UM SISTEMA DE COMUNICAÇÃO PARA AMBIENTE DE MULTIPROCESSADORES

Eliane Martins

Departamento de Engenharia de Computação - DCA
Diretoria de Engenharia e Tecnologia Espacial - ETE
Instituto de Pesquisas Espaciais - INPE
Avenida dos Astronautas 1758 - Jardim da Granja
12201 - São José dos Campos - São Paulo

RESUMO

O sistema de comunicação apresentado é parte de um sistema distribuído, projetado para o Multiprocessador de Comunicação em Redes(MCR) desenvolvido pelo INPE/MCT. Este sistema provê a comunicação entre processos através de canais. Um canal permite a transferência de dados entre processos cooperantes independentemente do processador em que eles se localizem. O MCR, com aplicação prevista para controle de interfaces seriais de comunicação, foi projetado para dar suporte à implementação dos níveis mais baixos de um protocolo e rotinas auxiliares de roteamento.

ABSTRACT

The communication system presented in this paper is part of a distributed system designed for the Multiprocessor for Network Communication (MCR) developed at INPE/MCT. This system provides the means for interprocess communications through channels. A channel permits the transfer of data between cooperating processes, independently of where these processes are being executed. The MCR, which purpose is the control of interfaces from serial communication, was designed to support the implementation of the lower levels of a communication protocol and auxiliary routing routines.

1 - INTRODUÇÃO

O Multiprocessador de Comunicação em Redes - MCR - é um equipamento de arquitetura distribuída com aplicação prevista para o controle de interfaces seriais para comunicação síncrona ou assíncrona, implementando para isto os níveis mais baixos de protocolos de comunicação e rotinas auxiliares de roteamento.

O sistema operacional desenvolvido para dar suporte ao tipo de aplicação descrito anteriormente caracteriza-se como um sistema distribuído, modular e totalmente independente da arquitetura da rede de comunicação [3]. O sistema de comunicação apresentado é parte deste sistema operacional e tem como objetivo tornar a estrutura física (hardware) do MCR transparente para a aplicação.

Uma descrição geral do sistema operacional é apresentada na Seção 2. A arquitetura do MCR é descrita resumidamente na Seção 3, o sistema de comunicação é mostrado na 4 e, finalmente, na Seção 5, é apresentado o estado atual do sistema.

2 - O SISTEMA OPERACIONAL

O sistema operacional do MCR é constituído de um núcleo e de um sistema de comunicação que são executados em cada processador. O núcleo dá suporte à exeucução de processos e à comunicação local a um processador. O sistema de comunicação é constituído de processos e rotinas responsáveis pelo controle dos recursos locais a cada processador.

O núcleo cria em cada processador um ambiente de processamento concorrente, em tempo real, e permite a comunicação local entre processos através da troca de mensagens. O núcleo mantém um descritor que contém para cada processo, entre outras informações, o seu estado e sua prioridade. Um processo pode assumir um dos seguintes estados: pronto - quando está apto a ser executado; em execução - quando está de posse do processador; suspenso - quando está aguardando, por um período de tempo específico, um serviço do núcleo, de outro processo ou a ocorrência de um evento; bloqueado - quando o processo aguarda um serviço do núcleo, de outro processo ou a ocorrência de um evento, sendo colocado em estado de pronto somente através de um comando específico. A escolha do processo a ser executado é baseada no seu estado atual e na prioridade: o processo pronto de mais alta prioridade obtém o acesso ao processador

O núcleo é dirigido por eventos. Desta forma um processo permanece em execução até que:

- necessite dos serviços do núcleo ou de um outro processo;
- termine sua exeução;
- a ocorrência de um evento externo faça com que seja escolhido para execução um processo de maior prioridade.

Os processos têm acesso aos serviços do núcleo através da chamada de rotinas, denominadas primitivas. Algumas destas primitivas são: ENVIA (permite o envio de mensagens para outros processo, e o emissor pode ficar bloqueado até que o receptor tenha tratado a mensagem), RECEBE(permite a recepção de mensagens enviadas por outros processos), SUSPENDE (permite a um processo suspender a sua execução por um período de tempo), ATIVA (permite passar um processo do estado de suspenso ou bloqueado para pronto), ESPERAINT (permite a um processo esperar a ocorrência de uma interrupção), TRATAINT (permite a um processo passar ao estado de pronto ou execução devido à ocorrência de uma interrupção).

O sistema de comunicação, descrito mais detalhadamente na Seção 4, trata do gerenciamento dos recursos locais de um processador.

