlador hierárquico, que supervisiona um certo número de processadores (não necessariamente idênticos).

O sistema admite também uma memória comum que pode ser utilizada como um concentrador de mensagens e para armazenamento de dados comuns.

O controlador aceita requisições externas e internas e inicia as ações de controle apropriadas.

A sincronização e coordenação da tarefa requerida é efetuada de modo assíncrono (sinais de "handshake") através do duto de controle e sincronismo.

As diversas tarefas disponíveis no sistema, estão de finidas localmente. Para melhorar a confiabilidade pode-se utilizar uma estratégia de duplicação de definição da tarefa.

O exemplo apresentado serve para mostrar, de uma maneira mais prática, algumas características atribuídas aos SMD ao longo deste trabalho.

1.7 - Considerações Adicionais Sobre Redes de Computadores

Em alguns tópicos abordados anteriormente foram apresentadas certas considerações sobre redes de computadores, relacionadas com aspectos físicos e lógicos. Cabe, porém, destacar ainda alguns pontos sobre a organização lógica que permite a implementação de uma rede, a partir de unidades computacionais autônomas.

Implementação de Redes de Computadores

As topologias discutidas em II-1.4 {Fa83,Sp77,St84 , Ta84} podem ser empregadas para a implementação de redes de computadores, de um modo geral. Cada elemento da rede (um nó) tem capacidade de processamento local, constituindo isoladamente um sistema computacional autônomo. Nesse caso tem-se a existência de um sistema operacional local e o nó pode operar independentemente da rede, caracterizando assim o acoplamento livre citado por Fathi em {Fa83}.

Para a implementação de uma rede a partir de sistemas computacionais autônomos, deve-se agregar a cada unidade um software de apoio necessário à comunicação da rede e uma interconexão física entre os sistemas considerados, segundo a topolo-

gia adotada. Esse software de controle não altera o sistema operacional local, apenas amplia as possibilidades de processamento de cada sistema individualmente e, efetivamente, serve como interface entre o sistema operacional local e a rede implementada, conforme esquematizado na figura II-1.7 {Di82}.

Organização Lógica das Redes de Computadores {Ry81, Di82}

A organização do software de controle da rede é geralmente feita segundo uma estrutura de níveis (ou camadas) funcionais, hierarquicamente relacionadas, conforme figura II-1.8.

Essa organização traz vantagens em termos estruturais, permitindo a elaboração de rotinas de controle bem definidas que favorecem a manutenção do sistema. Por outro lado o relacionamento hierárquico introduz a necessidade de definição de protocolos em vários níveis.

Protocolo de Comunicação {Ry81,Di82}

Para se estabelecer uma comunicação entre dois nós de uma rede com implementação através de níveis funcionais, definem-se protocolos que inter-relacionam esses níveis de modo adequado. Na figura II-1.8 é apresentado um esquema básico da DECnet {Di82} mostrando esse relacionamento.

2 - DUTOS DE SISTEMAS COMPUTACIONAIS

2.1 - Considerações Iniciais

Um duto pode ser definido como sendo uma via de transmissão de dados, que conecta dois ou mais elementos de um sistema digital {Bo81,Gu84,Le78}.

A necessidade de interligação entre diferentes elementos de um sistema computacional, onde são necessárias transferências de dados, levou ao surgimento dos dutos.

Essa interligação pode ser feita de diversas maneiras, como por exemplo através de uma ligação ponto a ponto, conforme figura II-2.1 .

Esse tipo de interligação, porém, é bastante cara e complexa, quando comparada com interligações por dutos, com linhas padronizadas e compartilhadas pelos elementos do sistema, conforme figura II-2.2 .

O emprego de um duto comum torna o sistema simples e barato, mas perde-se a presença do paralelismo existente numa ligação ponto a ponto, reduzindo sua velocidade.

Assim, em determinadas situações, onde a velocidade de transmissão de dados é um fator limitante, procura-se implementar dutos adicionais, com características de alta velocidade, contornando parcialmente o problema.

Um exemplo prático do problema, apresentado em {Le78}, são as máquinas mais velozes da Digital Equipment Corporation da série PDP 11, que apesar de terem sido desenvolvidas após o estabelecimento do padrão UNIBUS, tiveram em sua arquitetura a adição de dutos de alta velocidade, conforme figura II-2.3 .

Borril, em {Bo82a}, enfatiza que o duto é um elemento vital na arquitetura do sistema. Se é bem projetado, flexível, e rápido, obtém-se a máxima potência da arquitetura do processador, além de contribuir para uma maior generalidade do sistema. Ao contrário, se o duto é mal projetado, rígido ou lento, haverá limitação em sua aplicação e se tornará obsoleto, em pouco tempo.

Na definição de um duto é estabelecido um conjunto de linhas de comunicação, compartilhadas pelos elementos do sistema, com características mecânicas e elétricas bem definidas, e um protocolo de sinais que possibilita a comunicação entre os elementos interligados.

2.2 - Funcionamento Básico de Um Duto

O Conceito de Mestre e Escravo

As transações de dados que ocorrem no duto, são feitas por elementos do sistema que constituem mestres e escravos.

Um mestre é um elemento capaz de iniciar e controlar uma transferência de dados, solicitando a participação de um ou mais escravos.

Um escravo é um elemento capaz de participar de uma transferência de dados, por solicitação de algum mestre.

Existe também a possibilidade de dispositivos que atuam ou como mestre, ou como escravo (por exemplo, nos sistemas com múltiplos processadores).

A Transferência de Informação

Uma transferência de informação através do duto é sempre ativada pelos mestres.

As linhas do duto ficam disponíveis apenas para um mestre, em cada transação. No caso da existência de mais de um mestre requisitando o uso do duto, deve haver um processo de arbitragem, que decide qual é o mestre que acessará o duto. O problema de arbitragem é considerado com maiores detalhes em ítem posterior (*Arbitragem do Duto*).

O mestre que acessa o duto, pode trocar informações com um ou mais escravos.

O processo de troca de informações envolve:

- a. O mestre coloca o endereço do escravo nas linhas adequadas do duto; os escravos monitoram essas linhas, verificando se não estão sendo solicitados. Quando o endereço no duto coincide com o do escravo, este é conectado ao duto. O endereçamento pode ser feito, basicamente, por três esquemas distintos:
 - a.1. Endereçamento geográfico, onde a posição física do elemento no duto dá o seu endereço. É um método bastante utilizado em sistemas modulares;
 - a.2. Endereçamento lógico, onde o endereço do escravo é definido no próprio módulo, independente de sua posição no duto. Este método é versátil, considerando-se a independência da posição física e o uso racional do espaço de endereçamento. Provoca alguns problemas na manutenção dos sistemas.
 - a.3. Endereçamento conjunto ("Broadcast"), onde é estabelecido uma transação entre um mestre e vários escravos, ao mesmo tempo.
- b. Durante o tempo de conexão entre mestre e escravo, dados são transferidos em ambos os sentidos;
- c. Terminada a troca de informação, o mestre interrompe a conexão, liberando os escravos e o duto;

O processo de transação entre mestre e escravos, requer também um sincronismo, que é obtido com sinais de temporização enviados durante a troca de dados (detalhado em *Sincronismo*

do Duto).

Os Tipos de Transferência de Dados

Uma maneira de se diferenciar os dutos, é pelo número de linhas disponíveis para a troca de dados.

Basicamente, existe o duto serial e o duto paralelo.

No duto Serial, a informação é transmitida por uma única linha, havendo uma multiplexação dos sinais.

No duto Paralelo, a informação é transmitida por um conjunto de linhas paralelas.

Se o duto é totalmente paralelo, então não existe multiplexação de sinais.

Se o duto é paralelo, mas os dados e os endereços, por exemplo, são transmitidos no mesmo conjunto de linhas, existe uma multiplexação.

A escolha entre as técnicas serial e paralela, depende das seguintes considerações:

- a. velocidade de transmissão de dados: quanto mais bits são transmitidos ao mesmo tempo, maior será a velocidade do duto;
- b. o custo dos cabos, conectores e circuitos de alimentação, cresce rapidamente conforme aumenta o número de linhas do duto;
- c. conforme o comprimento das linhas do duto, são necessárias técnicas mais sofisticadas e mais caras para a eliminação de ruídos, interferências e perdas.

Para reduzir o custo, diversos dutos usam multiple - xar dados e endereços, transmitindo-os em um conjunto único de linhas.

Em ligações maiores do que 100 metros, geralmente em pregam-se dutos seriais.

Arbitragem do duto

A arbitragem do duto é o processo desencadeado, toda vez que mais de um mestre solicita o uso do duto.

No processo de arbitragem, vários mestres requisitam o uso do duto e apenas um, de acordo com regras pré-estabelecidas é autorizado a utilizar o duto.

Borril, em {Bo82b}, destaca que a técnica de arbitra

gem adotada no duto é uma decisão importante no seu projeto, pois a variedade das arquiteturas que podem ser implementadas, depende dessa decisão.

O mecanismo de arbitragem utilizado pode ser:

a. *arbitragem centralizada*: onde um único módulo do sistema, recebe os sinais de requisição do duto e decide qual dos mestres será o vencedor da disputa.

Os esquemas centralizados são mais flexíveis na alocação de prioridades para os mestres, mas são mais sensíveis à falha e requerem linhas dedicadas no duto, para a interligação em estrela dos diversos elementos.

A figura II-2.4, mostra um esquema desse tipo de arbitragem.

b. *arbitragem distribuída*: onde cada elemento do sistema, contém circuitos que decidem quando o elemento é o vencedor da disputa do duto.

Os esquemas distribuídos são menos sensíveis à falha, flexíveis quanto à definição da prioridade do mestre no duto e facilitam a inserção ou retirada de módulos do duto.

Basicamente, os esquemas de arbitragem distribuída se dividem

em dois grupos: decodificada e codificada.

b.1. na arbitragem decodificada, quando ocorre uma disputa, todos os mestres requisitantes acionam a linha que corresponde à sua prioridade. Cada mestre observa então essas linhas, começando pela de maior prioridade. Se existir uma linha de prioridade maior acionada, o mestre se retira da disputa. Caso contrário é o vencedor e ganha o controle do duto.

Esse tipo de arbitragem é de implementação simples, porém exige um número elevado de linhas dedicadas no duto, conforme aumentam os mestres.

b.2. na arbitragem codificada, cada mestre possui um código (único) e no instante da disputa, os requisitantes colocam nas linhas de arbitragem do duto, ao mesmo tempo, os seus códigos. Se um mestre envolvido observa um "bit" de maior prioridade acionado, retira o seu código das linhas, liberando o seu pedido.

Depois de um certo tempo (variável) de arbitragem, apenas um mestre estará com todos os bits nas linhas, tendo acesso ao duto.

Esse tipo de arbitragem é bastante empregado nos dutos modernos, e o problema do tempo de arbitragem requer certas considerações. Em {Ha80,Ta76,Ta82} são apresentadas alternativas para a redução desse tempo.

Na figura II-2.5 é apresentado um esquema para implementação da arbitragem codificada.

c. *esquema híbrido*: onde são observadas características dos dois anteriores. O esquema de encadeamento em cascata ("Daisy Chain") apresentado na figura II-2.6 é um exemplo desse tipo de arbitragem.

Figura II-2.6 - Esquema de arbitragem híbrido ("Daisy-chain")

Nesse caso, a linha de requisição é compartilhada por todos os mestres. A linha de reconhecimento (ou autorização) é única, e o sinal é transmitido através dos mestres. O mestre de maior prioridade, que está na disputa, recebe o sinal de reconhecimento e bloqueia os demais.

Aqui, a prioridade é definida pela posição física do mestre no duto. Existem problemas quando ocorrem falhas, pois se um mestre é retirado do duto, a cadeia é interrompida. Também o tempo de atraso, devido a propagação do sinal de reconhecimento através da cadeia de mestres, pode ser problemático.

Sincronismo do Duto

As transações entre mestres e escravos, através do duto, necessitam de algum tipo de sincronismo.

Define-se um duto como sendo síncrono, quando a validade dos sinais transferidos, possui um diagrama de tempo fixo (sinal de relógio pré-estabelecido). Nesse caso tem-se, teoricamente, a maior taxa de transferência de dados.

Um duto é dito assíncrono, quando a validade dos sinais transferidos é feita através de uma troca de sinais entre mestre e escravo. Isto define um protocolo conhecido por "handshake". Aqui o ciclo do duto é variável no tempo, permitindo uma maior flexibilidade em termos de acomodação de diversos tipos de processadores, periféricos e memórias, num mesmo duto.

Em geral, os dutos utilizados são assíncronos (mesmo sendo mais lentos). Nas aplicações onde a velocidade é um fator limitante, os tempos envolvidos na troca de sinais de sincronismo são dispensáveis, e dutos síncronos são recomendados.

Protocolo do Duto

Define-se como protocolo do duto, ao método escolhido pelo projetista, para validar os sinais que são transferidos no duto [Gu84].

Quando as transações no duto são síncronas, o protocolo estabelecido é síncrono; caso contrário o protocolo é assíncrono.

O estabelecimento de um protocolo é necessário porque numa transação através do duto, nem todos os bits transmitidos estão prontos ao mesmo tempo. Assim, deve-se indicar a quem recebe a informação, o momento certo para a recepção dos bits en

viados.

Interrupções

Uma interrupção é um pedido de atendimento vindo de um periférico, podendo ter inúmeras finalidades, como indicação de término de tarefa, indicação de falha, etc..

A utilização de interrupção permite que cada periférico seja atendido por um programa especializado, que é ativado quando a solicitação de interrupção é recebida e aceita.

Uma outra alternativa para esse atendimento, seria uma varredura cíclica, pela UCP, verificando em cada ciclo quais periféricos estão de prontidão. Esse esquema apresenta um desperdício de tempo de UCP e uma certa lentidão no atendimento [Ta76].

O esquema com interrupções é preferível, podendo-se ter interrupções com endereço fixado e interrupções vetorizadas, onde o endereço da rotina de atendimento é fornecido quando ocorre a solicitação da interrupção.

Os dutos normalmente possuem um conjunto de linhas dedicadas às interrupções, com certa prioridade estipulada.

2.3. - Classificação dos Dutos em Níveis de Interligação

Em função do nível em que ocorrem as interligações Borril propõem, em [Bo81], a seguinte classificação:

- a. *Nível 0 (dutos do nível de componentes)*: é o nível de duto mais baixo, que o projetista pode manipular diretamente. Existem níveis inferiores de interligação, mas que são definidos internamente aos circuitos integrados. (Ex. Microbus-National, Z Bus-Zilog etc..)
- b. *Nível 1 (dutos do nível de barramento)*: consistem de interligações que possibilitam a comunicação entre módulos do sistema, conectados em um barramento comum. Fornecem sinais com capacidade de acionamento adequada e complementam os sinais de controle, que não são fornecidos no nível 0. (Ex. Multibus-Intel, Versabus-Motorola, ZBI-Zilog, S-100 etc..)
- c. *Nível 2 (dutos do nível de Sistemas)*: consistem de interligações de dutos do nível 1. Em sistemas simples e pequenos, este nível se confunde com o nível 1.

Em [Zu81], Zuffo apresenta uma classificação semelhante à de

Borril, porém, os níveis 1 e 2 aparecem como uma mesma classe de dutos.

A implementação desses dutos, é feita por interconexão direta, por cabos, de um barramento a outro ou por meio de um sistema que reforce os sinais, para suportar a carga total das linhas.

O comprimento das linhas oferece alguns problemas, que devem ser considerados. Os problemas frequentes são interferências, tempo de propagação, etc..

d. *Nível 3 (Dutos de interface paralela - DIP):* constituem duas classes.

d.1. *DIP mapeado:* que é uma extensão parcial do duto do sistema, onde alguns ou todos os sinais de dados podem ser transportados através de circuitos retentores ("latches") para fora do sistema, com um certo número de linhas de endereço;

d.2. *DIP Não-Mapeado:* que é uma via de dados, separada dos sinais normais de transferência de dados entre o processador e sua memória. Frequentemente é implementado com dispositivos LSI.

e. *Nível 4 (Dutos de interface Serial):* que constituem os dutos para comunicação a longas distâncias, largamente empregados em Redes de Computadores.

Zuffo, em [Zu81], salienta que embora se faça uma classificação desse tipo, não existe em grande parte dos casos práticos, uma distinção nítida entre eles; existe ao contrário uma certa aproximação do nível 0 com os níveis 1 e 2 e destes com os níveis 3 e 4.

2.4 - Considerações Sobre Definição e Padronização de Dutos

O duto constitui uma parte fundamental no desenvolvimento de sistemas computacionais. Uma definição (ou escolha) inadequada tem reflexos imediatos em diferentes fases do projeto, ou da utilização do sistema [Be80,Bo80,Bo82a,Gu84].

O avanço da tecnologia vem permitindo cada vez mais, novos processadores, memórias e periféricos. Os sistemas computacionais projetados atualmente, devem possuir a habilidade de fã-

cil assimilação desses novos dispositivos, sem a necessidade de uma drástica redefinição do projeto.

Os sistemas modernos, geralmente com características modulares, raramente permanecem estáticos, requerendo mais memória, mais espaço em disco, uma taxa maior de transferência de dados etc..

Assim, na definição do duto deve ser procurada uma estrutura capaz de fornecer opções de expansão futura, e que apresente certo grau de independência do processador, facilitando o uso de processadores diversificados.

Existem muitas propostas distintas de dutos para a implementação de sistemas computacionais, baseados em microprocessadores ou não. Esses dutos geralmente contam com diversos circuitos de interfaceamento produzidos em LSI, facilitando a implementação.

A modularidade dos sistemas computacionais, favorece a existência de compatibilidade entre módulos de diferentes origens, podendo ser utilizados em um mesmo sistema.

A priori, isto parece não ser fácil, pois diferentes fabricantes definem sua própria estrutura de duto, e os seus módulos são construídos para esses dutos.

Para conseguir uma certa compatibilidade entre diversos fabricantes, há necessidade do estabelecimento de dutos padronizados e os módulos serem produzidos para um determinado duto padrão.

Em {Bo82a,Bo82b}, Borril faz certas considerações sobre os problemas envolvidos na padronização de dutos. O estabelecimento de um padrão envolve, além dos aspectos técnicos, aspectos políticos e interesses particulares.

Apesar das dificuldades, existem diversos comitês de padronização e alguns dos padrões propostos tem tido certa aceitação.

O IEEE mantém grupos de trabalho que analisam propostas de padronização de dutos. Os padrões sugeridos, tem sido bastante difundidos e vem sofrendo alguns aperfeiçoamentos, conforme surgem as necessidades. Alguns dos padrões propostos (ou em estudo) pelo IEEE, são apresentados no item 2.5.

Alguns dos aspectos considerados na padronização de dutos são:

- a. independência do processador
- b. tipos e quantidade de elementos (módulos) permitidos
- c. esquema de arbitragem do duto
- d. esquema de interrupções do duto
- e. protocolo de transações
- f. definição das linhas do duto, que são basicamente:
 - f.1. dados e endereços
 - f.2. controle, estado, interrupção e arbitragem
 - f.3. alimentação (tensão e terra)
 - f.4. linhas extras para expansão
- g. características físicas, tais como:
 - g.1. tamanho das placas
 - g.2. tipo de conector
 - g.3. comprimento das linhas
 - g.4. espaçamento entre placas
 - g.5. terminação das linhas
- h. circuitos acionadores e acopladores
- etc..

2.5 - Alguns Dutos Para Sistemas com Microprocessadores

Como já observado neste trabalho, existe uma grande variedade de dutos propostos para a implementação de sistemas computacionais, baseados em microprocessadores.

O interesse imediato está nos dutos a nível de barramento, por tratarem da interligação de módulos, em sistemas computacionais.

