

S-MICRO-SISTEMA PARA SIMULAÇÃO DE UNIDADE DE CONTROLE MICROPROGRAMADA

Ronaldo Luiz Dias Cereda
José Carlos Maldonado

Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq
Instituto de Pesquisas Espaciais - INPE
Departamento de Engenharia de Computação em Aplicações Espaciais - DCA
Caixa Postal 515 - São José dos Campos - SP - Brasil

Resumo

Propõe-se um sistema para desenvolvimento de sistemas digitais microprogramados, cujo objetivo principal é permitir a simulação integral da unidade de controle microprogramada. Procura-se com isso facilitar o desenvolvimento integrado da arquitetura da unidade de controle, do microprograma e do próprio sistema microprogramado em teste. Apresenta-se o sistema em detalhes, tanto do ponto de vista de "hardware" quanto de "software", bem como suas vantagens sobre outras proposições já existentes, destacando-se suas diferenças e algumas limitações.

Abstract

A system for development of microprogrammed digital systems is proposed, whose main design goal is to permit the whole simulation of the microprogrammed control unit. This way an effort is made to facilitate the integrated development of the control unit architecture, of the microprogram and even of the microprogrammed system under test. The system is introduced in detail both from the hardware and the software point of view. Its advantages over other existing propositions are also introduced, emphasizing its differences and a number of limitations.

1. INTRODUÇÃO

A técnica de microprogramação está sendo largamente aplicada no desenvolvimento de diversos equipamentos, nas mais diferentes áreas de aplicação de sistemas digitais. Algumas vantagens da implementação de circuitos de controle microprogramados em relação à implementação que utiliza lógica aleatória podem ser citadas (Amaral, 1979):

- 1 - Arquitetura melhor estruturada que resulta em uma redução da lógica aleatória na unidade de controle do sistema.
- 2 - A emulação de vários dispositivos (não somente processadores) pode ser feita por um único dispositivo de propósito geral alterando o conteúdo de sua memória de controle.
- 3 - Rotinas de diagnose e manutenção podem ser fácil e eficientemente implementadas na unidade de controle do sistema.
- 4 - Mudanças e adaptações específicas que se façam necessárias no sistema podem, em princípio, ser facilmente realizadas alterando o conteúdo da memória de controle.
- 5 - O desempenho de dispositivos microprogramados pode ser aumentado com custos menores que os custos para sistemas de lógica aleatória.

6 - Documentação e custos de treinamento e de manutenção podem ser reduzidos.

Pode-se perceber que as vantagens dos sistemas microprogramados sobre os enfoques convencionais dependem dos recursos disponíveis para desenvolvimento, teste e manutenção destes sistemas.

Os recursos para desenvolvimento de sistemas microprogramados disponíveis atualmente no DCA/INPE compreendem basicamente:

- 1 - Emulador de Memórias de Microcontrole Auxiliado por Computador (EMMAC): este sistema permite ao usuário depurar um microprograma durante a fase de teste e facilita a diagnose e manutenção do protótipo final (Amaral, 1979).
- 2 - Linguagem de Microprogramação (LMP): é uma linguagem para geração de microcódigos (Yamaguti, 1981).

Estes recursos formam o sistema LMP/EMMAC, que constitui um sistema completo para a geração e depuração de microprogramas.

No entanto o LMP/EMMAC é uma ferramenta de desenvolvimento - no que tange a microprogramas - restrita ao âmbito da memória de con

trole. Para sua utilização é necessário que toda a lógica de sequenciamento tenha sido implementada, pelo menos como protótipo. Esta característica torna mais difícil realizar as alterações necessárias na lógica de sequenciamento decorrentes da aplicação dos microprogramas no sistema microprogramado.

Neste sentido, o S-MICRO é um sistema de desenvolvimento para sistemas microprogramados, configurado parte em "hardware" e parte em "software", o qual propicia a simulação integral da Unidade de Controle Microprogramada, ou seja, o sequenciador, a memória de controle, a lógica de decodificação e mapeamento etc. Esta característica o coloca em vantagem sobre outras proposições por facilitar, em muito, redefinições e/ou alterações decorrentes da observação dos compromissos entre o microprograma e o sistema microprogramado durante a fase de especificação do sistema.