3 - ARQUITETURA DO MCR

O MCR é formado por vários processadores que trabalham no esquema mestre-escravo, cabendo ao processador mestre a supervisão da comunicação entre os demais processadores (escravos). Estes, por sua vez, são dedicados ao controle das interfaces seriais existentes. A arquitetura do MCR, mostrada na Figura 1, é constituída por três tipos de módulos: Supervisor (SV), Porta Interna (PI) e Porta Externa (PE).

O SV tem como função controlar o fluxo de dados no MCR e o funcionamento dos demais módulos (PI e PE). A tarefa de controlar o fluxo de dados internamente faz com que o SV assuma o controle total do barramento interno do MCR, denominado MCRBUS. É através deste barramento que são feitas as transferências de dados: PE $\longleftrightarrow$PE e PE $\longleftrightarrow$PI. A comunicação com o SV se dá através de um conjunto de registros localizados nas PEs e na PI, denominados registros de entrada e registros de saída.

A PI é a interface entre o MCR e os demais integrantes do nó, tendo três vias de acesso: CPUBUS, que é a via de acesso utilizada pelo SV; MCRBUS, que permite a conexão com as PEs, e MCR-C-BUS, que permite a conexão com os demais integrantes do nó (podendo inclusive ser outro MCR). Este módulo não tem processador próprio, sendo utilizado unicamente para armazenar temporariamente os dados que fluem de/para o MCR.

A PE é o módulo que realiza a principal função do MCR: a comunicação de dados na rede. É na PE, portanto, que é feita a implementação do protocolo que opera na sub-rede. Através das linhas seriais ligadas às PEs podem ser efetuadas conexões com outros nós, com hospedeiros da sub-rede ou com outros equipamentos.

Cada módulo, com exceção da PI, dispõe de um processador de 8 bits do tipo INTEL 8085 - A (6 MHz), de um barramento local (CPUBUS), memórias EPROM e RAM, temporizador programável. A menos do SV, os módulos contêm uma interface para o MCRBUS com uma unidade de DMA e os registros de entrada e de saída, entre outros. As PEs contêm ainda as interfaces seriais que podem ser de dois tipos: uma atende às necessidades dos protocolos de comunicação mais sofisticados, tais como HDLC, SDLC e X.25; opera em "full-duplex" ou "half-duplex" com até 64K de "baud-rate", geração e verificação automática de CRC (Cyclic Redundancy Check), orientado a bit, realiza a comunicação síncrona e assíncrona e possui comandos ao nível de quadros; a outra é do tipo USART e tem como características as operações síncrona e assíncrona, operando em modo "full-duplex" e "half-duplex", orientada a "byte", porém não tem circuito para geração/verificação de CRC e nem reconhece quadros.

Maiores informações sobre a estrutura física do MCR podem ser obtidas nas referências [1] e [2]. Uma proposta de arquitetura tolerante a falhas para o MCR pode ser vista na referência [4].

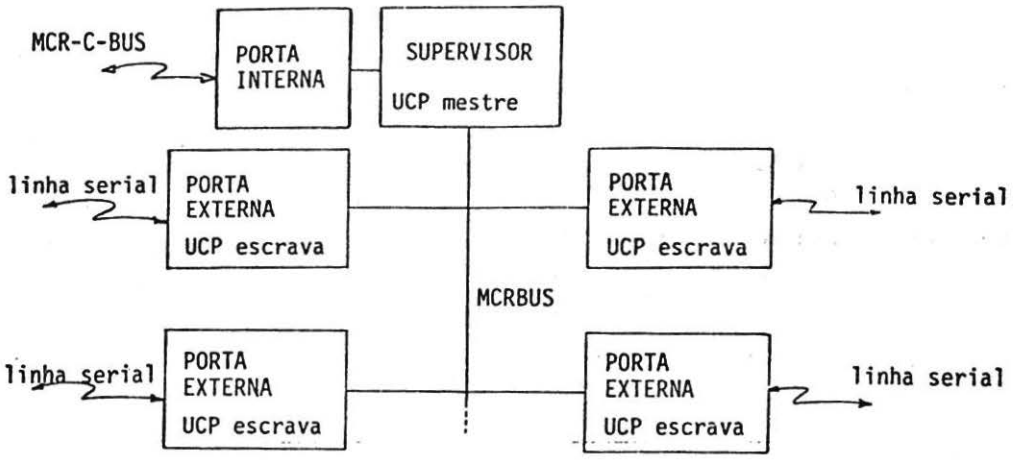

Fig. 1 - Arquitetura do MCR.