Dentre os dutos existentes, destacam-se:

a. S-100 (IEEE-P696) {Ga82, St79, Zu81}

Proposto pela MITS em 1976, se tornou um dos dutos de barramento mais difundidos, sendo mais tarde proposto como padrão (P696) pelo IEEE.

Destina-se a sistemas computacionais baseados em microprocessadores de 8 ou 16 bits, com um máximo de 22 módulos interconectados por um barramento comum de 100 linhas paralelas.

Nesse duto, tem-se mestres (permanente e temporários) e escravos. O sistema admite 1 mestre permanente e até 16 mestres temporários.

Pode-se implementar sistemas com arquiteturas distribuídas,

respeitando-se os limites apresentados (22 módulos, 16 mestres temporários e 1 permanente).

As 100 linhas do duto S-100 estão organizadas em:

- duto de dados - 16 linhas
- duto de endereços - 24 linhas
- duto de estado - 8 linhas
- duto de controle de E/S - 11 linhas
- duto de controle DMA - 8 linhas
- duto de interrupção vetorizada - 8 linhas
- duto de alimentação - 9 linhas
- linhas de sinais de relógio - 2
- linhas de funções de zeramento - 3
- linhas de controle geral - 3
- linhas reservadas - 5
- linhas não definidas - 3

A arbitragem do duto é distribuída e do tipo codificada (4 bits).

O duto é síncrono e permite 8 níveis de interrupções vetorizadas e uma interrupção não mascarável.

A ocorrência de um erro (erro de paridade, escrita em memória protegida etc..), pode ser sinalizada por uma linha especial. A utilização do S-100 deve ser feita em dutos de pequena distância e em ambientes com ruído elétrico relativamente baixo.

b. *Multibus* (IEEE-P796) {Bo80,Bo82,Zu81}

Proposto originalmente pela Intel em 1974, se tornou mais tarde proposta de padronização do IEEE (P796).

Destina-se a sistemas computacionais baseados em microprocessadores de 8 ou 16 bits e possibilita a conexão de diversos mestres, para a implementação de arquiteturas com multiprocessamento.

Possui 86 linhas organizadas em:

- duto de dados - 16 linhas
- duto de endereços - 22 linhas
- duto de alimentação - 20 linhas
- duto de controle - 17 linhas
- interrupções - 8 linhas
- reservas - 3 linhas

A estrutura do duto está baseada no conceito de mestre-escravo, com protocolo assíncrono, permitindo que módulos de diferentes velocidades possam ser interligados.

Permite a implementação da arbitragem do duto por um esquema de encadeamento em cascata ("daisy-chain") ou por um esquema paralelo, centralizado.

As interrupções podem ser requisitadas ativando-se uma das 8 linhas disponíveis, com prioridade associada (mais alta na linha 0). O reconhecimento da requisição é dado por uma linha de controle própria. É possível ter-se interrupções vetorizadas ou não.

c. *STD* (IEEE-P961) {E182}

Proposto pela Mostek e Pro-log em 1978, se tornou uma proposta de padrão do IEEE (P961) em 1981.

Destina-se a sistemas computacionais baseados em microprocessadores de 8 bits.

É um duto síncrono, não multiplexado, para interconexão de módulos (nível de barramento).

A capacidade de endereçamento básica é de 64 k bytes (16 linhas) com possibilidade de expansão (1 linha de expansão).

Permite a implementação de sistemas com múltiplos processadores.

O STD possui 56 linhas organizadas em:

duto de dados - 8 linhas

duto de endereços - 16 linhas

duto de controle - 18 linhas

extensão de memória - 1 linha

interrupções - 3 linhas

duto de alimentação - 10 linhas

A arbitragem do duto pode ser implementada por encadeamento em cascata.

A prioridade das interrupções vetorizadas pode ser implementada por um esquema paralelo e centralizado, que exige ligações especiais com cada periférico, ou de modo serial por encadeamento, utilizando 2 linhas do duto de controle.

d. *Eurobus* {Hi82}

Desenvolvido pelo UK Ministry of Defence e pela Ferranti

Computer System (1978), constitui um duto de barramento para sistemas computacionais baseados em microprocessadores de 8, 16 e 32 bits.

É um duto assíncrono, multiplexado, projetado para independer do processador utilizado, podendo até ser utilizado como duto de dados, sem necessidade de um processador como controlador do duto.

Permite a implementação de sistemas com arquiteturas distribuídas.

O esquema de arbitragem é centralizado, exigindo linhas de requisição, conectadas em estrela, através do duto. A arbitragem é feita antecipadamente, melhorando o desempenho do sistema.

As interrupções são tratadas por um arbitrador, que conhecendo a fonte da interrupção, deve ativar as ações que se façam necessárias.

Em sistemas com múltiplos processadores, está prevista a possibilidade de interrupção direcionada a um dado processador. Está também prevista nesse duto, a possibilidade de interligação com outros dutos do mesmo tipo, onde dispositivos de um podem endereçar dispositivos de outros.

O Eurobus utiliza o Eurocard (ou o Eurocard duplo - ambos cartões de circuito impresso padronizado na Europa) e conector de 64 pinos.

O número de linhas utilizadas no duto, varia de acordo com o tipo de implementação.

e. *ZBI - ZBUS Backplane Interconnect* {Be80}

Constitui um duto de barramento, desenvolvido pela ZILOG, para interconexão de módulos de sistemas computacionais baseados em microprocessadores de 8, 16 e 32 bits (projetado para explorar as capacidades dos microprocessadores da ZILOG - Z80, Z8000 ...).

Esse duto é assíncrono, multiplexado e permite a implementação desde sistemas simples com um único processador até sistemas com arquiteturas distribuída e multiprocessamento.

É composto de 96 linhas, organizadas em:

duto de dados/endereços - 32 linhas

duto de paridade - 4 linhas

duto de interrupção - 9 linhas
duto de controle - 28 linhas
duto de alimentação - 21 linhas
linhas de reserva - 2 linhas

A definição do tamanho da palavra é indicado por 2 linhas do duto de controle.

As linhas do duto de paridade e uma de indicação de erro do duto de controle, permitem detetar erros nos dispositivos de memória e nas transações no duto.

As 9 linhas de interrupções, são organizadas em 3 grupos independentes, com níveis de prioridade diferentes. Dentro de cada grupo, a prioridade dos dispositivos é assinalada pela posição de cada um.

O tratamento das interrupções depende do processador e pode ser mascarável ou não, vetorizada ou não, conforme a configuração da UCP.

A arbitragem do duto utiliza 4 linhas de controle (o que permite até 16 mestres), e a lógica de arbitração pode ser distribuída codificada.

f. *VERSAbus* - *VMEbus* {De82,Ka81}

O *VERSAbus* e o *VMEbus* são dutos semelhantes, sendo o *VMEbus* uma versão Europeia do *VERSAbus*, com algumas modificações.

Constitui um duto de barramento, com características modernas, destinado aos sistemas computacionais baseados em microprocessadores de 8, 16 e 32 bits.

Permite a implementação de sistemas com vários processadores, tendo algumas facilidades para implementação de arquiteturas distribuídas.

Em termos de flexibilidade é um duto que permite o manuseio de dados com até 32 bits e endereços com até 31 bits.

O tamanho das placas dos módulos, não varia grandemente entre o *VERSAbus* e o *VMEbus*, sendo que o VME emprega o Eurocard (ou Eurocard duplo) e conector DIN.

É um duto assíncrono, estando organizado em cinco sub-dutos:

1. duto de transferência de dados
2. duto de interrupção
3. duto de arbitragem
4. duto de utilidade

5. duto de inter-inteligência

Em 1., tem-se 16 linhas para dados (expandíveis até 32), 23 linhas de endereço (expandíveis até 31).

A quantidade de bits de uma transação no duto, é definida por três linhas de controle.

O duto de interrupção (2) permite com 7 linhas, a geração de interrupções a partir de qualquer dispositivo do sistema. As linhas definem 7 níveis de prioridade e através da implementação de um reconhecimento por encadeamento, pode-se ter mais de um dispositivo gerando interrupção na mesma linha, num modo compartilhado.

O problema de arbitragem do duto, é tratado via duto de arbitragem (3) (5 linhas no VERSAbus e 4 no VMEbus).

Existem três opções para a implementação da lógica de arbitragem:

1. por prioridade nas linhas do duto de arbitragem;
2. por um esquema cíclico, centralizado;
3. por encadeamento em cascata, usando-se uma única linha de requisição do duto;

No duto de utilidade (4), tem-se sinais de teste de estado (apenas no VERSAbus), indicação de falhas do sistema, controle de zeramento, reinicialização etc..

O duto de inter-inteligência (5) é um sub-duto serial (2 linhas), síncrono (4MHz), utilizado para troca de pequenas mensagens entre módulos inteligentes do sistema, liberando o duto de dados. Isto constitui uma característica interessante do VERSAbus e do VMEbus, pois conforme cresce o número de processadores no sistema, o tráfego no duto também cresce, podendo levar à saturação (conforme observado em II-1).

O número de linhas utilizadas no duto, varia de acordo com o tipo de implementação.

g. *Futurebus* (IEEE-P896) {A181, Ba84, Bo82b, Bø84, Ta84a}

O Futurebus é um duto de barramento, assíncrono, desenvolvido pelo IEEE com o objetivo de estabelecer um padrão que possa satisfazer os requisitos futuros dos sistemas com múltiplos microprocessadores.

O Futurebus permite independência da arquitetura do sistema, possibilitando a implementação de sistemas multiprocessadores

verdadeiros, (máximo 32 processadores), com filosofia de controle totalmente distribuída.

São previstos mecanismos que permitem múltiplas hierarquias de memória, como "cache" e memória virtual.

Também existe apoio à sincronização de processos paralelos.

A taxa de transferência de dados permitida no duto, supera a 10 Mtransf/seg.

O duto independe do processador e do fabricante, permitindo a utilização de diversos processadores de diferentes tipos e origem, no mesmo sistema.

Também existe independência quanto à tecnologia, de modo que os avanços tecnológicos podem ser absorvidos com facilidade. Está previsto suporte adicional para métodos de detecção automática de falhas, diagnóstico (localização da falha) e reconfiguração do sistema.

O padrão proposto para o Futurebus, deverá ser suficiente para utilização por cerca de 10 anos.

O esquema de arbitragem empregado é distribuído, codificado, com a lógica de arbitragem localizada nos módulos. Existem 7 linhas de arbitragem previstas, sendo 6 para o código do mestre e 1 para paridade.

O Futurebus está logicamente dividido em dois dutos: um paralelo (P-Bus) e um serial (S-Bus) com 2 linhas do barramento. O P-Bus serve para a transação de dados entre processadores e memória (por exemplo) a uma velocidade alta; o S-Bus serve para troca de informações entre processadores, com uma taxa de transferência de dados menor, mas frequentemente com maior urgência do que dados da memória. O S-Bus serve de apoio ao sistema operacional que controla todo o sistema multiprocessador implementado no Futurebus.

O S-Bus pode ser usado para transmitir solicitação de interrupção de um processador para qualquer outro e para ativar diretamente linhas de sinal nas placas de processadores. Isso permite a implementação de funções do tipo "HALT", "RUN", etc. O Futurebus possui duas configurações:

g.1 (nível 1) dados e endereços de 32 bits, multiplexados (conector de 64 pinos);

g.2 (nível 2) totalmente compatível com nível 1, é uma extensão, com maiores recursos de controle, detecção e correção

de erros (conector de 92 pinos).

Na tabela II-1 é apresentado um quadro resumo relacionando as principais características dos dutos descritos neste item, com o objetivo de estabelecer uma visão comparativa entre eles.

CARACTERÍSTICAS DUTO	Nº DE LINHAS DO BARRAMENTO	Nº DE BITS DO PROCESSADOR		TIPO DE SÍNCRONISMO	ESQUEMA DE ARBITRAGEM	LINHAS PARA INTERRUPÇÕES	ARQUITETURAS COM VÁRIOS PROCESSADORES	SUB-DUTO DE COMUNICAÇÃO	CARACTERÍSTI- CAS ESPECIAIS
		DADOS	ENDEREÇO						
S-100	100	8/16	24	síncrono	distribuído codificado (4 linhas)	8 linhas - permite int. vetorizadas	permitido (até 17 proc)	Não disponível	permite 16 mestres tem- porários e 1 permanente
MULTIBUS	86	8/16	22	assíncrono	encadeamento em cascata ou paralelo centralizado	8 linhas - permite int. vetorizadas	permitido	Não disponível	—
STD	56	8	16 + 1 linha ex- tensão	síncrono	encadeamento em cascata	3 linhas - permite int. vetorizadas	permitido	Não disponível	—
EUROBUS	variável de acordo com a versão	8/16/32	8/16/32 + 2 linhas modificação	assíncrono	centralizado- ligação em estrela - antecipado	esquema centralizado através de arbitrador	permitido	Não disponível	facilidades para imple- mentação de arquiteturas distribuídas
ZBI	96	8/16/32	32	assíncrono	distribuído codificado (4 linhas)	9 linhas/ 3 grupos - permite int. vetorizadas	permitido (até 16 proc)	Não disponível	duto de pari- dade para dete- ção de erros
VERSAbus/ VMEbus	variável de acordo com a versão	8/16/32	23/31	assíncrono	por encadea- mento; ciclo de prioridades de nas li- nhas	7 linhas - esquema de encadeamento	permitido	duto serial para comunicação	facilidades para imple- mentação de arquiteturas distribuídas
FUTUREBUS	variável de acordo com a versão	8/16/32	32	assíncrono	distribuído codificado (6 linhas + 1 paridade)	permite int. direcionadas pelo S-Bus	permitido (até 32 proc)	possui o S-Bus para comunicação	diversas característi- cas especiais são apresenta- das

Tabela II-1 - Quadro Comparativo de Características de Alguns Dutos Padronizados

CAPÍTULO III

SOBRE O SISTEMA DISTRIBUÍDO FUNCIONAL (SDF) ASPECTOS FÍSICOS E LÓGICOS DA ESTRUTURA DO SISTEMA

1 - DESCRIÇÃO DO SISTEMA

Basicamente, o SDF é um sistema computacional destinado a execução de tarefas bastante diversificadas, abrangendo:

- edição de arquivos
- compilação de programas
- execução de programas
- depuração de programas
- monitoração de sistemas físicos
- controle de sistemas físicos
- aquisição de dados
- processamento de sinais
- etc..

A diversificação das tarefas consideradas, o grau de independência entre elas e a necessidade de processamento rápido, principalmente quando se tem problemas em tempo real com a manipulação de grandes quantidades de dados, tornam atrativa a ideia de utilização de um sistema com diversos processadores interagindo física e logicamente, através de algum tipo de interconexão.

As características básicas propostas para o SDF são:

- sistema modular
- processamento distribuído de modo funcional
- capacidade de processamento local, em cada módulo
- possibilidade de comunicação rápida entre os diversos módulos do sistema
- fácil assimiliação de avanços da tecnologia
- independência de processador

Os módulos do sistema tem capacidade de processamento local, isto é, possuem um processador, uma memória e um sistema operacional local.

A comunicação entre os módulos do SDF é feita atra-

vões de dutos, permitindo a comunicação a nível de dados e endereços e a nível de comandos de alto nível.

Os módulos do SDF podem ser:

- Módulos para tarefas gerais (ou módulos de usuários);
- Módulos para tarefas específicas (com um conjunto de tarefas (ou funções) pré-estabelecidas).

No caso de módulos de usuários, tem-se um sistema com possibilidade de:

- execução de tarefas locais
- solicitação de execução de tarefas localizadas em outros módulos
- comunicação com outros módulos, a nível de dados e de comandos.

No caso de módulos para tarefas específicas, tem-se um conjunto de tarefas (ou funções) pré-estabelecidas para o módulo. Nesse caso, tem-se a possibilidade de:

- execução de tarefas locais previamente definidas
- solicitação de execução de tarefas localizadas em outros módulos
- comunicação com outros módulos, a nível de dados e de comandos.

A existência de um conjunto de tarefas (ou funções) definidas previamente para os módulos, caracteriza a distribuição funcional do sistema.

A modularidade do sistema permite que sua implementação se faça a partir de um conjunto básico de módulos, e sejam acrescentados novos módulos, conforme necessidade e disponibilidade. Além disso, em se tratando de um sistema distribuído com módulos funcionais, cada módulo poderá ser adaptado da melhor forma para o atendimento das funções pré-estabelecidas. Isto implica, na necessidade do sistema ser desenvolvido com certa independência quanto ao processador empregado em cada módulo. Também é de importância que sejam evitadas as dependências tecnológicas, aumentando o ciclo de vida útil do sistema.

Com estas características, também a manutenção e evolução do sistema ficam favorecidas.

A tecnologia atual favorece o uso de microprocessado

res, tanto pelo aspecto econômico como pelo aspecto tecnológico.

Conforme visto em {Fa83} , o uso de microprocessadores em sistemas distribuídos, de um modo geral, é bastante atra-tivo e recomendado.

Assim, o SDF será implementado utilizando-se micro-processadores, porém, isto não deve ser considerado uma restri-ção, pois o sistema não deverá ser dependente do processador nem da tecnologia.

2 - ARQUITETURA DO SDF

Na figura III-2.1 tem-se o esquema geral da arquitetura do SDF.

Figura III-2.1 - Esquema Geral da Arquitetura do SDF

Na arquitetura proposta, tem-se módulos independentes, que podem se comunicar de duas formas distintas:

- pelo duto de dados e endereços (DDE), para a troca de informações a nível de dados (memória e periféricos)
- pelo duto de comunicação de alto nível (DCAN), para a troca de informação a nível de comandos de alto nível.

Com essa arquitetura, a idéia fundamental para o funcionamento do sistema é que, os módulos podem solicitar a ativação de tarefas ou funções, em outros módulos, através do DCAN e a troca de informações a nível de dados sempre será pelo DDE.

3 - ALGUNS MÓDULOS PARA O SDF

Alguns módulos possíveis de serem implementados para o SDF são descritos a seguir.

- a. *Módulos de Usuários*: permitem a interação entre usuários e sistema, através da execução de tarefas locais ou remotas. Esse módulo deverá conter um processador local, uma memória local, possivelmente alguma capacidade de armazenamento secundário de pequeno porte (disco flexível, por exemplo) e uma interface serial para a conexão de terminais. Na figura III-3.1 tem-se um esquema básico para a arquitetura desse módulo.

- b. *Módulo de Memória Virtual*: consiste de um módulo de memória compartilhado por todos os outros módulos do sistema. Possui um processador local, uma memória local (compartilhada com os outros módulos) e um sistema de armazenamento secundário de porte médio (disco Winchester, por exemplo). Esse módulo, com as

características descritas, dá ao SDF a capacidade de implementação de uma memória virtual, incluindo possivelmente mecanismos de "Cache", aumentando tanto o espaço de endereçamento de cada módulo, como o seu desempenho. Na figura III-3.2 tem-se um esquema básico para a arquitetura desse módulo.

Em versões simplificadas do SDF, pode-se empregar módulos de memória sem capacidade de processamento local, funcionando apenas como um meio de armazenamento sem interconexão com o DCAN (qualquer solicitação pelo DCAN exige capacidade de processamento local, no módulo).

- c. *Módulo de Disco:* consiste de um módulo com um processador local, uma memória local, um sistema de armazenamento secundário de porte médio (disco Winchester, por exemplo), compartilhado por todos os outros módulos do sistema e uma impressora. Esse módulo deverá cuidar das tarefas de manipulação de ar-

quivos, localmente, por solicitação através do DCAN. Entre as funções a serem estabelecidas, citam-se: transferências de arquivos, impressão de arquivos, gerenciamento de diretórios, gerenciamento de áreas de usuários etc.. Na figura III-3.3 tem-se um esquema básico para a arquitetura desse módulo.