Outra vantagem desta proposição é a viabilidade de implementação, através do S-MICRO, de facilidades na execução dos testes dos microprogramas tais como: execução passo-a-passo, "breakpoints", etc que serão descritas na Seção 3.2.

2. ARQUITETURA DO S-MICRO

A idéia básica que norteou a especificação da arquitetura do S-MICRO é a de facilitar a interação do usuário com os recursos para desenvolvimento de sistemas microprogramados, bem como com o próprio sistema em teste. A ligação com um computador de grande porte (B-6800), no qual está residente o tradutor LMP, acrescenta facilidades na geração de microprogramas, além de permitir a utilização dos recursos de entrada e saída e de armazenamento de dados deste computador.

Uma visão geral do ambiente de utilização do S-MICRO pode ser encontrada na Figura 2.1.

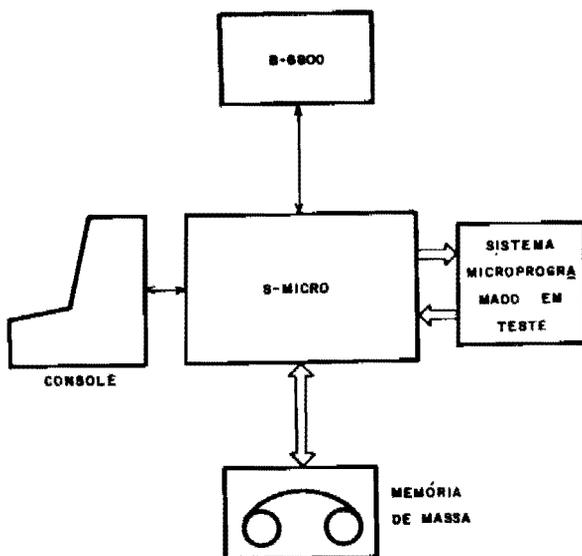


Fig. 2.1 - Ambiente de utilização do S-MICRO.

Do ponto de vista de "hardware", o S-MICRO é composto de sete unidades funcionais, conforme a Figura 2.2. O "software" do sistema, denominado "Software de Simulação" será descrito na Seção 3.

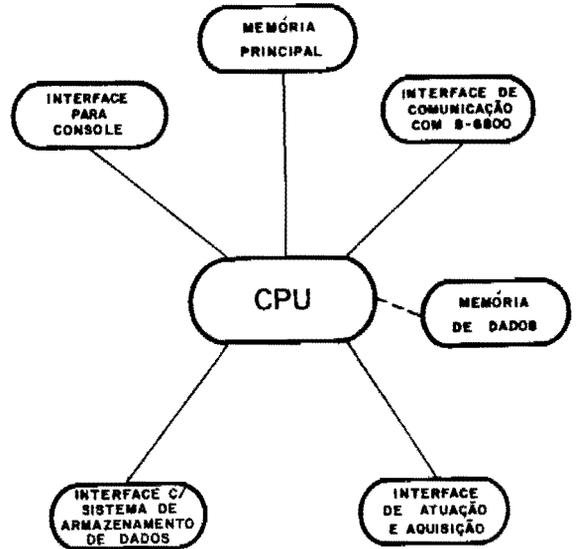


Fig. 2.2 - Unidades funcionais do S-MICRO.

O microcomputador de propósito geral ASTRO-L/V1, desenvolvido no DCA/INPE, provê as características funcionais adequadas para a configuração do S-MICRO, a menos da Interface de Atuação e Aquisição - que permite o interfaceamento com o sistema microprogramado - e da Memória de Dados, a serem descritos nas seções seguintes.

No ASTRO-L/V1 a unidade CPU é implementada com o processador de 16 bits TMS 9900 (Texas Instruments), com capacidade de endereçamento de memória de 64K "bytes" (32K palavras).

2.1 - CARACTERÍSTICAS DA INTERFACE DE ATUAÇÃO/AQUISIÇÃO (IAA)

Esta interface provê facilidades de "hardware" para atuação/aquisição nos circuitos digitais que compõem o sistema microprogramado. O diagrama de blocos da IAA é fornecido na Figura 2.3.

Os registros RI, PIPELINE e de ENTRADA podem ser carregados ou lidos através do barramento serial do ASTRO-L/V1. As entradas e saídas desta interface são conectadas aos circuitos do sistema microprogramado em teste, através de garras individuais que permitem uma maior flexibilidade na atuação/aquisição.