4 - O SISTEMA DE COMUNICAÇÃO

O sistema de comunicação está estruturado, em cada processador, nos seguintes níveis:

- nível físico: define as forma de acesso aos meios físicos de comunicação, quais sejam: barramento interno (MCRBUS), barramento externo (MCR-C-BUS) e linhas seriais;
- nível de transferência: trata da transferência de dados através dos meios físicos de comunicação;
- nível de aplicação: oferece aos processos aplicativos um conjunto de serviços de comunicação de dados.

4.1 - COMUNICAÇÃO ENTRE PROCESSOS

A comunicação entre processos se dá através de uma entidade lógica denominada canal. Um canal interliga dois processo que se comunicam, independente

mente do processador em que eles residam. A cada canal está associado um número, que o identifica perante os processos aplicativos. Vários processos podem enviar dados através de um canal, mas somente um processo pode receber dados através de le. Os canais são estabelecidos na fase de geração do sistema e permanecem ativos, a menos que haja um problema no meio físico de comunicação.

Os processos aplicativos têm acesso a um canal através da chamada de rotinas do nível de aplicação, denominadas comandos de acesso a um canal. Estes comandos estão listados na Tabela 1.

TABELA 1

COMANDOS DE ACESSO

ENV(cn, ms) Permite a um processo enviar a mensagem ms através do canal cn.
REC(cn, ms) Permite a um processo receber a mensagem ms do canal cn.
ENVB(cn, ms) Permite a um processo enviar a mensagem ms através do canal cn, permanecendo bloqueado até que a mensagem tenha sido tratada.
RECB(cn, ms) Permite a um processo receber a mensagem ms do canal cn, permanecendo bloqueado caso não haja mensagens.

4.2 - PROTOCOLO INTERNO

O protocolo interno é utilizado no nível de transferência para a troca de mensagens entre processadores e também para prover um controle de fluxo por canal.

Toda a comunicação interna ao MCR é realizada com o auxílio do SV. O SV atua como intérprete para a conversa entre dois módulos quaisquer do MCR. Todo o processo de comunicação interna é baseado em interrupções.

O protocolo interno utiliza dois tipos de informação: os comandos e as mensagens. As mensagens contêm um cabeçalho e os dados, e o seu tamanho máximo é estabelecido durante a inicialização do sistema. As mensagens são transferidas por DMA, com os dados fluindo da memória do módulo fonte até a memória do módulo destino. Portanto, é necessária uma troca de informações para estabelecer este caminho direto entre os módulos envolvidos, ou seja, o SV e os módulos fonte e destino devem habilitar as suas interfaces para o MCRBUS através de troca de informações, de tal forma que os dados lidos da memória do módulo fonte fiquem disponíveis ao barramento local do módulo destino para ser armazenados na sua memória. As informações trocadas para este fim constituem os comandos do protocolo interno. Estes comandos são trocados entre os módulos (PEs e SV) através de registros de entrada e registros de saída, mencionados na Seção 3. Quando uma PE escreve nos seus registros de saída, é gerada uma interrupção no SV; quando o SV escreve nos registros de entrada de uma PE, esta PE é interrompida. Os comandos do protocolo interno podem ocupar no máximo 4 bytes, e estão listados na Tabela 2.

Os procedimentos necessários para a transferência dos dados através do MCRBUS são descritos a seguir. Quando um processo aplicativo em um processador P1 emite um comando de envio de dados através de um canal C1 (ENV(C1, MS)ou ENVB(C1,MS)), o SV é avisado (através do comando TRM) e P1 fica aguardando a resposta do processador destino. Um temporizador em P1 controla o tempo de espera da resposta. Desta maneira pode ocorrer uma das seguintes situações: a chegada de um ACK, permitindo a transmissão; a chegada de NAK, indicando que o canal do processador destino não está ativo; um RNR, indicando que o canal não pode, temporariamente receber a mensagem; ou pode esgotar o tempo de espera pela resposta.

O SV atende um pedido de transmissão (TRM),de um processador fonte por vez. Caso mais de um processador envie pedidos, estes são enfileirados. Nesta fila só poderá haver um pedido por processador solicitante. Para atender ao pedido de transmissão do processador P1, através do canal C1, o SV consulta uma tabela (Tabela de Mapeamento de Canais) para descobrir o processador (P2) e o canal (C2) destinos. Em seguida o SV envia um pedido de recepção (REC) para P2 e aguarda a sua resposta. Caso a resposta seja afirmativa (ACK), o SV avisa o processador fonte (P1) e libera o barramento interno (MCRBUS) para transferência. Ao término da transferência o processador destino avisa o SV, emitindo um EOT (mensagem recebida sem erros) ou um NOT (mensagem recebida com erros). O SV repassa então a resposta ao processador fonte e passa a atender o próximo pedido.