- d. *Módulo de Monitoração do Sistema*: consiste basicamente de um módulo de usuário com algumas características adicionais, permitindo a monitoração do sistema. Tem-se um processador local, uma memória local, interface serial para conexão de terminais, possibilidade de utilização de unidades de disco flexível e uma memória organizada como uma lista circular, destinada ao registro das solicitações de alto nível do SDF. Esse módulo deverá monitorar as transações ocorridas pelo DCAN, detetar e se possível corrigir erros, supervisionar o estado do sistema e outras tarefas de apoio à depuração e manutenção do sistema. Na figura III-3.4 tem-se um esquema básico para a arquitetura desse módulo.

e. *Módulos para Aplicações Específicas*: diversos outros módulos, possuindo no mínimo um processador local e uma memória local, com funções ou tarefas específicas, podem ser definidos para utilização no SDF. Dentre as hipóteses possíveis tem-se:

- controladores de processos físicos
- monitores de processos físicos
- aquisição de dados em processos físicos
- processamento de imagens
- etc..

Na figura III-3.5 tem-se um esquema básico para a arquitetura desses módulos.

Figura III-3.5 - Arquitetura de Um Módulo Para Aplicações Específicas

4 - DUTOS DE INTERLIGAÇÃO DO SDF

O SDF possui, basicamente, duas estruturas de interligação dos seus módulos:

- a. o duto de dados e endereços (DDE): implementa a via de comunicação de baixo nível do sistema e compreende um duto paralelo, por onde circulam dados, endereços e sinais de controle; basicamente através desse duto ocorrem transferências de blocos de dados. No capítulo IV tem-se maiores considerações sobre o DDE.
- b. o duto de comunicação de alto nível (DCAN): implementa a via de comunicação de alto nível do sistema e compreende um duto paralelo multiplexado, proposto especialmente para implementação do SDF e objeto de consideração detalhada neste trabalho, por onde circulam solicitações de comandos de alto nível de um módulo para outro. No capítulo IV tem-se um estudo detalhado sobre a estrutura desse duto.

Cabe observar, preliminarmente, que o duto de comunicação DCAN foi escolhido como sendo paralelo e multiplexado, com o objetivo de se obter um melhor desempenho do sistema, evitando-se a necessidade de transferências seriais de informações. No caso de dutos seriais, o reconhecimento ou não de um comando, somente pode ser decidido após alguns ciclos de transferência de informações (quando todo o comando tiver sido recebido); no caso de um duto paralelo (ou paralelo multiplexado), isto não ocorre, pois a informação completa fica disponível em um único ciclo, ou possivelmente (devido à multiplexação) em poucos ciclos.

Para sistemas distribuídos cujos módulos estão fisicamente próximos, não existem dificuldades técnicas na implementação de dutos paralelos.

Com o uso de dutos paralelos eliminam-se os circuitos de conversão paralelo/serial - serial/paralelo e a lógica de repetição automática de informações, devida a erros de transmissão.

5 - SOBRE O SOFTWARE DE CONTROLE DO SDF

O software de controle dos sistemas computacionais (o sistema operacional) é organizado geralmente, segundo uma estrutura de camadas interligadas logicamente, onde há a criação de máquinas virtuais que facilitam a utilização do sistema e melhoram o aproveitamento dos recursos disponíveis. Na figura III-5.1 tem-se um esquema para essa organização.

Figura III-5.1 - Organização de Um Sistema Operacional

No caso de sistemas distribuídos, com vários módulos com processadores, pode-se manter essa estrutura para os sistemas operacionais mas deve-se acrescentar uma estrutura de apoio que permita a comunicação entre os módulos do sistema.

Nas redes de computadores, por exemplo, cada um dos nós (correspondente aos módulos aqui referenciados) possui uma estrutura de software adicional que permite a interligação e a comunicação entre dois nós da rede, organizada em camadas funcionais [Ry81]. Nesse caso, a comunicação entre os nós requer a im

plementação de protocolos em diversos níveis, relacionando as diversas camadas desse software de controle. A interligação física entre os diversos nós é, normalmente, uma linha serial (um cabo coaxial, por exemplo) ou um meio de telecomunicação (com o envio serial da informação) para os casos de maior porte.

No caso do SDF, cada um dos módulos possui um sistema operacional local, independente e de complexidade variada de acordo com o tipo de aplicações definidas para o módulo. O mecanismo lógico empregado para a interligação e comunicação entre dois módulos do sistema difere daquele empregado nas redes de computadores, pois as camadas funcionais não são empregadas. Nesse caso é possível, por exemplo, que solicitações de usuários ativem a geração de solicitações pelo duto de alto nível quase que diretamente.

Na figura III-5.2 é apresentado um diagrama da organização do software e a comunicação existente entre os módulos no SDF.

Figura III-5.2 - Organização do SO_p para o SDF

O protocolo de comunicação entre os módulos envolve:

- a. *Comunicação de Alto Nível*: protocolo implementado através de comandos de alto nível e sinais de controle;
- b. *Comunicação de Baixo Nível*: envolvendo acessos a memória, podendo incluir transferências de blocos de dados (por acesso direto à memória - ADM) e sinais de controle.

A utilização do DCAN ocorre sempre que é necessária a solicitação de tarefas em outros módulos, através de comandos específicos. A geração e a recepção desses comandos é efetuada por rotinas incorporadas ao sistema operacional local.

No caso de recepção, tem-se rotinas que atendem a uma interrupção gerada na interface do DCAN. Dependendo do comando recebido o atendimento é imediato; em outros casos utiliza-se uma estrutura de fila de atendimento.

No caso de geração dos comandos, a necessidade do comando ocorre durante a execução de alguma tarefa (programa de usuário, rotina do sistema operacional ou alguma rotina previamente definida no módulo). Nesse caso as rotinas adequadas do sistema operacional são solicitadas e o comando será transmitido pelo DCAN, a um módulo específico.

No capítulo IV são apresentados alguns detalhes sobre a geração, recepção e atendimento dos comandos de alto nível.

CAPÍTULO IV

SOBRE A COMUNICAÇÃO NO SDF

1 - ASPECTOS DO MECANISMO DE COMUNICAÇÃO ENTRE MÓDULOS DO SDF

1.1 - Considerações Iniciais

Como descrito no capítulo III, o SDF possui dois dutos de interligação entre módulos; um para a comunicação de alto nível (DCAN), outro para a comunicação de baixo nível (DDE).

Um ponto fundamental no desenvolvimento do sistema, é a definição de um mecanismo que permita a comunicação entre os diversos módulos que o compõem.

Nesta seção, é proposto um mecanismo de comunicação, que permite a troca de mensagens entre os diversos módulos do SDF, utilizando os dois dutos disponíveis.

Basicamente, no esquema que será apresentado, assume-se que em memórias não pertencentes ao próprio módulo, apenas é permitida a execução de leituras (sendo proibido escritas). Assim, a integridade das informações contidas na memória de cada módulo não é ameaçada por escritas indevidas, de outros módulos.

Assume-se também que a necessidade de um envio de mensagem de A para B ou a necessidade de recepção de mensagem em A, vinda de B, é informada através do DCAN; as mensagens, porém, circulam pelo DDE sempre através de leituras e transferências de blocos de dados (se for o caso).

Na figura IV-1.1 tem-se um esquema básico para o mecanismo de comunicação do SDF, considerando-se dois módulos.

1.2 - Elementos Básicos Para a Comunicação

Como sugerido na figura IV-1.1 (blocos pontilhados), na troca de mensagem deve-se ter pelo menos dois elementos básicos: um que contenha a mensagem a ser lida pelo módulo (ou enviada para o módulo) que recebe mensagem, localizado no módulo que gera mensagem; outro, localizado no módulo que recebe mensagem, destinado a armazenar a mensagem recebida (lida). Esses elementos constituem a base do mecanismo proposto e podem ser algum tipo de estrutura de dados, capaz de armazenar linhas de informações, em um formato pré-estabelecido. São os "escaninhos de mensagens" (EM) e podem ser organizados de diversas maneiras.

Uma possível implementação para um EM é através do uso de listas de registros, onde um registro pode conter:

- a. um campo indicando se a mensagem contida nessa posição está disponível para leitura ou se já foi lida por algum outro módulo;

- b. um campo de proteção, com o objetivo de se evitar acessos não autorizados à mensagem;
- c. um campo de mensagem, contendo a mensagem propriamente dita.

Na figura IV-1.2 tem-se o esquema de uma opção para a organização e implementação de um EM. Nesse caso utiliza-se uma matriz, porém a idéia pode facilmente ser adaptada para o uso de outras estruturas de dados.

No esquema da figura IV-1.2 cada linha da matriz EM_i , corresponde a um registro. A referência a um EM_i se faz pelo seu endereço base i; para o acesso ao k-ésimo registro do EM_i , utiliza-se:

$$\text{endereço do registro} = \text{endereço base } i + k$$

Os registros podem ser utilizados sempre que estiverem vazios ou quando a mensagem contida já foi lida por algum outro módulo.

O tamanho de um registro (implicitamente o tamanho de

uma mensagem) e o número de registros existentes em um determinado EM, devem ser especificados por ocasião da sua criação.

1.3 - Utilização de Um EM

Um EM pode ser utilizado de três formas distintas:

- a. para conter mensagens geradas no próprio módulo a que pertence (EM tipo a);
- b. para conter mensagens lidas em outro módulo e copiadas no módulo a que pertence (EM tipo b);
- c. para conter mensagens dos dois tipos anteriores (EM tipo c).

Assim, ficam definidas três classes de EM, uma para cada caso descrito.

Fica evidente a característica de transação não destrutiva, tendo-se apenas leituras de um módulo em outro.

Para uma troca de mensagem é necessário pelo menos um EM em cada módulo envolvido. Na figura IV-1.3 tem-se esquemas possíveis de serem implementados, com indicação do fluxo de mensagens possíveis.

Na definição dos módulos do SDF, pode-se adotar:

- a. cada módulo apresenta obrigatoriamente, pelo menos um EM tipo a e um tipo b (caso a da figura IV-1.3);
- b. cada módulo apresenta obrigatoriamente, pelo menos um EM tipo c (caso b da figura IV-1.3);
- c. cada módulo apresenta obrigatoriamente, um determinado número de EM de tipos a, b e c.

Em qualquer caso adotado, deve-se prever a possibilidade de criação de novos EM, dinamicamente, conforme necessidade detectada no decorrer do processamento das tarefas.

Pode-se, portanto, adotar inicialmente apenas EM tipos a e b e, com subseqüentes definições de novos EM, ter-se um caso semelhante ao de se adotar inicialmente apenas EM tipo c ou mesmo tipos a, b e c.

A adoção inicial de EM tipos a e b, pode ser mais adequada ao SDF considerando-se que nesse caso as mensagens geradas no próprio módulo e as recebidas de outros módulos, são arma

a.) Utilização de EM classes a e b

b.) Utilização de EM classe c

Figura IV-1.3 - Esquema Para Utilização dos EMs

zenadas em estruturas distintas. Isso pode facilitar o mecanismo de utilização das mensagens, bem como aumentar a confiabilidade do sistema.

Qualquer que seja a hipótese adotada, deve-se ter em cada módulo uma tabela de acesso aos EM. A tabela deve conter os ponteiros que referenciam cada um dos EM implementados (incluindo-se os definidos dinamicamente). Na figura IV-1.4 tem-se um diagrama mostrando uma possível organização.

Os EM pré-definidos para os módulos são criados por ocasião da ativação do sistema operacional de cada módulo, tendo suas características pré-estabelecidas na geração do sistema.

Os EM definidos dinamicamente requerem para sua criação, um suporte adicional, pois como mostrado na figura IV-1.3 quando dois módulos A e B trocam mensagens entre si, há uma correspondência entre as suas estruturas de EM envolvidas. Assim,

quando há a criação de um EMx em um módulo A, visando uma troca de mensagens com um módulo B, deverá obrigatoriamente ser criado em B um EMx'. Isso requer um comando de alto nível pelo DCAN, solicitando a criação do EMx'.

Tem-se a seguinte relação na criação dos EM:

<u>Módulo A</u>	<u>Módulo B</u>
(requisitante)	(requisitado)
EM criado por decisão local	EM criado por solicitação via DCAN
tipo a	tipo b
tipo b	tipo a
tipo c	tipo c

1.4 - *Funcionamento do Mecanismo Proposto*

O mecanismo de comunicação proposto requer, para o seu funcionamento, a definição de alguns comandos de alto nível no DCAN. Assim, são necessários os seguintes comandos:

- a. comando para solicitar a criação de um EM;
- b. comando para informar a criação de um EM;
- c. comando para solicitar a desativação de um EM;
- d. comando para solicitar envio de mensagem através de um EM;
- e. comando para informar o envio de mensagem através de um EM.

Outros comandos podem ser implementados, porém o conjunto apresentado é suficiente para o funcionamento do mecanismo.

O funcionamento básico do mecanismo de comunicação é apresentado a seguir.

Envio de Mensagem de um Módulo A para um Módulo B.
Nesse caso, tem-se:

- a mensagem gerada em A é colocada no EMx;
- através do DCAN envia-se um comando do tipo e, informando o EM onde a mensagem foi colocada;
- no módulo B, quando for possível receber a mensagem, faz-se

uma conexão com A, através do DDE, efetua-se a leitura da mensagem no EMx e a posterior escrita no EMx'.

Quando o EM a ser utilizado na troca de mensagens não está criado, o módulo A (requisitante) inicialmente solicita a sua criação no módulo B, segundo o que foi descrito em IV-1.3. Nesse caso, tem-se:

- através do DCAN o módulo A envia um comando do tipo a, associando um código ao EM solicitado;
- quando o módulo B criar o EM solicitado, este envia ao módulo A um comando do tipo b, informando a criação do EM e dando o código de referência do EM em B.

Quando um EM criado dinamicamente não é mais necessário, o módulo que o solicitou pode determinar sua desativação, enviando pelo DCAN um comando do tipo c, com a informação do código do EM.

Pode-se implementar também um sistema de proteção para esse mecanismo, procurando melhor coordenar a utilização dos escaninhos de mensagens.

Cabe ser observado também que, as idéias fundamentais empregadas na comunicação entre os módulos do SDF, podem ser aplicadas à comunicação entre tarefas, dentro de um mesmo módulo.

Nesse caso, podem-se utilizar estruturas de EM semelhantes às empregadas na comunicação entre os módulos, facilitando futuras migrações de tarefas de um módulo para outro. Esta compatibilidade é altamente desejável, principalmente em sistemas em desenvolvimento.

1.5 - Aspectos Relevantes do Mecanismo Proposto

Nas redes de computadores, normalmente utiliza-se comunicação serial, com comandos e mensagens circulando pelo mesmo duto.

No caso do SDF o mecanismo proposto utiliza comunicação paralela, com os comandos de alto nível circulando pelo DCAN

e as mensagens circulando pelo DDE, através de transferência de blocos de dados. Nesse caso tem-se os seguintes aspectos favoráveis:

- a. um maior volume de comandos e mensagens circulando no sistema;
- b. o reconhecimento de endereçamento de módulos e dos comandos recebidos por um módulo é facilitado;
- c. as transferências de mensagens são mais rápidas;
- d. o mecanismo de apenas leitura favorece a proteção do sistema.

Por outro lado, no SDF tem-se um número significativamente maior de linhas de comunicação, pois, nos mecanismos seriais tem-se apenas um par de linhas. Isto pode acarretar alguns problemas na comunicação, como visto em II-2, se as distâncias entre módulos forem grandes.

No caso do SDF, o sistema deve possuir os módulos conectados por ligações de pequena distância, o que vem minimizar os problemas de ruídos e interferências.

Resta, porém, o aumento do custo de implementação, devido ao maior número de linhas e o conseqüente aumento dos circuitos de terminação, acionadores, acopladores e conectores.

No caso de transferência de arquivos, pode-se ter um número elevado de blocos de dados para serem transferidos, o que inviabiliza o emprego do mecanismo proposto para mensagens (poucos blocos de dados). Nesse caso pode-se optar pelo uso de transferência por acesso direto à memória (ADM), sempre empregando-se os EM como descrito aqui.

2 - O DUTO DE COMUNICAÇÃO DE ALTO NÍVEL (DCAN)

2.1 - Caracterização do DCAN

O Duto de Comunicação de Alto Nível do SDF, o DCAN, é uma via de comunicação que suporta os comandos de alto nível, implementados no sistema para a comunicação entre módulos.

O objetivo da definição desse duto é o de estabelecer uma via de comunicação rápida, entre os diversos módulos do sistema independente do duto de dados e endereços, normalmente e existente, permitindo o envio de comandos de alto nível que acionam tarefas pré-estabelecidas em cada um dos módulos. Esse acionamento de tarefas pode ser obtido pelo uso de interrupções geradas na interface do módulo. Uma análise das possibilidades existentes é apresentada em IV-4. Cabe, porém, destacar aqui que o esquema estabelecido pela introdução do DCAN no SDF, deve permitir um rápido reconhecimento dos comandos válidos e do endereçamento do módulo, para se obter uma característica de alta velocidade na comunicação entre os módulos.

2.2. - Formato de Um Comando de Alto Nível

Basicamente, o DCAN deve permitir comandos para:

- a. manipulação de arquivos;
 - b. troca de mensagens;
 - c. criação e desativação de escaninhos de mensagens (EM);
 - d. ativação e desativação de tarefas;
 - e. gerenciamento de áreas de usuários;
 - f. informação de falhas no sistema
- etc..

Um comando de alto nível no SDF, tem o formato mostrado na figura IV-2.1.

O código do comando permite a sua identificação e o estabelecimento de um mecanismo de reconhecimento, nos módulos previamente preparados para o atendimento de cada comando. O número de bits (n) necessários depende do número de comandos definidos.

O endereço do módulo destino é utilizado para definir qual módulo do sistema está sendo requisitado, para o aten

dimento do comando. O número de bits (k) necessários depende de quantos módulos o sistema possui.

O ponteiro para informações adicionais é utilizado para indicar, ao módulo destino, em qual registro de um EM pré-estabelecido, no módulo fonte, são encontradas as informações adicionais necessárias ao atendimento do comando enviado. Como exemplo, considere-se um comando de impressão do tipo:

Imprima arquivo x

Nesse caso, o código do comando se refere a "Imprima", e as informações adicionais à identificação do arquivo (nome, tipo, tamanho etc..).

O endereço do módulo fonte é utilizado para identificar, para o módulo destino, o autor da solicitação. Basicamente tem-se duas finalidades com esse endereçamento:

- permitir que o módulo destino possa obter as informações adicionais no módulo fonte, utilizando o ponteiro fornecido no comando e o mecanismo de comunicação através do DDE, estabelecido para o SDF (IV-1).

- b. permitir que o módulo destino possa enviar mensagens, quando necessário, relatando ocorrências relacionadas com o comando solicitado, para o módulo fonte.

Na figura IV-2.2 tem-se um esquema mostrando o fluxo da informação no envio de um comando pelo DCAN e na obtenção das informações adicionais, para o atendimento.

Com relação ao endereçamento do módulo destino, cabe observar que podem ocorrer situações em que um comando específico precise ser enviado a todos os módulos do sistema; isto pode ser implementado, reservando-se um código de endereço para es

se modo de comunicação (por exemplo, o endereço zero).