2.2 - ALOCAÇÃO DA MEMÓRIA DE CONTROLE

A memória de controle, onde está alocado o microprograma a ser depurado, é simulada utilizando parte da Memória Principal do ASTRO-L/V1, conforme a Figura 2.4. Esta filosofia acrescenta maior flexibilidade ao sistema, em aspectos como: o tamanho da Memória de

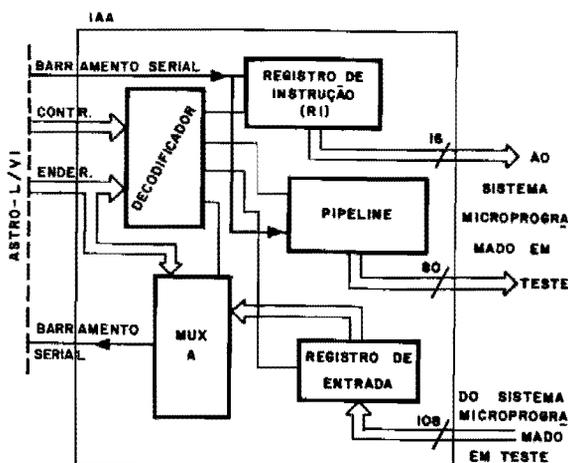


Fig. 2.3 - Diagrama de blocos da IAA.

Controle poderá ser definido pelo usuário; o comprimento da palavra de controle está limitado apenas pelo "hardware" da IAA, já que se podem utilizar posições consecutivas da Memória Principal do ASTRO-L/V1 para armazenar uma única microinstrução; e alterações do conteúdo da Memória de Controle poderão ser realizadas com maior facilidade.

Além destas facilidades, este enfoque possibilita maior aproveitamento da ligação existente com o B-6800, pois habilita o carregamento dos microcódigos gerados através do tradutor LMP residente naquele computador diretamente na Memória de Controle simulada.

A transferência das microinstruções da Memória de Controle para o registro PIPELINE é feita através do "Software de Simulação". Em cada vez são transferidas N palavras de posições contíguas da Memória Principal, que definem uma microinstrução, para o registro PIPELINE, onde $N \leq 5$. A microinstrução corrente fica estável neste registro enquanto execute as microoperações sobre os recursos de "hardware" do sistema microprogramado em teste; desta forma o registro assume funções idênticas aos tradicionais registros "pipeline" utilizados em projetos microprogramados.

2.3 - ALOCAÇÃO DAS MACROINSTRUÇÕES

As macroinstruções, que constituem o programa a ser executado pelo sistema microprogramado em teste, e que são interpretadas pelo microprograma e, conseqüentemente, executadas pelo "hardware" deste sistema, estão alocadas num trecho da Memória Principal do ASTRO-L/V1, denominado "Memória de Programa", conforme ilustrado na Figura 2.4.

Estas macroinstruções são transferidas pelo "Software de Simulação", sob o controle do microprograma, para o Registro de Instrução (RI). Este registro, localizado na IAA, tem 16 bits cujas saídas são ligadas diretamente ao "hardware" do sistema em teste.

É importante notar que todo o processo de

"fetch" da macroinstrução não é realizado em nível de microprograma, como seria usual numa máquina microprogramada. Isto se deve ao fato de que o registro RI, que normalmente integra o "hardware" da máquina microprogramada, está implementado na Interface de Aquisição/Atuação (IAA). Para que o processo de busca da próxima macroinstrução seja possível, é necessário que o "Software de Simulação" tenha acesso às informações de contexto do sistema microprogramado, como por exemplo: PC, registros de uso geral etc, através do REGISTRO DE ENTRADA da IAA.