Para não ocorrer uma parada no sistema devido ao mau funcionamento de algum módulo, as esperas de respostas e de término de transferência são temporizadas. Em caso de esgotar um tempo de espera, é solicitada uma diagnose (DIA).

TABELA 2

COMANDOS DO PROTOCOLO INTERNO

TRM - Processador fonte faz um pedido de transmissão para o SV.
REC - SV faz um pedido de recepção para o processador destino.
ACK - Processador destino avisa que está pronto para receber mensagens.
NAK - SV avisa que o processsador destino não pode receber mensagens.
RNR - Processador destino avisa que não pode, temporariamente, receber mensagens.
EOT - Processador destino avisa que a recepção terminou sem erros.
DIA - Pedido de diagnose.

Convém ainda lembrar que, caso um dos módulos envolvidos em uma transferência seja a PI, o SV é quem executa todos os procedimentos necessários.

5 - ESTADO ATUAL

O sistema está sendo desenvolvido para uma arquitetura do MCR que contém os seguintes módulos: SV, PI e até 5 PEs. O SV e cada PE contém os seguintes módulos: SV, PI e até 5 PEs. O SV e cada PE contém 56 Kbytes de memória local (sendo no mínimo 16 Kbytes de EPROM) e a PI tem 8 Kbytes de memória RAM.

O núcleo do sistema ocupa atualmente 3,5 Kbytes, o que inclui o código , área de dados e a pilha. O sistema de comunicação está parcialmente implementado, permitindo a comunicação interna do MCR.

A primeira versão do sistema não incluirá a comunicação externa do

MCR, ou seja, a conexão com outros nós, outros hospedeiros ou outros MCRs ficará a cargo dos processos explicativos.

O sistema encontra-se em fase de avaliação, visando possibilitar melhorias no desempenho e na ocupação de memória. Estuda-se também a possibilidade de introduzir outros mecanismos de controle de erro, além da temporização, usada atualmente, no sentido de tornar a comunicação interna mais confiável.

6 - AGRADECIMENTOS

Agradeço a Ana Maria Ambrosio e ao Sérgio Katsumi Oshiro o esforço e a dedicação, sem os quais não teria sido possível a implementação do sistema.

REFERÊNCIAS

[1] HASHIOKA, M. H.; "Módulo e Análise de uma Interface de Comunicação Distribuída para Aplicação em Redes de Comunicação por Comutação de Pacotes". Dissertação de Mestrado. São José dos Campos, INPE, 1983.

[2] SALGADO, A.E.M.; SANTOS JR., P.P.; QUEIRÓZ, J.S.M.; "MCR - Interface Hardware/Software". Relatório interno do DCA/INPE, 1986.

[3] MARTINS, E.: "SOPMCR - Um Sistema Operacional para um Multiprocessador de Comunicação em Redes". Relatório interno do DCA/INPE (em desenvolvimento).

[4] SALGADO, A.E.M. "Uma Arquitetura com Mecanismos de Tolerância a Falhas para o Multiprocessador de Comunicação em Rede - MCR". Dissertação de Mestrado. INPE-3933-TDL/227. São José dos Campos, 1985.

BIBLIOGRAFIA

- Bowen, B.A. e Buhr, R. J. A.; "The Logical Design of Multiple-Microprocessor Systems". Prentice-Hall Inc. 1980.

- Menasce, D.A. e Schwabe, D.; "Redes de Computadores. Aspectos Técnicos e Operacionais". Terceira Escola de Computação. Pontifícia Universidade Católica do Rio de Janeiro. 1982.

- Wicklund, T.NL.; "MINI-EXEC: A Portable Executive for 8-bit Microcomputers". Communications of the ACM. 25(11). 1982.

- Banino, J.S.; Fabre, J.C.; Guillemont, M.; Morissel, G.; Lazier, M.; "Some Fault-Tolerant Aspects of the Chorus Distributed System". Proceedings of the 5th. Conference on Distributed Computing Systems. Denver, Colorado. 1985.

- Zhongxiu, Sun Du, Zhang; Peigen, Yan; "ZCZOS: A Distributed Operating System for a LSI-11 Microcomputer Network". ACM Operating Systems Review, 17(3), July 1983.

- Gaglianello, Robert D.; "MEGLOS - An Operating System for a Multiprocessor Environment". Proceedings of the 5th. Conference on Distributed Computing Systems. Denver, Colorado, 1985.