2.3 - *Definição e Organização das Linhas do DCAN*

A primeira questão a ser analisada na definição do DCAN, é a escolha entre um duto serial e um duto paralelo. Inicialmente consideram-se:

- a. o objetivo descrito em III-2.1 para o DCAN;
- b. que os comandos enviados pelo DCAN, são destinados a um módulo específico do sistema (a menos de casos especiais);
- c. que um módulo do sistema somente deve considerar as informações do DCAN, quando tiver sido endereçado;
- d. que em um duto serial, o reconhecimento do endereçamento ou do comando, tem de ser feito após o recebimento de todos os bits envolvidos, o que pode ocasionar um certo atraso na decisão;

Com base nessas considerações iniciais, a escolha de um duto paralelo é a mais indicada pois nesse caso tem-se:

- a. o código do comando ou o endereçamento pode ser enviado de uma só vez;
- b. o reconhecimento de um comando ou endereço pode ser feito de modo bastante rápido, por um circuito de interface paralela;
- c. no caso de ocorrência de erros na transmissão ou recepção das informações, que exijam sua retransmissão, tem-se um esquema mais simples e mais rápido (normalmente esse esquema é dispensado no caso de dutos paralelos, pois raramente ocorrem erros de transmissão temporários).

Sabe-se porém, que a opção por um duto paralelo traz, fatalmente, um aumento indesejável do custo do sistema, devido ao aumento no número de linhas com o correspondente aumento no número de circuitos de terminação, acionadores, acopladores, conectores, etc.. Como alternativa pode-se utilizar um duto paralelo, mas multiplexado, que mantém o problema do número de linhas do duto sob controle, apesar de introduzir certa perda na velocidade de transmissão das informações.

A proposta apresentada aqui para o DCAN é, portanto,

a de um duto paralelo multiplexado.

Para o envio das informações contidas na definição de um comando de alto nível do SDF, deve-se estabelecer alguns ciclos distintos, enviando-se em cada um parte das informações.

No caso do DCAN, tem-se a sugestão de utilização de dois ciclos, com:

1º ciclo → <código do comando><endereço destino>

2º ciclo → <ponteiro p/inf.adic><endereço fonte>

Assim, no 1º ciclo de envio de informações, tem-se o acionamento dos circuitos de reconhecimento do endereçamento e do comando.

Entre o 1º e o 2º deverá ocorrer um sinal de recebimento de informação, enviado pelo módulo destino para o módulo fonte (caracterizando um procedimento assíncrono). Aqui inclui-se também a possibilidade de sinais de erros de paridade, por exemplo.

No caso de não envio do sinal de recebimento (devido a inexistência do módulo destino, por exemplo), deve existir uma limitação por tempo de conexão, após o qual o módulo fonte cancela o envio do comando.

Também após o 2º ciclo, tem-se considerações semelhantes às apresentadas para o 1º.

Esses pontos serão considerados mais detalhadamente em IV-3.

Descrição dos Sub-Dutos do DCAN:

- a. *Sub-Duto D1:* composto por "N" linhas que servem para a codificação dos comandos solicitados no 1º ciclo e para o ponteiro para as informações adicionais no 2º ciclo.
- b. *Sub-Duto D2:* composto por "k" linhas que servem para a codificação do endereço do módulo destino no 1º ciclo e para o endereço do módulo fonte no 2º ciclo.
- c. *Sub-Duto D3:* composto por "M" linhas que se destinam à implementação do processo de arbitragem para os módulos do SDF.
- d. *Sub-Duto D4:* composto por 7 linhas que se destinam aos sinais de controle:
 - linha de disponibilidade do DCAN

- linha de validação dos sinais do DCAN
- linha de reconhecimento dos sinais
- linha de paridade
- linha de erro de paridade
- linha de falha
- linha de arbitragem

e. *Sub-Duto D5*: composto por "X" linhas que se destinam à expansão futura do DCAN.

O número de linhas do DCAN, depende diretamente do número de módulos e de comandos a serem implementados. Assim, N, k, M, e X devem ser definidos em função do que se pretende implementar, evitando-se um super-dimensionamento, com o conseqüente aumento de custo.

Na figura IV-2.3 tem-se um esquema básico para a organização das linhas do DCAN.

2.4 - Funcionamento do DCAN

Para o estudo do funcionamento do DCAN, cabe definir o tipo de sincronismo empregado. Com base no estudo bibliográfico desenvolvido, observa-se que os dutos assíncronos tem sido uma tendência moderna, principalmente devido à facilidade na obtenção de independência de processador e de tecnologia. Como exemplo tem-se o Futurebus {A181,Bo84}, projetado totalmente assíncrono, visando alcançar essas características.

Assim, para o DCAN, adota-se também um esquema assíncrono, que será descrito neste item.

Antes da descrição do funcionamento do DCAN, apresentam-se as finalidades de cada uma das linhas de controle do duto:

a. *Linha de Disponibilidade (LD)*

Indica quando o DCAN está disponível para ser acessado pelo mestre, que já foi selecionado no esquema de arbitragem para a utilização do duto.

b. *Linha de Validação (LV)*

Indica o momento a partir do qual os sinais do DCAN estão válidos. É acionada pelo módulo mestre do duto.

c. *Linha de Reconhecimento (LR)*

Indica que as informações transmitidas no DCAN foram recebidas com sucesso, pelo módulo destino.

d. *Linha de Paridade (LP)*

Indica o bit de paridade a ser utilizado para a detecção de erro na transmissão.

e. *Linha de Erro de Paridade (LEP)*

Indica, quando ativa, que o módulo destino detetou um erro de paridade na informação recebida no DCAN.

f. *Linha de Falha (LF)*

Indica, quando ativa, que uma falha grave foi detetada em algum ponto do sistema. Deverá ativar uma rotina de atendimento de emergência em todos os módulos do sistema. O módulo de monitoração do sistema se encarrega da ativação dessa linha.

g. *Linha de Arbitragem (LA)*

Indica se o processo de arbitragem está autorizado.

Conhecendo-se basicamente a finalidade das linhas do DCAN, pode-se definir o seu esquema de funcionamento.

Monitoração das Linhas do DCAN

Os diversos módulos do sistema monitoram permanentemente as linhas do DCAN, através de suas interfaces dedicadas ao DCAN.

Quando há um reconhecimento do endereçamento nas linhas do sub-duto D2 e do comando solicitado no sub-duto D1, gera-se no módulo de destino um sinal de interrupção, para o acionamento do mecanismo de recebimento das informações contidas na interface e também, quando nenhum erro adicional for detetado, o sinal de reconhecimento para o módulo fonte (mantido na linha por um período pré-estabelecido). O efeito desse sinal de reconhecimento é informar o sucesso da transmissão e liberar o módulo fonte para completar as informações necessárias no duto (quando for o caso) ou liberar o duto para que um novo módulo passe a ser o mestre.

Pode-se ter as seguintes situações após o envio de informações no DCAN:

<i>LR</i>	<i>LEP</i>	<i>estado do módulo fonte</i>
Ø	Ø	aguardando
1	Ø	reconhecimento efetuado com sucesso
Ø	1	erro de paridade
1	1	não ocorre

Cabe observar que "Ø" indica um estado não ativo da linha e "1" um estado ativo.

O recebimento de um sinal de erro de paridade gera no módulo fonte (o mestre do DCAN) um conjunto de ações que podem:

- recolocar as informações no DCAN e repetir o seu envio (pouco recomendado no caso do DCAN, por ser um duto paralelo);
- interromper a comunicação, gerando uma mensagem de erro ã

tarefa solicitando a transação.

Pode-se ter um esquema que repita o caso a., um certo número de vezes antes de executar o caso b.

As Fases na Conexão Entre Dois Módulos no DCAN

Definem-se três fases distintas na conexão entre dois módulos no DCAN:

a. *A Arbitragem do DCAN (1a. fase)*

Quando um módulo do SDF precisa utilizar o DCAN, este deve entrar no processo de arbitragem do duto. Nesse processo, que define a 1a. fase da transação no duto, apenas um dos módulos do SDF é autorizado a acessar o DCAN.

b. *Transmissão do 1º Grupo de Informações (2a. fase)*

Após um módulo ter obtido o controle do DCAN, através do processo de arbitragem, este aguarda a liberação do duto, indicada pela linha LD. Quando disponível, o duto é acessado, sua disponibilidade é alterada e ocorre a transmissão do 1º ciclo do comando de alto nível considerado (código do comando e endereço de destino) e os sinais de controle necessários (validação e paridade).

Após a transmissão, o módulo fonte aguarda o sinal de reconhecimento ou o sinal de erro de paridade.

b.1. Se esses sinais não são enviados após um certo tempo (definido na implementação do sistema), a transação é encerrada.

b.2. Se o sinal de erro de paridade é acionado, pode-se repetir a transmissão um certo número de vezes e em caso de permanência do erro, encerrar a transação com a geração de uma mensagem apropriada, ou simplesmente encerrar a transação com a geração de mensagem.

b.3. Se o sinal de reconhecimento é acionado, pode-se passar para a 3a. fase.

Também em b.1, pode ser adotada uma política semelhante à descrita em b.2. A decisão de como implementar esses casos dependerá do grau de complexidade que se queira dar ao sistema e da fase de desenvolvimento em que se encontra (versão inicial, preferivelmente mais simples ou versões avançadas).

das, com um certo grau de complexidade maior).

No caso b.1, deve-se ter também um mecanismo que garanta a integridade da transação, isto é, não pode ocorrer do módulo fonte interpretar um não reconhecimento exatamente quando o módulo de destino consegue reconhecer a informação. Uma sugestão é a de primeiro o módulo fonte invalidar a informação para depois interpretar o não reconhecimento. Em caso de dúvida, revalidar a informação e verificar o reconhecimento.

c. *Transmissão do 2º Grupo de Informações (3a. fase)*

Quando a 2a. fase é encerrada com sucesso, tem-se a 3a. fase, onde são transmitidos o 2º ciclo do comando de alto nível (ponteiro para as informações adicionais e endereço do módulo fonte) e os sinais de controle necessários (validação e paridade).

Após a transmissão tem-se um procedimento análogo ao da 2a. fase, sendo que no caso de ocorrência de erros definitivos, deve-se gerar mensagens apropriadas no módulo fonte.

No módulo de destino, devem-se tomar providências adequadas para manter a integridade das informações, no caso de erros durante a transação via DCAN.

Na figura IV-2.4 tem-se um diagrama, ilustrando as fases definidas.

Superposição das Fases de Conexão

Uma análise detalhada das fases de conexão definidas mostra que a 1a. fase, a arbitragem, é totalmente independente das demais. Pode-se, portanto, implementá-la antecipadamente fazendo-se uma superposição no tempo ("pipeline"), como sugerido no diagrama mostrado na figura IV-2.5 .

A superposição sugerida leva em consideração a possibilidade de ocorrência de algum problema na 2a. fase da transação anterior no DCAN, antecipando a liberação do duto. Assim, nesse caso, o DCAN será utilizado mais adequadamente.

O Mecanismo de Arbitragem

Para a implementação do mecanismo de arbitragem pode-se adotar um dos esquemas que vem sendo utilizados nos dutos padronizados. Como visto no estudo bibliográfico apresentado, podem-se ter três tipos básicos de esquemas:

- a. distribuído
- b. centralizado
- c. híbrido

No caso do DCAN, a proposta é a de utilização de um esquema distribuído, com o objetivo de se minimizar o número de linhas necessárias no duto e a possibilidade de problemas glo-

bais, que podem ocorrer no caso de falha dos esquemas centralizados e híbridos.

Um possível esquema a ser empregado é o apresentado na figura II-2.5 que consta de um circuito simples e eficiente {Ta76,Ta82,Ta84a}.

Com relação ao código de prioridade utilizado pelo módulo na arbitragem do DCAN, pode-se empregar o endereço do módulo dentro do SDF. Isso impõem que módulos com endereços mais altos (por exemplo) tenham maior prioridade no uso do DCAN. Em certas situações pode não ser o mais adequado. Uma alternativa é o emprego de um código que agrupe duas informações básicas:

- a. o tipo de utilização que se fará no DCAN;
- b. um código de identificação do módulo.

Nesse caso pode-se formar o código de prioridade com a concatenação do código da classe a que pertence o comando a ser utilizado (IV-4) e o código de endereço do módulo. Assim, tem-se o código de prioridade esquematizado na figura IV-2.6 .

Com esse tipo de código tem-se um esquema dinâmico de atribuição de prioridade no uso do DCAN, pois apenas o código de endereço do módulo é fixado. A classe do comando é definida em função do processamento corrente no módulo.

No caso de vários módulos estarem solicitando o uso do DCAN, para uma mesma classe de comandos, a decisão é tomada em função do endereço do módulo, porém sempre o acesso ao DCAN será permitido a um módulo que o tenha requisitado para o uso da classe de comandos de mais alta prioridade.

A definição do número de linhas do sub-duto D3, depende da implementação do sistema (número de módulos, número de classes de comandos, etc..) .

Na figura IV-2.7, é apresentado um diagrama de tempos mostrando uma transação correta entre dois módulos do SDF através do DCAN.

Figura IV-2.7 - Diagrama de Tempos Para Uma Transação no DCAN

3 - ASPECTOS FÍSICOS DA ARQUITETURA DE CONTROLE DO DCAN

3.1 - Elementos Básicos

A utilização do DCAN ocorre através de um sistema de interface, que interliga o duto a um módulo do sistema. Distinguem-se três partes básicas nesse sistema:

- a. *Sistema de arbitragem*: faz o interfaceamento do módulo com as linhas do sub-duto D3;
- b. *Sistema de escrita no DCAN*: permite que um módulo possa acessar o duto para fazer uma solicitação de comando. Nesse caso, como visto em IV-2, são utilizados os sub-ductos D1 e D2.
- c. *O sistema de reconhecimento e recepção do comando*: faz o reconhecimento do endereçamento do módulo, do comando solicitado e desempenha as funções necessárias para a preparação do processo de atendimento da solicitação recebida. Para esse caso, são utilizados os sub-ductos D1 e D2.

Em cada um dos casos descritos, utilizam-se alguns sinais de controle (para recebimento de informações ou para a geração de novas informações) do sub-duto D4.

3.2 - Princípios de Funcionamento do Sistema de Arbitragem

Em IV-2.4 foi apresentado o esquema a ser empregado na arbitragem do DCAN (esquema distribuído - Fig.II-2.5) e um diagrama de tempos (Fig. IV-2.7), onde pode ser observado os instantes em que o processo de arbitragem é iniciado e é encerrado.

O esquema apresentado na figura IV-3.1 segue o diagrama de tempos da figura IV-2.7 e mostra o relacionamento entre o processador de um módulo do SDF, o circuito de arbitragem e o sub-duto de arbitragem D3 do DCAN.

Funcionamento do Esquema

- a. *Requisição do DCAN*: o processador do módulo requisita o uso do DCAN através da linha de requisição local, fornece ao circuito de arbitragem a classe do comando considerado e aguarda pelo sinal SA, que indica quando o módulo poderá acessar o DCAN.

- b. *A arbitragem*: o processo de arbitragem somente é iniciado quando a linha LA estiver desativada. Quando o módulo detecta que é o vencedor do processo de arbitragem (RA em nível alto, por exemplo), o processador é avisado do sucesso (SA) e a linha LA tem o seu estado alterado (inibe a arbitragem).
- c. *Liberação de D3*: quando o módulo inicia a utilização do DCAN, o circuito que cuida da "escrita" no DCAN, gera o sinal SLA, que deverá fazer com que o circuito de controle, libere a arbitragem, desativando a linha LA.

3.3 - Procedimentos Para o Envio de Comandos

Após um módulo ter tido sucesso no processo de arbitragem, este deverá utilizar o DCAN para o envio do comando necessário. Para isso o módulo deve acessar as linhas do DCAN (sub-dutos D1 e D2) para o envio das informações, sob o controle das linhas do sub-duto D4.

No diagrama da figura IV-2.7 tem-se um esquema mostrando uma transação no DCAN, onde pode-se observar o relacionamento das linhas de controle de D4, com o envio das informações em D1 e D2.

A colocação das informações no duto é efetuada por um sistema de interface. Os códigos de comandos, ponteiros e endereços, fornecidos pelo processador, são escritos na interface por meio de algum recurso apropriado do processador. Por exemplo, pode-se utilizar uma escrita em periférico (do tipo OUTn) ou uma escrita em memória, conforme o tipo de endereçamento adotado (ou disponível) na arquitetura interna do módulo.

Na figura IV-3.2 é apresentado um esquema básico para o sistema de interface capaz de desempenhar as funções necessárias à escrita no DCAN, incluindo a geração e monitoração dos sinais de controle adequados.

Funcionamento do Esquema

- a. *Posse do DCAN*: após o término da arbitragem, o processador aguarda pelo sinal de disponibilidade do duto, (linha LD). O circuito de controle de disponibilidade recebe a linha LD do sub-duto D4 e o sinal de requisição do duto (RD) vindo do processador (gerado a partir do sinal SA, do sistema de arbitragem). Quando o duto está disponível e há requisição do processador, é gerado um sinal para o processador indicando a posse do duto e faz-se um grampeamento da linha LD indicando que o DCAN está ocupado.
- b. *Informações para os sub-dutos D1 e D2*: essas informações são geradas pelo processador e colocadas nos registradores correspondentes aos sub-dutos D1 e D2. O acoplamento com o DCAN ocorre através de circuitos acopladores e as informações enviadas seguem as fases já descritas em IV-2.4.

c. *Os sinais de controle utilizados na interface*

- c.1. *signal de disponibilidade LD*: (já mencionado em a.)
- c.2. *signal de paridade LP*: gerado no circuito de paridade e colocado no DCAN através de uma habilitação acionada por SAC. A geração de paridade utiliza as informações que se se guem pelos sub-dutos D1 e D2.
- c.3. *signal de validação LV*: gerado no circuito de validação das informações enviadas por D1 e D2. A partir do instante em que ocorre o sinal de sincronismo para o acoplamento ao DCAN (gerado pelo processador), introduz um atraso (tempo suficiente para a estabilização das linhas) e então é gerado o sinal de validação, que é mantido no duto enquanto durar a conexão entre os módulos (até ocorrer um erro ou reconhecimento da informação).
- c.4. *signal de erro de paridade*: o circuito de detecção de erro de paridade monitora a linha LEP do DCAN, para receber a informação de possível erro de paridade, detectado no modulo destino. Quando um sinal é detetado na linha LEP, o circuito gera uma informação para o processador, para que este tome as providências previstas.
- c.5. *signal de reconhecimento*: o circuito de reconhecimento neste esquema, monitora a linha LR do DCAN e quando um sinal de reconhecimento é detetado, gera uma informação para o processador.
- d. *O circuito de controle de tempo*: monitora o tempo entre uma conexão (a partir do sinal de validação) e o recebimento de um sinal de reconhecimento ou erro. O sinal de validação, ativa o circuito de controle de tempo. Se o tempo máximo é atingido sem o recebimento do sinal de reconhecimento (ou erro), os sinais de D1 e D2 são invalidados e o processador é avisado do esgotamento de tempo através do sinal STE, devendo tomar as providências pré-estabelecidas.
- e. *Controle de acionamento da interface*: quando acaba uma transação no DCAN, ocorre a liberação do duto (todos os acopladores são desativados e a linha LD, antes grampeada em nível baixo, por exemplo, é liberada.
Entre a 2a. fase e a 3a. fase de uma transação, há a desativaa

ção dos acopladores, porém LD continua grampeada, pois o módulo continuará a usar o DCAN.

3.4 - Procedimentos Para Reconhecimento e Recepção de Comandos

Os módulos do SDF monitoram os sub-ductos D1 e D2 e algumas linhas de controle do DCAN, verificando o endereço do módulo destino e o código do comando solicitado, através do sistema de interface de reconhecimento e recepção de comandos.

Na figura IV-3.3 tem-se um esquema básico para essa interface, que funciona segundo as fases de uma transação no DCAN.

Funcionamento do Esquema:

- a. *Durante a 1a. fase (arbitragem):* não tem participação.
- b. *Durante a 2a. fase (código do comando e endereço do destino no DCAN)*
 - b.1. *Monitoração e Reconhecimento:* o reconhecimento do endereçamento e do código do comando, recebidos pelo DCAN, é efetuado pelo sistema de monitoração e reconhecimento, que recebe os sinais de D1 e D2, o sinal de validação dos dados do DCAN e o sinal de verificação da paridade (fornecido pelo circuito de verificação de paridade, da interface do módulo).
Quando um módulo reconhece estar sendo endereçado e o comando solicitado pertence ao conjunto de comandos pré-estebelecidos para o módulo, é gerado o sinal de reconhecimento do DCAN e uma interrupção para o processador. A interrupção pode ter diversas finalidades distintas, conforme o tipo de implementação adotada para a recepção dos dados do DCAN, o atendimento do comando ou o tipo de comando recebido.
 - b.2. *Recepção de Dados:* após o reconhecimento das informações do DCAN, deve-se registrar o dado disponível em D1 para posterior utilização no módulo. Esses dados registrados podem, posteriormente, ser enviados diretamente ao processador, ou para algum tipo de estrutura de dados implementada por software ou por hardware. O que define o tipo de manipulação dos dados recebidos, é o esquema de atendimento do comando. Independentemente disso os dados devem ser registrados, nesta fase.
 - b.3. *Verificação de Paridade:* no caso de ocorrência de erro de paridade nesta fase, a interface gera o sinal de erro correspondente que é enviado ao módulo fonte pela linha LEP do DCAN. Internamente, no módulo destino, não ocorrem maiores consequências, pois o erro de paridade inibe o sistema de reconhecimento.
 - b.4. *Sinais de Interrupção:* sempre que ocorre o reconhecimento das informações recebidas pelo DCAN nesta fase, é gerado um sinal de interrupção para o processador local, so

licitando sua intervenção para:

b.4.1. o recebimento dos dados dos registradores e manutenção das filas de atendimento, se for o caso da implementação adotada;

b.4.2. ou para o atendimento imediato dos comandos que requerem um tratamento diferenciado (maior prioridade, por exemplo).

c. *Durante a 3a. fase (ponteiro e endereço fonte)*

c.1. *Monitoração e Reconhecimento*: após o reconhecimento das informações recebidas pelo DCAN na 2a. fase e a recepção do código do comando enviado em D1 na 3a. fase, o processador inibe o sistema de reconhecimento, pois a conexão entre os módulos fonte e destino já está estabelecida. Isto permite que se efetue um auto-endereçamento, o que pode ser uma ferramenta bastante útil em testes do sistema. Tem-se também a ativação de um sistema de controle de tempo de espera, que monitora o tempo gasto entre o envio de um sinal de reconhecimento das informações da 2a. fase e o recebimento das informações restantes, na 3a. fase. Se um valor limite for ultrapassado, o módulo destino rejeita as informações restantes, ignorando o comando recebido e reativando o sistema de monitoração e reconhecimento.

O sinal de reconhecimento é gerado quando os dados são recebidos com sucesso.

c.2. *Recepção de Dados*: após o recebimento do novo sinal de validação das informações do DCAN, há o registro dos dados disponíveis em D1 e D2 para posterior utilização no módulo. As observações de b.2, são válidas também aqui.

c.3. *Verificação de Paridade*: no caso de ocorrência de erro de paridade nesta fase, a interface gera um sinal para o processador local, para que sejam adotadas as providências pré-definidas para a manutenção da consistência dos dados no módulo (observar que somente parte das informações foram recebidas corretamente).

Se algum esquema de repetição da informação no duto, por parte do módulo fonte, for adotado, então o módulo de

destino deverá repetir o recebimento um certo número de vezes, sempre com monitoração da paridade.

- c.4. *Sinais de Interrupção*: basicamente, nesta fase não é necessário novos sinais de interrupção, pois o processador deverá estar atendendo à interface, para a recepção dos dados (ativado pela interrupção na fase anterior).

3.5 - *Observações Sobre os Sinais de Paridade e Falhas*

Uso de Paridade

Nos esquemas apresentados tem-se um circuito gerador de paridade e um circuito de verificação de paridade. Podem ser efetuadas algumas alterações no esquema, sem prejuízo no funcionamento do duto. Pode-se ter:

- a. em versões simplificadas do SDF, pode-se omitir o circuito gerador de paridade, porém para manter a possibilidade de futura implementação, deve-se prever a linha correspondente no DCAN.
- b. o circuito de verificação de paridade, pode ser implementado de modo centralizado, por exemplo no módulo de monitoração do DCAN ou em uma placa dedicada a essa função, evitando a implementação de diversos circuitos para essa finalidade. Deve-se observar, entretanto, que no caso de cada módulo possuir um circuito desse tipo em sua interface, pode-se detectar possíveis problemas ocasionados na conexão entre o módulo e o duto;
- c. o circuito de verificação de paridade, pode ser omitido no módulo e para manter a possibilidade de futura implementação, deve-se colocar em SPV (figura IV-3.3) um nível adequado ao funcionamento do sistema de monitoração e reconhecimento.

Indicação de Falhas

No DCAN existe a linha de falha LF, destinada a sinalizar a ocorrência de falhas graves no sistema. Essa linha não foi considerada em nenhum dos esquemas propostos aqui. Deve-se utilizá-la em cada módulo para a geração de um sinal de interrup-

ção, de alta prioridade, para que as ações pré-estabelecidas para os casos de falha, sejam ativadas.

O acionamento de LF é executado pelo módulo de monitoração do DCAN.

4 - CONSIDERAÇÕES SOBRE OS COMANDOS DE ALTO NÍVEL DO DCAN

4.1 - As Solicitações no SDF e os Comandos do DCAN

Os comandos de alto nível do SDF são utilizados, por um módulo, para gerar solicitações de ativação de tarefas em outros módulos do sistema. A via de comunicação utilizada é o DCAN e o formato para um comando de alto nível é o descrito em IV-2.2.

No SDF o usuário pode se comunicar com o sistema através da linguagem de comando do sistema operacional do módulo que estiver utilizando.

Uma solicitação de usuário pode ocasionar:

- a. a ativação de procedimentos locais, para o atendimento da solicitação dentro do próprio módulo, sem a geração de comandos para o DCAN;
- b. a ativação de procedimentos locais, para o atendimento da solicitação parcialmente no próprio módulo, com a geração de comandos para o DCAN, durante a execução solicitada;
- c. a ativação de procedimentos locais para atendimento da solicitação que geram diretamente comandos para o DCAN e ativam os procedimentos necessários para a comunicação de alto nível.

Observa-se, portanto, que o atendimento de uma solicitação do usuário se resume à ativação de procedimentos localizados no próprio módulo; o que pode variar é a necessidade ou não de conexão entre módulos.

O diagrama da figura IV-4.1 apresenta as possibilidades para o processamento de uma solicitação de usuário.

Para a geração dos comandos de alto nível, cada módulo deverá possuir uma tabela de informações com a relação das tarefas (fixas e temporárias) localizadas no próprio módulo e as localizadas em outros módulos, que possam ser solicitadas com comandos do DCAN.

As tarefas fixas são aquelas pré-estabelecidas nos módulos e que não podem ser desativadas por procedimento simples. Como exemplo tem-se os casos de programas gravados em memórias de apenas leitura.

Figura IV-4.1 - Atendimento de Uma Solicitação de Usuário

As solicitações recebidas em um dado módulo, através de comandos do DCAN, seguem um esquema de atendimento análogo ao das solicitações de usuários, porém com duas possibilidades apenas, conforme o diagrama da figura IV-4.2.

Figura IV-4.2 - Atendimento de Um Comando do DCAN

Também nesse caso pode-se observar que uma solicitação recebida pelo DCAN requer a ativação de procedimentos dentro do módulo destino, podendo ou não haver necessidade de novas conexões entre os módulos do SDF, pelo DCAN.

4.2 - Organização e Reconhecimento dos Comandos

Um comando do DCAN tem a estrutura da figura IV-2.1.

O código dos comandos tem uma organização própria, onde se define como um particular comando é identificado. Na figura IV-4.3 tem-se a estrutura básica do código de um comando do DCAN.

a. *classe do comando*: identifica um grupo de comandos.

a.1. com características semelhantes (mesmo tipo de atendimento - imediato, por fila ... -, igual prioridade de atendimento no sistema, comunicação de atendimento etc.);

a.2. ou que se refiram a um mesmo módulo do SDF.

A definição das classes é estabelecida na implementação do sistema e deve prever a possibilidade de alterações, visando expansão ou redução do número de classes definidas.

A escolha apropriada das classes tem reflexos diretos no desempenho do SDF pois, na arbitragem do DCAN, o código da classe do comando é utilizado como a parte de mais alta ordem na

formação do código de prioridade.

b. *Identificação do comando*: fornece a identificação de um particular comando, pertencente a uma dada classe.

A definição desse campo deve prever igualmente, possibilidades de fácil expansão ou redução do número de comandos, favorecendo as alterações necessárias para o reconhecimento dos comandos na interface do módulo.

A organização apresentada permite a geração dinâmica do código de prioridade para a arbitragem e facilita a implementação do circuito de reconhecimento de comandos da interface do DCAN, conforme esquema da figura IV-4.4 .

Nesse esquema tem-se:

- a. *Circuito de Identificação da Classe*: verifica se a classe do comando solicitado, pode ser atendida no módulo a que pertence. Esse circuito pode ser implementado por um sistema combinacional, conforme figura IV-4.5.

- b. *Circuito de Identificação do Comando*: determina qual comando, dentro de uma dada classe, está sendo solicitado. Basicamente é um circuito demultiplex, com as linhas de controle recebendo o código de identificação do comando. Se o módulo estiver apto a receber o comando solicitado, a linha de saída do demultiplex, correspondente ao código de identificação do comando, será acionada, indicando o reconhecimento; caso contrário o acionamento da linha correspondente indicará que o comando não pode ser recebido (ocorre então o bloqueio do reconhecimento).
- c. *Circuito de Ativação de Reconhecimento e Interrupções*: as linhas de saída do demultiplex que correspondem aos comandos válidos no módulo, acionam um circuito gerador de sinais de interrupção para o processador, que constituem dois grupos:
- c.1. Interrupção para recebimento de comandos que podem aguardar o atendimento em uma fila;
 - c.2. Interrupções especiais (uma linha para cada comando, por

exemplo), que indicam o recebimento de um comando que exige atendimento imediato, definido pela correspondente rotina de atendimento da interrupção.

Alguns dos códigos de indentificação dos comandos de uma dada classe podem não ter sido utilizados (deixados para expansão, por exemplo), ou embora correspondam a um comando válido no SDF não possam ser recebidos por um particular módulo do sistema (a tarefa ou função correspondente ao comando não está disponível no módulo). Nesse caso a linha de saída, correspondente ao comando, quando ativada indica que o reconhecimento do comando deve ser impedido.

O esquema da figura IV-4.6 mostra uma opção para esse circuito.

Para o reconhecimento de comandos de diversas classes por um mesmo módulo, pode-se empregar o esquema da figura IV-4.7 .

Figura IV-4.7 - Reconhecimento de Várias Classes de Comandos

4.3. Proposta de Comandos Para o DCAN

O objetivo deste ítem é mostrar alguns possíveis comandos de alto nível para o DCAN, que podem ou não ser implementados no sistema. A implementação estará sujeita à existência dos recursos necessários, na configuração do SDF que seja adotada.

Não é pretensão deste trabalho, propor um conjunto fixo de comandos de alto nível para o DCAN; o que se objetiva é apresentar as possibilidades e traçar diretrizes gerais para as futuras implementações.

A maior parte dos comandos pode ser implementada em configurações básicas do SDF, alguns comandos, porém, podem necessitar da existência de módulos mais complexos no sistema.

Para se referenciar a um comando de alto nível do DCAN, define-se a seguinte estrutura:

D\$xx...x <endereços, chaves, opções...>

onde

- D\$ indentifica um comando do DCAN e o diferencia de alguns comandos, com representação semelhante, da linguagem de comando do Sistema Operacional;
- xx...x é o mnemônico do comando correspondente;
- <....> correspondem aos endereçamentos, chaves especiais, opções de processamento e demais dados necessários para o envio e o atendimento do comando.

Na tabela IV-1 são apresentados alguns possíveis comandos para o DCAN, com o código sugerido e uma breve descrição de seu objetivo.

COMANDO	OBJETIVO
D\$USENT	- entrada de usuário no sistema
D\$USSAI	- saída de usuário do sistema
D\$SENHA	- alteração de senha de usuário
D\$DATHR	- fornecimento de data e hora
D\$ALOCA	- alocação de um módulo por outro
D\$LIBERA	- liberação de um módulo alocado
D\$ATIVA	- ativação remota de tarefa
D\$PARAT	- parada temporária de tarefa
D\$REATIVA	- reativação de tarefa paralizada
D\$CANCELA	- cancelamento de tarefa solicitada
D\$ATIVPP	- ativação remota de tarefa, passo a passo
D\$ESTSIS	- solicitação do estado do sistema
D\$CRIARQ	- criação de arquivo no módulo de disco (MD)
D\$ABREARQ	- abertura de um arquivo no MD
D\$FECHARQ	- fechamento de um arquivo no MD
D\$COPIARQ	- cópia de um arquivo
D\$APAGARQ	- desativação de um arquivo no MD
D\$ALTNOME	- alteração de nome de um arquivo no MD
D\$IMPARQ	- impressão de um arquivo do MD
D\$ESTARQ	- solicitação do estado de um arquivo do MD
D\$TRANAD	- transferência de arquivo do MD
D\$CATAREA	- solicitação de catálogo da area no MD
D\$TRANMM	- transferência do conteúdo da Memória de Contro le do Módulo de Monitoração
D\$FALHA	- aviso de falha no módulo destino
D\$REATMOD	- reativação de um módulo paralizado
D\$CRIAEM	- criação de um escaninho de mensagens (EM)
D\$ACRIAEM	- aviso de criação de um EM
D\$CANCEM	- cancelamento de um EM
D\$ENVIAM	- envio de mensagens (fonte para destino)
D\$ESPERAM	- solicitação de envio de mensagem (destino para fonte)

Tabela IV-1 - Comandos Para o DCAN

4.4 - Descrição dos Comandos

A definição dos comandos do DCAN pode, basicamente, seguir um dos seguintes caminhos:

- a. possibilitando a ocorrência de encadeamentos, entre os comandos, com um comando solicitando outros, diretamente;
- b. não permitindo esse encadeamento e, portanto, mantendo os comandos com uma caracterização de autonomia. Quando uma nova solicitação é necessária, deve haver uma sinalização para que o sistema operacional do módulo, após o término do comando, analise e efetue a solicitação normalmente. Esse tipo de procedimento diminui o risco de travamento do sistema.

A descrição dos comandos sugeridos neste trabalho (tabela IV-1), satisfaz ao exposto em b. e considera a organização dos módulos do SDF seguindo as sugestões do item III-1, com o emprego dos escaninhos de mensagens (EM).

DESCRIÇÃO RESUMIDA DOS COMANDOS DO DCAN

Comando	Informações Adicionais	Módulo Fonte	Módulo Destino	Observações
D\$USENT	- identificação do usuário - senha	Módulo de usuário	Módulo de disco	No módulo de disco ficam as estruturas de controle sobre usuários
D\$USSAI	- identificação do usuário	Módulo de usuário	Módulo de disco	Altera estado do sistema indicando a liberação do Módulo fonte
D\$SENHA	- identificação do usuário - senha antiga - senha atual	Módulo de usuário	Módulo de disco	No caso de criação de senha, o código antigo pode ser um padrão
D\$DATHR	- chave (solicitação ou alteração) - data - hora	Módulo de usuário	Módulo de monitoração	Pode-se implementar o controle de tempo em outro módulo do SDF
D\$ALOCA	- Módulo a ser alocado	Módulo de usuário	Módulo de Disco	O controle do estado do sistema pode ser efetuado no módulo de disco

<i>Comando</i>	<i>Informações Adicionais</i>	<i>Módulo Fonte</i>	<i>Módulo Destino</i>	<i>Observações</i>
D\$LIBERA	- Módulo a ser liberado	Módulo de usuário	Módulo de disco	Devem existir proteções contra solicitação de liberação indevida
D\$ATIVA	- identificação da tarefa	qualquer módulo do SDF	Módulo onde reside a tarefa	Utilizado somente para ativação de tarefas, remotamente
D\$PARAT (*)	- identificação da tarefa	qualquer Módulo do SDF	Módulo onde foi solicitada a tarefa	Pode se prever a implementação sem o envio da identificação da tarefa
D\$REATIVA (*)	- identificação da tarefa	qualquer Módulo do SDF	Módulo onde foi realizada uma tarefa	igualmente pode ser implementado sem a identificação da tarefa
D\$CANCELA (*)	- identificação da tarefa	Módulo de usuário	Módulo onde se solicitou a tarefa	Pode se ter implementação sem a identificação da tarefa
D\$ATIVPP	- identificação da tarefa - chaves especiais	Módulo de usuário	Módulo onde reside a tarefa	Podem ser implementadas as chaves para indicar o número de instruções de cada passo
D\$ESTSIS	- EM considerado - chaves especiais	Módulo de usuário	Módulo de disco	As chaves podem indicar o tipo de informação desejada
D\$CRIARQ	- identificação do arquivo - chaves especiais	qualquer Módulo do SDF	Módulo de disco	As chaves podem indicar características do arquivo
D\$ABREARQ	- identificação do arquivo	qualquer Módulo do SDF	Módulo de disco	—
D\$FECHARQ	- identificação do arquivo	qualquer Módulo do SDF	Módulo de disco	—
D\$COPIARQ (#)	- identificação do arquivo - fonte - destino - chaves especiais	qualquer Módulo do SDF	Módulo de disco	Transferência de arquivo do Módulo A para B, com: a. { A:M.fonte B:M.disco b. { A:M.disco B:M.fonte c. { A:M.disco B:M.disco

<i>Comando</i>	<i>Informações Adicionais</i>	<i>Módulo Fonte</i>	<i>Módulo Destino</i>	<i>Observações</i>
D\$APAGARQ	- identificação do arquivo - chaves especiais	Módulo de usuário	Módulo de disco	—
D\$ALTNAME	- identificação antiga - identificação nova	Módulo de usuário	Módulo de disco	—
D\$IMPARQ	- identificação do arquivo - chaves especiais	qualquer Módulo do SDF	Módulo de disco	Impressão na impressora associada ao módulo de disco
D\$ESTARQ	- identificação do arquivo - EM considerado - chaves especiais	Módulo de usuário	Módulo de disco	As chaves podem indicar o tipo de informação desejada
D\$TRANAD	- identificação - formato - dispositivo saída	Módulo de usuário	Módulo de disco	Transferência em ASCII, HEXADECIMAL ou BINÁRIO, conforme formato
D\$CATAREA	- identificação do usuário - EM considerado - chaves	Módulo de usuário	Módulo de disco	Catálogo referente a área do usuário; as chaves selecionam o tipo de informação desejada
D\$TRANMM	- dispositivo de saída - formato	Módulo de usuário	Módulo de Monitoração	Semelhante ao comando D\$TRANAD
D\$FALHA (*)	—	Módulo de Monitoração	qualquer Módulo do SDF	Pode ser enviado a todos os módulos ao mesmo tempo
D\$REATMOD (*)	—	qualquer Módulo do SDF	Módulo de Monitoração	Enviado para um módulo paralizado por D\$FALHA
D\$CRIAEM	- características - identificação de referência	qualquer Módulo do SDF	qualquer Módulo do SDF	Características: tipo e tamanho; o EM criado é temporário
D\$ACRIAEM	- identificação de referência - características - endereço base	qualquer Módulo do SDF	Módulo que enviou o D\$CRIAEM	—
D\$CANCEM	- endereço base do EM - características	qualquer Módulo do SDF	Módulo onde o EM foi criado	Cancelamento de um EM criado por solicitação do Módulo Fonte

<i>Comando</i>	<i>Informações Adicionais</i>	<i>Módulo Fonte</i>	<i>Módulo Destino</i>	<i>Observações</i>
D\$ENVIAM	- EM considerado - ponteiro da mensagem - tamanho	qualquer Módulo do SDF	qualquer Módulo do SDF	Módulo Fonte indica disponibilidade de mensagem para o Módulo de destino
D\$ESPERAM	- EM considerado - chaves especiais	qualquer Módulo do SDF	qualquer Módulo do SDF	Módulo Fonte sinaliza que está aguardando mensagem

(*) os comandos assinalados com asterisco, requerem atendimento imediato através da geração de interrupções especiais na interface;

(#) podem ser implementados comandos especiais para cada um dos casos a, b ou c.

Como observação geral sobre os comandos tem-se que, ao final de cada um são geradas mensagens adequadas, indicando o sucesso ou não da operação, ou outro fato relevante. Essas mensagens são passadas às rotinas do sistema operacional local, que, nos casos pertinentes, se responsabilizam pela emissão do comando de envio de mensagens aos módulos que solicitaram os comandos.

Para uma opção mais rápida, nos casos em que não são necessárias transferências de mensagens para outros módulos, mas sim apenas uma breve comunicação de sincronismo (por exemplo), podem-se adotar comandos no DCAN que tornem o mecanismo mais eficiente.

No caso do término de uma tarefa síncrona (por exemplo) acionada em um módulo Y por solicitação do módulo X, pode-se emitir um comando no DCAN que informe o sucesso (ou não) do atendimento, diretamente.

Isto pode ser implementado com a organização proposta para os comandos do DCAN, de modo bastante simples. Uma sugestão é a previsão de um ou mais bits no código da classe de comandos, por exemplo, e para o comando de solicitação mantém-se esse bit (ou esses bits) em zero; para o correspondente comando de resposta pelo DCAN (sincronismo) mantém-se esse bit em um (ou no caso de vários bits, no valor adequado à implementação).

Se alguma informação adicional, de referência, for necessária, pode-se utilizar o campo reservado ao envio do ponteiro do EM (não utilizado aqui).

Como exemplo, para o caso de um comando de ativação de tarefa (D\$ATIVA) pode-se ter:

	Comando	Código
solicitação:	D\$ATIVA	0xx...x
resposta:	D\$ATIVA*	1xx...x

Nesse caso o asterisco foi utilizado para diferenciar o comando na solicitação e na resposta.

Na figura IV-4.8 é apresentado o esquema de utilização do DCAN para o exemplo.

solicitação

⟨D\$ATIVA⟩ ⟨end.destino⟩ ⟨ponteiro inf.adic.⟩ ⟨mōd. fonte⟩

resposta

⟨D\$ATIVA*⟩ ⟨end.fonte⟩ ⟨referência sincronismo⟩ ⟨mōd.destino⟩

Figura IV-4.8 - Esquema Para Sincronismo Através do DCAN

5 - ASPECTOS DA ESTRUTURAÇÃO LÓGICA DO SDF

5.1 - Estrutura Lógica para Envio e Atendimento de Comandos de Alto Nível

Para que uma solicitação seja efetuada pelo DCAN, através de algum comando de alto nível, deve-se ter:

- a. Ativação da Estrutura Lógica de Envio de Comandos, do módulo fonte.

Nesse caso, tem-se um conjunto básico de ações para todos os comandos, que está sintetizado no diagrama de blocos da figura IV-5.1 ;

- b. Ativação do mecanismo de transmissão e recepção dos comandos pelo DCAN (descrito em IV-3);

- c. Ativação da Estrutura Lógica de Atendimento dos Comandos, do módulo destino.

No caso do atendimento dos comandos distinguem-se dois esquemas básicos:

- c.1. esquema para os comandos que podem aguardar em filas de atendimento;

- c.2. esquema para os comandos que requerem atendimento imediato.

Para os comandos que podem ser atendidos segundo c.1, distinguem-se três fases no atendimento:

1a. Fase - Busca:

- a. de informações na fila de atendimento (código do comando, módulo fonte e ponteiro para obtenção de informações adicionais);
- b. de informações no EM do módulo fonte (as informações adicionais);

2a. Fase - Preparação:

- a. decisão sobre a possibilidade de atendimento do comando;
- b. decisão sobre as opções consideradas, de acordo com as informações recebidas;

Figura IV-5.1 - Procedimentos Para o Envio de Comandos do DCAN

3a. Fase - Execução

Atendimento da solicitação, propriamente dita, com o acionamento dos procedimentos que compõem a tarefa solicitada.

No caso de comandos que seguem o esquema c.2, ocorre o acionamento de interrupções, no reconhecimento do comando. Esse atendimento pode:

- a. requerer a busca de informações adicionais no EM.
Nesse caso tem-se as três fases descritas, porém a fase de busca composta apenas do item b.
- b. não requerer a busca de informações adicionais no EM.
Nesse caso tem-se apenas duas fases (2a. e 3a.) sendo que a fase de preparação fica reduzida ao item a.

A figura IV-5.2 apresenta um diagrama de blocos dos procedimentos para o atendimento de comandos do DCAN, segundo as considerações apresentadas aqui.

Figura IV-5.2 - Procedimento para o Atendimento de Comandos

5.2 - Esquema Lógico Para os Módulos do SDF

Na figura IV-5.3 tem-se um esquema básico para a organização lógica de um módulo do SDF, mostrando a interação entre usuário, software e interligações físicas do sistema.

Tradicionalmente, nos sistemas distribuídos com interligação serial, utilizam-se a estrutura de níveis funcionais apresentada em II-1.7 .

Nesse caso, a organização lógica obriga a definição de protocolos de comunicação em vários níveis onde, a cada nível, novas informações de controle são agregadas à informação original (que pode ser uma mensagem, um comando, um dado, etc..). Na figura IV-5.4 é apresentada uma representação desse esquema {Di82,Ry81}.

No SDF, uma vez definida a necessidade de uma solicitação de alto nível, define-se o comando adequado através de rotinas do sistema operacional local e, empregando-se a lógica de envio de comando do DCAN, faz-se a transmissão da solicitação sem a concatenação de outras informações, a não ser endereçamentos e

dados necessários ao atendimento (IV-2 e 3).

O protocolo de comunicação é implementado pelas linhas de controle do DCAN e pelos mecanismos de controle da interface.

O envio de mensagens longas, que não podem ser sintetizadas em um único comando, é efetuado através do DDE. Nesse caso utilizam-se rotinas do sistema operacional local e a lógica de envio e recebimento de informações de baixo nível, para a conexão com o módulo adequado. A transferência segue o esquema proposto em IV-1.

No recebimento e atendimento das solicitações de alto nível, utiliza-se a estrutura de atendimento de comandos e o mecanismo de recepção de comandos. A decodificação do comando pode ser feita diretamente em hardware para certos casos e por software em outros, conforme mostra a figura IV-5.5.

No caso dos protocolos em vários níveis, a decodificação é feita nível por nível, num caminho inverso ao da composi

ção da mensagem final, conforme mostrado na figura IV-5.6

6 - O DUTO DE DADOS E ENDEREÇOS (DDE)

6.1 - Considerações Gerais

O DDE é a via de comunicação de baixo nível do SDF, destinando-se à interligação dos módulos do sistema para as transferências de dados, endereços e sinais de controle.

As características gerais do SDF (descritas em III-1) direcionam a definição (ou escolha) do DDE. Assim, o DDE deve apresentar as seguintes características:

- a. facilidades para a implementação de arquiteturas distribuídas;
- b. independência do processador, possibilitando o emprego de processadores de 8, 16 ou 32 bits, conforme as aplicações de cada módulo do sistema;
- c. facilidade de adaptação a novas tecnologias, evitando-se as soluções particularizadas que dependam diretamente da tecnologia disponível;
- d. possibilidade de endereçamento com grande número de bits, facilitando a implementação do módulo de memória virtual do SDF (tipicamente espera-se ter até 32 bits para endereçamento);
- e. facilidades para implementação de mecanismos de transferência de blocos de dados, concorrendo para um melhor desempenho do SDF;
- f. possibilidade de implementação de esquemas de arbitragem adequados a sistemas distribuídos, evitando-se soluções que empreguem um número elevado de linhas no duto (razões físicas e econômicas) ou que tornem o sistema mais vulnerável a falhas;
- g. facilidade de expansão do sistema, pois o SDF é um sistema em desenvolvimento estando sujeito a alterações e expansões. Além disso uma característica forte do SDF é a modularidade, com a conseqüente possibilidade de alterações na composição do sistema. O DDE deve ser flexível de modo a acompanhar e facilitar essas alterações.

O desempenho global do SDF é fortemente influenciado pelos aspectos abordados, pois as características do DDE ficam definidas a partir deles.

Neste trabalho não se pretende, porém, apresentar uma proposta única para a definição do DDE. O objetivo com relação ao DDE, é o de ressaltar suas características básicas e apontar, dentre os dutos disponíveis e considerados em II-2, aqueles que podem ser utilizados no SDF, com determinadas adaptações.

6.2 - As opções disponíveis

Dos dutos considerados em II-2, observa-se que com algumas adaptações podem ser adotadas as seguintes opções:

- a. *S-100* {Ga82,St79,Zu81}: com adaptações para expansão do endereçamento, se necessário (dependendo da configuração do SDF considerada);
- b. *ZBI* {Be80}: com implementação de arbitragem distribuída (opção existente na definição desse duto);
- c. *VERSABUS/VMEBUS* {De82,Ka81}: sendo dispensada a utilização do duto de inter-inteligência (II-2), pois a comunicação de alto nível entre os módulos do SDF, é feita pelo DCAN;
- d. *FUTUREBUS* {Al81,Bo84}: sendo dispensada a utilização do S-bus, devido a existência do DCAN.

Os demais dutos apresentados em II-2 não estão considerados aqui, por apresentarem características não recomendadas ao SDF, como descrito a seguir:

- a. *MULTIBUS* {Bo80,Bo82,Zu81}: arbitragem por encadeamento em cascata ("daisy-chain") ou por esquema centralizado;
- b. *STD* {E182} : destinado a sistemas computacionais com processadores de 8 bits, síncrono, baixa capacidade de endereçamento e arbitragem por encadeamento em cascata;
- c. *EUROBUS* {Hi82} : esquema de arbitragem centralizado, exigindo linhas de requisição ligadas em estrela, através do duto.

6.3 - Análise das Opções Apresentadas

Das opções apresentadas para o DDE, acredita-se que o Versabus (ou o VMEbus) e o Futurebus sejam os dutos que melhor

se adaptam ao SDF, devido as suas características modernas.

Em particular o Futurebus, definido segundo diretrizes modernas de independência quanto ao processador e quanto à tecnologia, tecnicamente é a melhor opção, porém a sua implementação por ser de custo elevado e por apresentar certa complexidade no interfaceamento, pode não ser a mais adequada.

Tanto no caso do VMEbus como no do Futurebus, o subduto serial de comunicação entre os módulos do sistema é, em princípio não utilizado, devido à existência do DCAN.

As outras opções apresentadas (S-100 e ZBI) podem, sem dúvida, serem adequadas à implementação do SDF, principalmente para configurações básicas, onde o número de módulos e a utilização do sistema favoreçam o emprego desses dutos.

A escolha entre um desses dutos (ou especificação de um novo duto) para o SDF deve ser feita levando-se em consideração a configuração a ser implementada e sua possível expansão, compatibilizando as características do SDF com o duto adotado.

CAPÍTULO V

SOBRE ALGUMAS CONSIDERAÇÕES PARA IMPLEMENTAÇÃO DO SDF

1 - UMA CONFIGURAÇÃO BÁSICA PARA O SDF

1.1 - Os Módulos da Configuração Básica

A configuração básica proposta para o SDF é a apresentada na figura V-1.1 e possui

- Módulos de Usuários (2)
- Módulo de Memória Virtual (1)
- Módulo de Disco (1)
- Módulo de Monitoração do Sistema (1)

Figura V-1.1 - Arquitetura da Configuração Considerada do SDF

Essa configuração permite a implementação de uma grande variedade de comandos de alto nível pelo DCAN (item V-1.2) e satisfaz os requisitos básicos estipulados para o SDF.

Módulos de Usuários

Os módulos de usuários (descritos em III-3) permitem a implementação de comandos envolvendo:

- a. solicitações de usuários de um modo geral
- b. manipulação de arquivos no módulo de disco
- c. troca de mensagens entre módulos
- d. alocação de módulos
- e. uso de memória virtual
- etc..

A existência de módulos de usuários nessa configuração do SDF, permite a sua utilização para o desenvolvimento de software, que constitui uma parcela considerável do trabalho que o sistema deverá suportar.

Módulo de Memória Virtual

A presença do módulo de memória virtual é importante, pela expansão da capacidade individual de armazenamento de cada módulo e pelo aspecto funcional, pois todo o gerenciamento e implementação da memória virtual é efetuado localmente, de modo a centralizar os circuitos e as rotinas necessárias.

Em versões iniciais do SDF pode-se dispensar a implementação desse módulo, porém com isto cada módulo do sistema terá apenas a sua memória local. Pode-se também implementar um módulo de memória sem processador, desprovido de capacidade de gerenciamento e implementação de memória virtual (nesse caso o módulo não pode ser interligado ao DCAN, pois os comandos de alto nível requerem um processador para o seu atendimento).

Módulo de Disco

O módulo de disco é parte fundamental do sistema, pois é nele que estará implementado o gerenciamento de arquivos e do sistema (controle do estado do sistema, catálogos de áreas de usuários, cadastros de usuários, estatísticas de uso, etc..).

Como pode ser observado em IV-4, a solicitação desse módulo é bastante intensa e, portanto, sua implementação deve apresentar certas características que favoreçam a sua utilização, evitando que se torne um ponto crítico no SDF.

Módulo de Monitoração do Sistema

Como visto em III-3, o módulo de monitoração constitui um módulo de usuário com características adicionais que permitem a monitoração do SDF, em particular do fluxo de informações que circulam pelo DCAN.

A presença desse módulo na configuração básica do SDF é indispensável, pois permitirá uma melhor depuração do sistema. A tarefa de depuração é, por si, bastante árdua; no caso de sistemas em desenvolvimento isto se torna mais crítico e a existência de um módulo dedicado, que sirva de apoio, é altamente desejável.

1.2 - Comandos de Alto Nível Para a Configuração Proposta

A configuração apresentada pode suportar todos os comandos de alto nível considerados em IV-4. Nas versões iniciais pode-se escolher um sub-conjunto básico de comandos e prever, na implementação, a inserção de novos comandos posteriormente.

No caso considerado tem-se, por exemplo:

a. Classes de comandos

Uma análise dos comandos sugeridos em IV-4 mostra que 6 classes são suficientes para sua organização na configuração básica, tendo-se:

<i>classe 0</i> <i>(menor prioridade)</i>	<i>classe 1</i>	<i>classe 2</i>
D\$USENT	D\$ATIVA	D\$CRIARQ
D\$USSAI	D\$PARAT	D\$ABREARQ
D\$SENHA	D\$REATIVA	D\$FECHARQ
D\$DATHR	D\$CANCELA	D\$COPIARQ
D\$ALOCA	D\$ATIVPP	D\$APAGARQ
D\$LIBERA	D\$ESTSIS	D\$ALT NOME
D\$CATAREA		
<i>classe 3</i>	<i>classe 4</i>	<i>classe 5</i> <i>(maior prioridade)</i>
D\$IMPARQ	D\$CRIAEM	D\$TRAMM
D\$ESTARQ	D\$ACRIAEM	D\$FALHA
D\$TRANAD	D\$CANCEM	D\$REAT
	D\$ENVIAM	
	D\$ESPERAM	

Nesse caso, o número máximo de comandos por classe é 7.

b. *Organização dos códigos dos comandos*

Para a proposta do item a deve-se ter:

para classes - 3 bits no mínimo;

para identificação do comando - 3 bits no mínimo.

Se as considerações sobre comandos para implementação de sincronismo pelo DCAN (IV-4.4) forem adotadas, devem-se prever mais bits no código do comando. Também pensando-se em expansão é aconselhável o aumento do número de bits.

Considerando-se que com 3 bits podem-se ter até 8 classes com no máximo 8 comandos em cada, sugere-se para essa configuração 2 bits extras que podem servir tanto para um aumento no número de classes implementadas (consequentemente aumento no número de comandos), como para implementação do esquema de sincronismo sugerido em IV-4.4.

Assim, tem-se 8 bits para um comando do DCAN. Na figura V-1.2 tem-se a organização sugerida.

1.3 - Configuração do DCAN

Para a organização do DCAN, consideram-se:

a. *Código do Comando e Ponteiro para Informações Adicionais*

Para o sub-duto D1 do DCAN necessitam-se de 8 linhas, de acordo com V-1.2 . Nesse caso, os ponteiros para informações adicionais, que são utilizadas no acesso ao EM que contém os dados adicionais necessários ao atendimento do comando, podem assumir valores desde 0 até 255. Assim é possível a definição de um EM com até 256 registros de informação.

b. *Endereçamento dos Módulos*

Na configuração da figura V-1.1 tem-se 5 módulos, o que exige um mínimo de 3 linhas para endereçamento no DCAN. Com isto podem-se ter até 8 módulos no sistema. Como sugestão, prevendo-se expansão, pode-se acrescentar 1 linha possibilitando até 16 módulos no sistema. Assim o sub-duto D2 do DCAN deverá ter 4 linhas.

Considerando-se a possibilidade de endereçamento conjunto para os módulos, com 4 bits o sistema admite até 15 módulos, ficando um endereço reservado ao endereçamento conjunto.

A sugestão para endereçamento é:

Módulo de Monitoração -	endereço 15 $(1111)_2$
Módulo de Disco -	endereço 14 $(1110)_2$
Módulo de Memória Virtual -	endereço 13 $(1101)_2$
Módulo de Usuário 1 -	endereço 1 $(0001)_2$
Módulo de Usuário 2 -	endereço 2 $(0010)_2$

O código reservado ao endereçamento conjunto é o endereço zero - $(0000)_2$.

O endereço do módulo é também usado na arbitragem do DCAN, como parte do código de prioridade (IV-2.4).

Algumas alterações nos endereços dos módulos podem ser necessárias, por exemplo no caso de introdução de novos módulos no sistema. Assim, a definição do endereço de um módulo deve ser efetuada por um esquema que permita alterações simples, como mostra o esquema da figura V-1.3 .

Figura V-1.3 - Definição do Endereço de Um Módulo

O módulo de monitoração, nesse exemplo tem o endereço mais alto com o objetivo de que, nos casos necessários, tenha maior prioridade no uso do DCAN.

Novos módulos inseridos no sistema devem ser endereçados de 3 a 12, a menos que se procedam alterações.

c. A Arbitragem

Para o sub-duto D3 do DCAN, o sub-duto de arbitragem, são necessárias 5 linhas para a classe do comando a ser utilizado e 4 linhas para o endereço do módulo (fixado na implementação). Essas linhas permitem a implementação do esquema de arbitragem, segundo a descrição dada em IV-2.4, com a definição do código de prioridade efetuada dinamicamente.

d. O sub-duto de controle

Para o sub-duto D4 do DCAN, onde estão as linhas de controle utilizadas no duto, tem-se 7 linhas conforme já descrito em 2.3 e 2.4 do capítulo IV.

e. As Linhas de Expansão

No caso do endereçamento dos módulos e no caso do código dos comandos, foram consideradas algumas linhas sobressalentes, in

corporadas aos sub-ductos D1 e D2. Com isto, a definição das linhas do sub-ducto D3 também sofreram um acréscimo natural, dada a maneira como foram consideradas (código de prioridade). Assim, para o sub-ducto D5 do DCAN são sugeridas apenas 2 linhas de expansão, para eventuais necessidades que não possam ser resolvidas com as linhas já reservadas.

Na figura V-1.4 apresenta-se o esquema geral do DCAN, segundo as propostas apresentadas.

1.4 - O Duto de Dados e Endereços

Como apresentado em IV-6, um dos dutos padronizados existentes pode ser adotado para o SDF. No caso da configuração apresentada, as duas opções apontadas como as melhores no capítulo anterior, podem ser adotadas. Se a implementação desses dutos (VMEbus ou Futurebus) for considerada não adequada nas versões iniciais do sistema, devido ao custo ou complexidade do sistema de interface, pode-se então optar pelo ZBI ou mesmo pelo S-100, ambos de complexidade menor.

No caso do S-100, a implementação já é bastante difundida, não oferecendo grandes dificuldades. Porém deve-se observar que o duto de endereços, nesse caso, tem 24 linhas e os processadores deverão ser de 8 ou 16 bits. As linhas de endereçamento poderão restringir a expansão do sistema com relação ao módulo de memória virtual e o tipo de processador pode quebrar a versatilidade esperada do SDF. Contudo, podem-se introduzir algumas alterações adequando a implementação aos objetivos do SDF, porém fugindo-se do padrão estabelecido para o duto.

1.5 - *O Interfaceamento dos Módulos com o DCAN*

O sistema de interface para a configuração básica apresentada deve seguir as diretrizes apresentadas em IV-3. No caso de versões iniciais, pode-se porém introduzir algumas simplificações que facilitem a implementação do sistema. Alguns sinais de controle sugeridos nos diagramas apresentados em 3.2, 3.3 e 3.4 do capítulo IV, para implementação em hardware, podem ser implementados por software; outros podem ser omitidos sem o prejuízo global do sistema (é o caso do sinal de paridade).

Na seção 2 deste capítulo são abordados alguns tópicos relacionados com as diretrizes básicas para implementação das interfaces do DCAN, apresentando uma versão simplificada.

2 - DIRETRIZES PARA IMPLEMENTAÇÃO DA INTERFACE DO DCAN

2.1 - Considerações Iniciais

Como mencionado anteriormente, existem diversas possibilidades para a implementação do sistema de interfaceamento entre um módulo do SDF e o DCAN. Os esquemas apresentados em IV-3 têm um caráter geral e, em princípio, prevêm uma implementação em hardware, ocupando o processador do módulo apenas no que for indispensável sua participação.

No caso de sistemas distribuídos com módulos funcionais, como o caso do SDF, o processador de cada módulo tem um conjunto de tarefas pré-definidas para atender, o que permite um balanceamento de modo a se poder, eventualmente, utilizá-lo para o processamento de algumas rotinas dedicadas ao sistema de interface do DCAN. É evidente que numa implementação por hardware os tempos envolvidos são minimizados, mas em certas circunstâncias o atraso introduzido pelo software não é prejudicial ao desempenho do sistema e pode simplificar sua implementação.

Alguns dos sinais de controle definidos para o DCAN podem ser omitidos em versões simplificadas do SDF. Por exemplo, como observado em IV-2 e IV-3, o sinal de paridade pode ser excluído; conseqüentemente o sinal de erro de paridade também. Nesse caso pode-se deixar prevista no duto a linha correspondente, para futura implementação.

A exclusão desse sinal não prejudica o desempenho do SDF, por se tratar de um sistema com interligações de pequena distância. Porém, a sua implementação aumenta a confiabilidade do sistema.

Uma versão simplificada para o sistema de interface do DCAN, onde alguns sinais de controle são gerados e monitorados por software, é apresentada a seguir.

2.2 - Exemplo de Implementação

Com a finalidade de verificar o pleno funcionamento do protocolo proposto para uma transação no DCAN, a possibilidade de uma implementação simples para o sistema de interface do DCAN e avaliar, preliminarmente, o desempenho do sistema com relação ao uso do DCAN, implementou-se uma versão experimental da in

terface, conectando-se dois módulos baseados no microprocessador 8085 (KMD-85). Foram desenvolvidas as rotinas necessárias para a geração e monitoração dos sinais de controle do DCAN e as rotinas para o envio e recepção dos comandos de alto nível.

Na figura V-2.4, no final desta seção, é apresentado o circuito implementado para a interface do DCAN.

Descrição do Esquema Implementado

Para o esquema implementado foram adotadas as seguintes linhas no DCAN:

- *sub-duto D1*: 3 linhas (1 linha para a classe do comando e 2 para sua identificação);
- *sub-duto D2*: 2 linhas para o endereçamento dos módulos;
- *sub-duto D3*: 2 linhas para a arbitragem (utilizando apenas o endereço do módulo como código de prioridade);
- *sub-duto D4*: 4 linhas (não foram implementadas as linhas de paridade - LP, de erro de paridade - LEP e de falha - LF).

Para os módulos conectados foram consideradas as seguintes definições:

Módulo 1

endereço $(01)_2$

classe de comandos válida $(0)_2$

comandos válidos $(11)_2$ e $(00)_2$

Assim, para uma conexão com este módulo deve-se ter a seguinte configuração nas linhas do DCAN:

$$\begin{array}{ccc}
 D_1 \left\{ \begin{array}{l} 0 \\ 0 \\ 0 \end{array} \right. & & D_1 \left\{ \begin{array}{l} 0 \\ 1 \\ 1 \end{array} \right. \\
 & \text{ou} & \\
 D_2 \left\{ \begin{array}{l} 1 \\ 0 \end{array} \right. & & D_2 \left\{ \begin{array}{l} 1 \\ 0 \end{array} \right.
 \end{array}$$

Módulo 2

endereço $(10)_2$

classe de comando válida $(1)_2$

comandos válidos $(11)_2$ e $(00)_2$

Nesse caso, para uma conexão com este módulo deve-se ter a seguinte configuração nas linhas do DCAN:

$$\begin{array}{ccc} D_1 & \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} & \text{ou} & D_1 & \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} \\ D_2 & \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} & & D_2 & \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \end{array}$$

Basicamente, no esquema implementado empregou-se uma interface paralela 8255, com suas portas programadas na seguinte configuração:

porta A - saída de dados

porta B - entrada de dados

porta C $\begin{cases} \text{- entrada de dados (bits 0,1,2 e 3)} \\ \text{- saída de dados (bits 4,5,6 e 7)} \end{cases}$

Para o acionamento da linha de arbitragem, utilizou-se um circuito bi-estável tipo D, controlado pelo processador.

O detalhamento do esquema considerado e o seu funcionamento é apresentado a seguir:

- a. *Sistema de Arbitragem*: o circuito adotado para a arbitragem é o sugerido por Taub em [Ta84a], com utilização das linhas de arbitragem (sub-duto D3 do DCAN) em coletor aberto (indicado por *), como mostrado na figura V-2.1

Funcionamento do Esquema

Para a ativação da arbitragem, quando um módulo precisa utilizar o DCAN, monitora-se a linha de arbitragem LA (do sub-duto de controle do DCAN) através de leitura do bit 3 da porta C

da interface paralela (8255), verificando se a arbitragem está autorizada (nível lógico 1 na linha). Quando a arbitragem estiver autorizada aciona-se a linha de requisição do duto (interna à interface), através do bit 6 da porta A e o módulo entra no processo de arbitragem.

Através do bit 7 da porta B o módulo monitora o sinal de saída do circuito de arbitragem, verificando se foi o vencedor. Se algum outro módulo venceu o processo de arbitragem, a linha LA estará em nível lógico 0 e o processo é encerrado; caso contrário quando o módulo observa que venceu o processo de arbitragem, aciona a linha LA mantendo-a em nível lógico 0. Um novo processo de arbitragem é autorizado quando se inicia o uso do DCAN, conforme esquema de superposição no tempo proposto em IV-2.

A rotina utilizada para o controle do processo de arbitragem, está apresentada no apêndice.

- b. *Sistema de Envio de Comandos*: após um módulo ter obtido sucesso na arbitragem, é acionada a rotina de coordenação do envio de comandos pelo DCAN. Nesse caso há uma monitoração, através do bit 2 da porta C, da linha de disponibilidade do duto (LD-em coletor aberto), até ocorrer a liberação do DCAN (LD em nível lógico 1). O módulo coloca essa linha em nível lógico 0, através do bit 7 da porta C, e pode então acessar o DCAN. O envio de comandos segue o protocolo estabelecido em IV-2 e IV-3 e a "escrita" no DCAN é feita através da porta A da 8255. O esquema básico utilizado é o da figura V-2.2 e deve ser observado que:
- não é empregado o sinal de paridade;
 - os sinais de controle são gerados e monitorados na programação;
 - o controle do tempo de conexão entre os módulos é implementado também na programação.

Funcionamento do Esquema

Após o módulo ter acessado o DCAN ($LD = \emptyset$), há a liberação do novo processo de arbitragem, colocando-se LA em nível lógico 1 através do bi-estável tipo D utilizado e inicia-se o envio do 1º ciclo de informações pelo DCAN. O processador escreve a informação adequada na porta A da 8255 e acionando o bit 4 da porta C, acopla os sinais nas linhas do DCAN. Em seguida é feita a validação da informação (bit 6 da porta C) e é iniciada a monitoração do sinal de reconhecimento (bit 0 da porta C) e a contagem de tempo de conexão. Se o sinal de reconhecimento é recebido com sucesso, inicia-se o 2º ciclo que, basicamente, é idêntico ao 1º; caso contrário o envio é cancelado e o processador sinaliza o erro.

No final de cada ciclo os sinais de controle acionados são retornados aos valores iniciais. No final da transação o duto é liberado.

As rotinas utilizadas para o controle do envio de comandos pelo DCAN, estão apresentadas no apêndice.

- c. *Sistema de Reconhecimento e Recepção de Comandos*: o reconhecimento do endereçamento do módulo e do comando recebido é efetuado pelo circuito combinacional mostrado na figura V-2.3 e segue basicamente o esquema proposto em IV-4.

Funcionamento do Esquema

Nesse caso tem-se um circuito que monitora as linhas do DCAN constantemente, desde que a linha de autorização do processador esteja acionada (bit 6 da porta A em nível lógico 1) e os sinais no DCAN sejam válidos (linha LV do DCAN em nível lógico 0).

Se o endereço é reconhecido e o comando solicitado é válido, há o acionamento da interrupção 6.5 do microprocessador 8085 (ativa em nível lógico 1, mas no KMD-85 deve-se ter um nível lógico 0 na linha correspondente, pois há um circuito inversor conectado internamente), que aciona a rotina de atendimento adequado. No caso desta implementação, tem-se apenas uma rotina de atendimento que simula as funções necessárias para a recepção e armazenamento dos dados recebidos, na memória do módulo.

A rotina de atendimento utilizada está apresentada no Apêndice.

Figura V-2.4 - Um Circuito de Interface para o DCAN

2.3 - *Considerações de Desempenho para a Implementação Efetuada*

Analisando-se o sistema de interface implementado e a execução de algumas transferências de informação pelo DCAN, observa-se que o protocolo sugerido é exequível e o tipo de conexão obtida eficiente.

Uma análise preliminar mostra que:

- a. A superposição do processo de arbitragem com o 1º ciclo de envio de informações pelo DCAN é totalmente viável e eficiente;
- b. O tempo para a conexão de um módulo com outro envolve:

- rotina de envio: ~ 150 μ s
- reconhecimento de endereço e código de comando:
atraso de 4 portas TTL (no caso desta implementação), totalizando de 30 a 40 ns. (podendo ser diminuído com o emprego de circuitos com coletor aberto).
- recepção das informações: ~ 140 μ s

Certamente, se a opção adotada fosse a de um esquema serial, o tempo de conexão seria maior pois envolveria:

- recepção dos bits do endereço (conversão serial/paralelo)
- decodificação do endereço para reconhecimento
- recepção dos bits do comando (conversão serial/paralelo)
- decodificação do comando para reconhecimento
- recepção bit a bit das demais informações do 2º ciclo de informações do DCAN

Como pode ser observado, o tempo de conversão serial/paralelo e a recepção bit a bit das informações introduz atrasos inevitáveis, nesse tipo de transação.

- c. O auto endereçamento na implementação efetuada é possível, porém não permite uma transação completa pelo DCAN, pois a rotina de atendimento da interrupção, e a de envio do comando são processadas no mesmo processador. Nesse caso pode-se introduzir alguns comandos especiais, voltados para esse tipo de endereçamento, com a finalidade de auto-teste

- d. Para o endereçamento conjunto, pode-se ter um esquema análogo ao empregado no reconhecimento do endereçamento do módulo, conforme figura V-2.5.

Figura V-2.5 - Esquema Para Reconhecimento de Endereçamento Conjunto

Nesse caso, a única alteração introduzida é um atraso de 1 porta TTL no tempo gasto para o reconhecimento do endereço e comando. Ainda assim, pode-se modificar o esquema de modo a eliminar a porta "ou" (por exemplo introduzindo o uso de circuitos com coletor aberto).

2.4 - Opções Para Futuras Implementações

Para futuras implementações do SDF, baseando-se na implementação efetuada, observa-se que algumas partes do sistema de interface podem ser implementadas por software sem comprometer o desempenho global do sistema.

Uma implementação por hardware da interface é possível e, certamente, dá ao sistema uma maior velocidade nas transações pelo DCAN.

Em qualquer dessas opções deve-se seguir o esquema a apresentado em IV-3.

Como observado anteriormente, a implementação de al-

gumas partes do sistema de interface por software representa uma carga de processamento adicional para o processador do módulo. Uma alternativa existente é a utilização de um processador dedicado à interface, facilitando a implementação e não sobrecarregando o processador do módulo. Isto representa uma opção para futuras implementações do SDF, porém cabe salientar que envolve a implementação de um esquema de exclusão mútua entre os processadores, no acesso à memória comum (por exemplo) onde for implementada a fila de atendimento.

CAPÍTULO VI

CONCLUSÕES

1 - CONCLUSÕES FINAIS

1.1 - Sobre o Estudo Bibliográfico

O estudo bibliográfico desenvolvido mostra que a conceituação dos sistemas de processamento de dados distribuídos é diversificada, com os sistemas computacionais sendo caracterizados como distribuídos ou não, segundo pontos de vista distintos. Uma conceituação generalizada {En78} apresenta critérios difíceis de serem atendidos e inadequados a muitos sistemas modernos.

Uma visão moderna da conceituação de sistemas distribuídos, apresentada em {Fa83}, considera os sistemas distribuídos baseados em microprocessadores como uma classe particular e os recomendam para aplicações onde são necessários módulos dedicados, existindo razoável independência entre as diversas tarefas desenvolvidas no sistema. O SDF, segundo as definições e características apresentadas neste trabalho, satisfaz essa conceituação.

Os trabalhos considerados no estudo sobre dutos de sistemas computacionais, abrangem a conceituação geral e a descrição de diversos dutos padronizados, disponíveis atualmente.

O quadro resumo apresentado na Tabela II-1 permite uma visão comparativa das características básicas dos diversos dutos considerados. Destaca-se dessa comparação que o VME e o Futurebus apresentam um sub-duto serial, destinado à implementação de comunicação entre os diversos módulos de um sistema com arquitetura distribuída. Também, com relação ao sincronismo dos dutos considerados, observa-se que a maioria é assíncrono, confirmando a preferência por esse esquema por permitir uma melhor acomodação de módulos diversificados em um mesmo sistema, favorecendo fortemente a independência quanto ao processador e a assimilação de avanços tecnológicos.

1.2 - Sobre a Proposta Apresentada para o SDF

A proposta apresentada neste trabalho para o Sistema Distribuído Funcional (SDF) satisfaz as características básicas estipuladas para o sistema (apresentadas em III-1) e fixa diretrizes que permitam futuras implementações do sistema, iniciando por versões simplificadas e podendo evoluir para versões mais complexas, agregando-se novos módulos funcionais.

A característica de distribuição funcional imposta ao sistema, evidencia que a distribuição desejada do processamento ocorre a nível de execução de tarefas (ou funções) completas e não a nível de instruções ou operações elementares, como é o caso de diversas outras arquiteturas distribuídas.

A comunicação entre módulos proposta para o SDF, utilizando dutos paralelos distintos para o fluxo de comandos de alto nível (que acionam o processamento distribuído funcional) e para o fluxo de dados e endereços (destinado à transferência de informações de baixo nível, entre módulos), dá ao sistema possibilidade de obtenção de alta velocidade na conexão entre módulos, tanto para solicitações de alto nível como na transferência de dados.

O DCAN, como proposto neste trabalho, constitui um duto de comunicação rápido, de implementação simples e que pode suportar processadores de diversos tipos. O sistema de interface necessário é simples e pode ser implementado segundo filosofias distintas. A versão experimental implementada mostra a possibilidade de se iniciar a implementação do SDF por uma versão simplificada, com determinadas partes da interface implementadas e controladas por hardware e outras por software.

Uma comparação entre os esquemas de comunicação serial, empregados em diversos sistemas distribuídos (como no caso das redes de computadores), e o esquema proposto para o SDF mostra que:

- a. a existência de dois dutos, um para os comandos de alto nível e outro para a transferência de dados e endereços, favorece diretamente o desempenho do SDF, pois ocorre uma

desvinculação entre os tipos de comunicação, permitindo uma superposição no tempo. É sempre possível a ocorrência de uma conexão entre dois módulos pelo DCAN, simultaneamente com uma transferência pelo DDE entre outros dois módulos;

- b. o fato dos dutos serem paralelos favorece, também, o desempenho do SDF, pois evitam-se os sistemas de conversão paralelo/serial e serial/paralelo, reduzindo o tempo necessário para o reconhecimento de endereçamentos e comandos;
- c. em sistemas com dutos paralelos a depuração é facilitada, principalmente no caso de sistemas modulares onde um dos módulos seja dedicado a essa tarefa, como no caso do SDF;
- d. a organização lógica do SDF facilita tanto o envio de solicitações pelo DCAN como o reconhecimento e atendimento;
- e. O SDF não permite interconexão entre módulos de longa distância, por utilizar dutos paralelos. No caso dos esquemas com duto serial as interconexões podem ser de longas distâncias.

1.3 - *Sobre a Implementação do SDF*

A implementação do sistema como um todo deverá ser obtida em etapas posteriores do projeto, pois está vinculada ao desenvolvimento de outros trabalhos relacionados com os módulos do SDF.

Os pontos abordados neste trabalho devem servir como um primeiro passo no planejamento do sistema e as diretrizes apresentadas devem direcionar a sua implementação.

1.4 - *Sobre Pontos Relevantes do Trabalho*

Como pontos relevantes do trabalho destacam-se:

- a. a proposta do SDF como um todo, por reunir um conjunto de idéias e possibilidades para o desenvolvimento e implementação de um sistema distribuído não convencional;
- b. o DCAN, que permite a comunicação de alto nível entre os módulos do SDF, a partir de conceitos básicos presentes na maioria dos dutos paralelos empregados atualmente;

- c. o esquema de superposição no tempo e a definição dinâmica do código de prioridade, empregados na arbitragem;
- d. a implementação simples do sistema de interface, permitindo a utilização de versões simplificadas como a apresentada no exemplo do capítulo V;
- e. o conjunto de comandos de alto nível sugerido para o SDF, bastante abrangente, podendo facilitar grandemente o desenvolvimento do sistema operacional de cada módulo.

2 - A CONTINUIDADE DO TRABALHO

Os grupos de Instrumentação Eletrônica (IFQSC-USP) e de Hardware e Software Básico (ICMSC-USP) pretendem dar continuidade ao projeto do SDF através do desenvolvimento de trabalhos que concorram para a sua implementação final.

Atualmente encontram-se em fase de desenvolvimento trabalhos envolvendo:

- a. estudo sobre uma arquitetura para o Módulo de Memória Virtual, com inclusão de um sistema de cache, monitoração e proteção dedicados a cada módulo do SDF;
- b. desenvolvimento e implementação de uma arquitetura para o Módulo de Disco, incluindo um sistema de gerenciamento de base de dados;
- c. estudo de um sistema operacional para um módulo do SDF, compatível com o CP/M;
- d. estudos relacionados com o desenvolvimento do Módulo de Monitoração do Sistema.

Com relação a este trabalho, sugere-se que num estágio posterior deva-se efetuar a implementação do DCAN conectando-se alguns módulos com as características estipuladas para o SDF e que se defina, de acordo com o tipo de versão do sistema, o duto a ser empregado para o DDE.

BIBLIOGRAFIA

- A181 - Allison,A.A. - Status Report on the P896 Backplane Bus, IEEE-MICRO, FEB.81.
- Ba84 - Balakrishnan,R.V. - The Proposed IEEE 896 Futurebus - A Solution to the Bus Driving Problem, IEEE-MICRO, AUG. 84.
- Be80 - Bender,L. - ZBI: A System Bus for the Z8000, Mini-Micro Systems, JUN.80.
- Bo80 - Boberg,R.W. - Proposed Microcomputer System 796 Bus Standard, COMPUTER, OCT.80.
- Bo81 - Borril,P.L. - Microprocessor Bus Structures and Standards, IEEE-MICRO, FEB.81..
- Bo82 - Boberg,R.W. - Major Standartization Issues of the Proposed, IEEE 796 Bus - Multibus, MICROPROCESSORS and MICROSYSTEMS, NOV.82.
- Bo82a - Borril,P.L. - Backplane Bus Standards - Why We Need Them, What We Have Got and Who Makes Them, MICROPROCESSORS and MICROSYSTEMS, NOV.82.
- Bo82b - Borril,P.L. - IEEE P896 - The Futurebus Project MICROPROCESSORS and MICROSYSTEMS, NOV.82.
- Bo84 - Borril,P.L. - An Advanced Communication Protocol for the Proposed IEEE 896 Futurebus, IEEE-MICRO, AUG.84.
- Ch80 - Chu,W.W. et al - Task Allocation in Distributed Data Processing, COMPUTER, NOV.80.
- C178 - Clark,D.D. et al - An Introduction to Local Area Networks, Proc.IEEE, NOV.78.

- De82 - De Bock,R. - Versabus - Multiprocessor bus Standard-
and VMEbus - its Eurocard counterpart, MICROPROCESSORS
and MICROSYSTEMS, NOV.82.

- Di82 - Digital Equipment Corporation - Introduction to DECnet,
JAN.82.

- E182 - Elsmore,T. - Standard Bus for 8-bit Microprocessor
Systems, MICROPROCESSORS and MICROSYSTEMS, NOV.82.

- En78 - Enslow,P.H. - What is a "Distributed Data Processing
System", COMPUTER, JAN.78.

- Fa83 - Fathi,E.T. and Krieger,M. - Multiple Microprocessors
Systems: What, Why and When, COMPUTER, MAR.83.

- Ga82 - Garetz,M. - P696/S100 - A Bus Which Supports a Wide
Range of 8-and 16-bit processors, MICROPROCESSORS
and MICROSYSTEMS, NOV.82.

- Gu84 - Gustavson,D.B. - Computer Buses - A Tutorial, IEEE-
MICRO, AUG.84.

- Ha80 - Hansen,S. - ORed Busing Features Distributed Control,
ELETRONIC DESIGN, AUG.80.

- Hi82 - Hill,J. - Eurobus - the UK - Designed General-Purpose
Backplane Bus, MICROPROCESSORS and MICROSYSTEMS, NOV.
82.

- Ka81 - Kaplinsky,C. - Decentralizing uP Bus Grows Easily From
16 to 32 Bits, ELECTRONIC DESIGN, NOV.81.

- Le78 - Levy,J.V. - Buses, The Skeleton of Computer Structures,
in (Bell,C.G. et al - ed). Computer Engineering - A
DEC View of Hardware Systems Design - Cap. 11, DIGITAL
PRESS, 1978.

- Ry81 - Ryan,R. et al - Intel Local Network Architecture, IEEE-MICRO, NOV.81.
- So81 - Soltis,F.G. - Design of a Small Business Data Processing System, COMPUTER, SEP.81.
- Sp77 - Spetz,W.L. - Microprocessors Networks, COMPUTER, JUL. 77.
- St79 - Stewart,R.G. - Standard Specification for S-100 Bus Interface Devices, COMPUTER, JUL.79.
- St84 - Strack,J. - Sistemas de Processamento Distribuído, Livros Técnicos e Científicos Editora S/A, 1984.
- Ta76 - Taub,D.M. - Contention - Resolving Circuits for Computer Interrupt Systems, Proc. IEEE, SEP.76.
- Ta82 - Taub,D.M. - Worst-Case Arbitration Time in S-100-Type Computer Bus Systems, Electronics, SEP.82.
- Ta84 - Tarouco,L.M.R. - Redes de Comunicação de Dados (3a. edição), Livros Técnicos e Científicos Editora S/A,1984.
- Ta84a - Taub,D.M. - Arbitration and Control Acquisition in the Proposed IEEE 896 Futurebus, IEEE-MICRO, AUG.84.
- Zu81 - Zuffo,J.A. - Microprocessadores: Dutos de Sistemas, Técnicas de Interface e Sistemas de Comunicação de Dados, Ed. Edgard Blücher Ltda., 1981.

APÊNDICE

A - ABREVIATURAS EMPREGADAS

ADM	- (DMA) Acesso Direto à Memória
DCAN	- Duto de Comunicação de Alto Nível
DDE	- Duto de Dados e Endereços
EM	- Escaninho de Mensagem
IEEE	- Institute of Electrical and Electronics Engineers
LSI	- Large Scale Integration
SDF	- Sistema Distribuído Funcional
SMD	- Sistemas com Microprocessadores Distribuídos
SOp	- Sistema Operacional
SPDD	- Sistemas de Processamento de Dados Distribuídos
UCP	- Unidade Central de Processamento

B - GLOSSÁRIO

ARBITRAGEM - (contention) processo de escolha do próximo mestre para o duto, no qual apenas um dos módulos requisitantes do duto é atendido, conforme prioridades pré-estabelecidas.

BARRAMENTO - (Backplane) linhas de interconexão entre módulos de um sistema computacional.

CACHE - memórias de pequeno porte e alta velocidade, utilizadas em computadores, para armazenar, temporariamente, porções da memória que estão (ou esperam-se que estejam) frequentemente em uso.

CIRCUITO ACOPLADOR - (Buffer) circuito utilizado para o acoplamento entre linhas de sinais, fornecendo na saída sinais com características adequadas à aplicação.

CIRCUITO RETENTOR - (LATCH) circuito utilizado para armazenar temporariamente um dado a ser utilizado.

COLETOR ABERTO - tipo de circuito lógico que permite que dois ou mais dispositivos tenham suas saídas interligadas, implementando uma lógica de fiação do tipo "E".

COMANDO DE ALTO NÍVEL - referente aos comandos que circulam no DCAN e que se caracterizam como de alto nível, pelo acionamento das tarefas (ou funções) pré-estabelecidas no SDF.

DISTRIBUIÇÃO FUNCIONAL - tipo de distribuição do processamento, onde se estabelecem conjuntos de tarefas ou funções a serem atendidas, em cada módulo do sistema.

ENCADEAMENTO EM CASCATA - (Daisy Chain) tipo de implementação do processo de arbitragem, com características híbridas (centralizado e distribuído), onde a prioridade é fixada pela posição relativa do elemento na cadeia de interconexão.

ESCANINHO DE MENSAGEM - (Mailbox) estrutura de dados utilizada para o armazenamento de linhas de informação.

ESCRAVO - elemento de um sistema computacional capaz de participar de uma transação através do duto, por solicitação.

ção de um mestre.

EUROCARD - (e EUROCARD duplo) tipo de cartão de circuito impresso padronizado na Europa.

HANDSHAKE - tipo de sinal de sincronismo utilizado na implementação de sistemas assíncronos.

INTERRUPÇÃO - sinal enviado a um processador solicitando sua atenção para um atendimento de periférico, falha, recepção de dados, etc..

MESTRE - elemento de um sistema computacional capaz de iniciar e controlar uma transação através do duto, solicitando a participação de um ou mais escravos.

MODULARIDADE - medida do grau de flexibilidade na composição de sistemas, através da compactação e isolamento dos seus elementos.

PROCEDIMENTO - (ROTINA) conjunto de instruções elementares que implementam uma função básica no sistema, podendo ser utilizado em várias situações.

SISTEMA DE INTERFACE - dispositivos que promovem a comunicação entre cada módulo de um sistema e a correspondente via de interligação.

SISTEMA OPERACIONAL - conjunto de procedimentos (ou rotinas) básicos que coordenam o funcionamento de um sistema computacional.

TAREFA - conjunto de procedimentos ou rotinas adequadamente relacionadas, com o objetivo de desempenhar uma dada função.

TRI-ESTADO (tri-state) tipo de circuito lógico capaz de fornecer não apenas níveis lógicos 0 e 1, mas também permanecer em um estado de alta impedância, permitindo que dois ou mais circuitos possam ter suas saídas interligadas.

C - ROTINAS DESENVOLVIDAS PARA A IMPLEMENTAÇÃO DESCRITA EM V-2

```

;***** INICIALIZACOES *****

070B      APDIS EQU 070BH
0558      RBYTE EQU 0558H

          ORG 4400H
4400' 00   FILA.DE_ATENDIMENTO: DB 00
4401' 00   DB 00

          ORG 4300H
4300' 00   SINALIZA_ERRO_ATENDIMENTO: DB 00

          ORG 4BDAH
4BDA'      END_ATEN_65:
4BDA' C3 4200'      JMP ATENDINT ; desvia para rotina de atendimento
                                ; da interrupcao 6.5

; Rotina Para Inicializacao do Sistema
; -----

ORG 4000H

4000' 3E 83      INI: MVI A,83H ; CONTROLE 8255
4002' D3 17      OUT 17H ; programacao da interface

4004' 3E E0      MVI A,0E0H ; valores iniciais das linhas de controle
4006' D3 16      OUT 16H

4008' 3E 80      MVI A,80H ; valor inicial do sinal de recebimento
400A' D3 14      OUT 14H

400C' 3E 00      MVI A,00H ; Mascara de Interrupcao
400E' 30         SIM
400F' FB         EI ; ativacao da interrupcao 6.5

4010' 3E 01      MVI A,01 ; valor inicial da linha de arbitragem
4012' D3 18      OUT 18H

4014' C9         RET ; fim da inicializacao

```



```
; Rotina Para Arbitragem do DCAN
; -----
```

```
ORG 4030H
```

```
4030' DB 16      ARB:  IN    16H    ; Verificacao da liberacao do processo
4032' E6 08      ANI    08      ;                               de arbitragem
4034' CA 4030'   JZ     ARB

4037' 3E C0      MVI    A,0C0H ; Sinalizacao de Requisicao do DCAN
4039' D3 14      OUT    14H

403B' DB 15      RES:  IN    15H    ; Verificacao do Resultado da arbitragem :
403D' E6 80      ANI    80H
403F' C2 404C'   JNZ    BLOQ      ; Se ganhou desvia para BLOQ
4042' DB 16      IN     16H      ; Se NAO obteve o duto, mas o processo
4044' E6 08      ANI    08      ;                               nao terminou
4046' C2 403B'   JNZ    RES      ; retorna a RES para verificacao
4049' C3 4030'   JMP     ARB      ; Se perdeu tenta novamente

404C' 3E 00      BLOQ: MVI    A,00   ; Bloqueia Nova Arbitragem
404E' D3 18      OUT    18H

4050' 3E 80      MVI    A,80H    ; Libera Requisicao
4052' D3 14      OUT    14H

4054' CD 4050'   CALL   COMUN    ; Inicia Uso do DCAN

4057' C9        RET
```

```
; Rotina Para Comunicacao Pelo DCAN
; -----
```

```
ORG 4050H
```

```
4050' CD 0708    COMUN: CALL   APDIS ; limpa display
4053' CD 0558    CALL   RBYTE ; le um byte do teclado
4056' 21 4500    LXI     H,4500H ; (HL) <-- 4500

4059' 77        MOV     M,A      ; (4500) <-- byte lido
405A' CD 0708    CALL   APDIS ; limpa display
405D' CD 0558    CALL   RBYTE ; le um novo byte
4060' 2C        INR     L        ; (L) <-- (L) + 1
4061' 77        MOV     M,A      ; (4501) <-- novo byte

4062' CD 0708    CALL   APDIS ; limpa display

4065' 4E        MOV     C,M      ; (C) <-- (4501)
4066' 2D        DCR     L        ; (L) <-- (L) - 1
4067' 46        MOV     B,M      ; (B) <-- (4500)

4068' CD 4100'   CALL   ENVIO    ; Rotina de Envio de informacao pelo DCAN

406B' C2 406F'   JNZ     ERRO1    ; Desvia para sinalizar o ERRO

406E' C9        RET              ; (A) = 0 --> SUCESSO
```

406F'	3D	ERR01:	DCR	A	
4070'	C2 4075'		JNZ	ERR02	
4073'	3C		INR	A	
4074'	C9		RET		; (A) = 1 --> ERRO no segundo ciclo
4075'	3C	ERR02:	INR	A	
4076'	C9		RET		; (A) = 2 --> ERRO no primeiro ciclo

; Rotina de envio de informacoes pelo DCAN
; -----

ORG 4100H

4100'	DB 16	ENVIO:	IN	16H	; espera disponibilidade do DCAN
4102'	E6 04		ANI	04	
4104'	CA 4100'		JZ	ENVIO	
4107'	3E 60		MVI	A,60H	; bloqueio do DCAN
4109'	D3 16		OUT	16H	
410B'	3E 01		MVI	A,01	; liberacao da arbitragem
410D'	D3 18		OUT	18H	; primeira fase no DCAN
410F'	78		MOV	A,B	; segunda fase no DCAN
4110'	CD 4150'		CALL	ESCREVE	; rotina que escreve no DCAN
4113'	C2 411E'		JNZ	SAIDA	; erro na segunda fase
4116'	79		MOV	A,C	; terceira fase no DCAN
4117'	CD 4150'		CALL	ESCREVE	
411A'	CA 411E'		JZ	SAIDA	; sucesso no envio
411D'	3C		INR	A	; erro na terceira fase
411E'	57	SAIDA:	MOV	D,A	; salva (A)
411F'	3E 70		MVI	A,70H	; desacoplado porta A, do DCAN
4121'	D3 16		OUT	16H	
4123'	3E F0		MVI	A,0F0H	; liberacao do DCAN
4125'	D3 16		OUT	16H	
4127'	7A		MOV	A,D	; recupera (A)
4128'	B7		ORA	A	; criando flag
4129'	C9		RET		; fim do envio de informacoes pelo DCAN

```
; Rotina que escreve no DCAN
; -----
```

ORG 4150H

```
4150'      ESCRREVE:
4150'  F6 80      ORI    80H    ; acertando bit A7
4152'  D3 14      OUT    14H    ; escreve informacao na interface

4154'  3E 70      MVI    A,70H  ; acopla porta A ao DCAN
4156'  D3 16      OUT    16H

4158'  3E 30      MVI    A,30H  ; sinal de validacao
415A'  D3 16      OUT    16H

415C'  16 FF      MVI    D,0FFH ; loop de espera pelo sinal de reconhecimento
415E'      ESPERA:
415E'  DB 16      IN     16H
4160'  E6 01      ANI    01
4162'  CA 4171'   JZ     FIMESC ; se (A) = 0 recebido
4165'  15        DCR    D
4166'  C2 415E'   JNZ    ESPERA

4169'  3E 70      MVI    A,70H  ; tira validade
416B'  D3 16      OUT    16H

416D'  3E 01      MVI    A,01   ; sinalizando o erro
416F'  B7        ORA    A       ; criando flag

4170'  C9        RET          ; fim da escrita com erro

4171'      FIMESC:
4171'  3E 70      MVI    A,70H  ; tira validade
4173'  D3 16      OUT    16H
4175'  AF        XRA    A       ; zera acumulador com flag

4176'  C9        RET          ; fim da escrita com sucesso
```

```
; Rotina Para Atendimento da Interrupcao 6.5
; -----
;
; Recebimento de Comando pelo DCAN
; -----
```

ORG 4200H

```
4200'      ATENDINT:
4200'  F3        DI          ; inibe interrupcoes
4201'  F5        PUSH    PSW   ; salva estado do sistema
4202'  C5        PUSH    B
4203'  D5        PUSH    D
4204'  E5        PUSH    H
```

4205'	DB 16	IN	16H	; verifica validade
4207'	E6 02	ANI	02	
4209'	C2 4256'	JNZ	TRATERR1	; tratamento do Erro ocorrido
420C'	DB 15	IN	15H	; leitura do DCAN
420E'	E6 1F	ANI	1FH	
4210'	57	MOV	D,A	; armazena o dado lido
4211'	3E C0	MVI	A,0C0H	; sinal de reconhecimento
4213'	D3 16	OUT	16H	
4215'	3E 02	MVI	A,2	; espera antes de retirar o sinal
4217'		R1ECONH:		
4217'	3D	DCR	A	; de reconhecimento
4218'	C2 4217'	JNZ	R1ECONH	
421B'	3E E0	MVI	A,0E0H	; retira o sinal de reconhecimento
421D'	D3 16	OUT	16H	
421F'	3E 00	MVI	A,00	; bloqueia a interface para novo recebimento
4221'	D3 14	OUT	14H	
4223'	06 FF	MVI	B,0FFH	; espera por um novo sinal
4225'		ESPVAL:		
4225'	DB 16	IN	16H	; de validade dos dados no DCAN
4227'	E6 02	ANI	02	
4229'	CA 4233'	JZ	CONT	
422C'	05	DCR	B	
422D'	C2 4225'	JNZ	ESPVAL	
4230'	C3 425E'	JMP	TRATERR2	;tratamento do erro ocorrido
4233'	DB 15	CONT: IN	15H	; leitura do DCAN
4235'	E6 1F	ANI	01FH	
4237'	5F	MOV	E,A	; armazena o dado lido
4238'	3E C0	MVI	A,0C0H	; sinal de reconhecimento
423A'	D3 16	OUT	16H	
423C'	3E 02	MVI	A,02	; espera antes de retirar o sinal
423E'		R2ECONH:		; de reconhecimento
423E'	3D	DCR	A	
423F'	C2 423E'	JNZ	R2ECONH	
4242'	3E E0	MVI	A,0E0H	; retira o sinal de reconhecmneto
4244'	D3 16	OUT	16H	
4246'	3E 80	MVI	A,80H	; libera a interface para novo reconhecimento
4248'	D3 14	OUT	14H	
424A'	21 4400	LXI	H,4400H	; armazena dados recebidos na fila
424D'	72	MOV	M,D	; de atendimento
424E'	2C	INR	L	; (4400) (--- dados da segunda fase
424F'	73	MOV	M,E	; (4401) (--- dados da terceira fase

4250'	E1	FIM:	POP	H	; recupera estado inicial do sistema
4251'	D1		POP	D	
4252'	C1		POP	B	
4253'	F1		POP	PSW	
4254'	FB		EI		; libera interrupcoes
4255'	C9		RET		; fim do atendimento da interrupcao

4256'		TRATERR1:			
4256'	21 4300		LXI	H,4300H ; (4300) (-- tipo do erro ocorrido	
4259'	36 01		MVI	M,01	
4258'	C3 4250'		JMP	FIM	

425E'		TRATERR2:			
425E'	21 4300		LXI	H,4300H	
4261'	36 02		MVI	M,02	
4263'	C3 4250'		JMP	FIM	