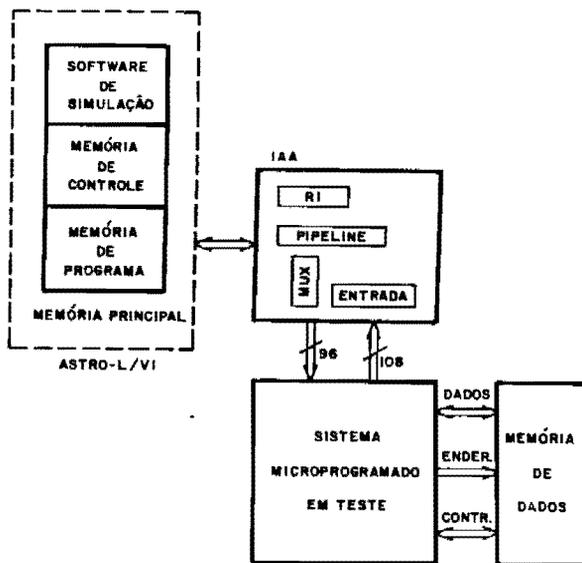


Fig. 2.4 - Alocação de memória no S-MICRO

Esta solução implica que toda a lógica de decodificação e mapeamento seja implementada também pelo "Software de Simulação", o que acrescenta uma flexibilidade ainda maior à depuração do microprograma e do próprio programa do sistema em teste.

Uma possível restrição de memória para armazenamento de programa não deve ser encarada como uma limitação do S-MICRO, pois o seu objetivo é fornecer subsídios para especificação de unidades de controle microprogramadas e de seus microprogramas, e não para desenvolvimento de programas a serem executados no sistema microprogramado. Portanto a área de "Memória de Programa" necessária é, em princípio, igual à cardinalidade do conjunto de macroinstruções da máquina microprogramada.

2.3.1 - ÁREA DOS DADOS ASSOCIADOS ÀS MACROINSTRUÇÕES

Na hipótese de haver dados associados à execução das macroinstruções, eles estarão alocados na unidade denominada "Memória de Dados", conforme pode ser visto na Figura 2.4.

O acesso a esta unidade é realizado, de maneira convencional pelo "hardware" do próprio sistema microprogramado, ou seja, através de linhas de Dados, Endereços e Controle. Esta divisão da Memória Principal do sis

tema microprogramado em "Memória de Dados" e "Memória de Programa" evidentemente só tem significado durante a fase de simulação da unidade de controle.

A não-existência da "Memória de Dados" acarretaria uma complexidade indesejável no "software" do S-MICRO: o "Software de Simulação", responsável pelo mapeamento e pelo sequenciamento das microinstruções, teria também de realizar a simulação das microinstruções de acesso à Memória Principal, o que seria altamente indesejável, pois desta forma as microinstruções deste tipo, que são parte bastante significativa do microprograma, não poderiam ser testadas e validadas, bem como os recursos de "hardware" associados.

2.4 - MANIPULAÇÃO DE BITS DE CONDIÇÃO E DE STATUS

O "Software de Simulação" realiza principalmente as funções de sequenciamento do microprograma. Estas funções exigem, em certos momentos, testes de informações de status e de "bits de condição" da máquina microprogramada para que decisões sejam tomadas adequadamente. Estas informações podem ser lidas pelo "Software de Simulação" através do REGISTRO DE ENTRADA da IAA.

Desta forma o Mux A da Figura 2.3 realiza, nestes momentos, funções semelhantes às dos "Mux de Condições" ("Conditional Mux") tradicionais em unidades de controle microprogramadas.

Como dito anteriormente outras informações são adquiridas através destes recursos (REGISTRO DE ENTRADA e Mux A), tais como PC, registros de uso geral etc., provenientes do sistema em teste.

3 - "SOFTWARE DE SIMULAÇÃO" DO S-MICRO (SS)

O S-MICRO é composto, do ponto de vista de "software", conforme já dito anteriormente, de um "Software de Simulação" (SS) cujas funções básicas são: aquisição, decodificação e mapeamento das macroinstruções, e sequenciamento das microinstruções.

Além destas funções básicas, o SS provê recursos de interação com o usuário através do console do sistema, bem como facilidades para a depuração de microprogramas e do próprio sistema microprogramado através da implementação de modos de operação, os quais serão descritos posteriormente. Além destas facilidades, foram especificados recursos de depuração também em nível do conjunto de microinstruções.

O SS está alocado na Memória Principal do ASTRO-L/V1, conforme Figura 2.4, e interage diretamente com a IAA.

O SS é implementado com o apoio de uma linguagem de alto nível ALGOL-M (Martins, 1976), cujo compilador reside no computador B-6800. A utilização de uma linguagem com estas características favorece o desenvolvimento estruturado e modular do SS. Este contexto torna altamente flexível o projeto da ló-

gica de sequenciamento da unidade de controle, que é simulada pelo SS, pois redefinições e alterações necessárias nesta lógica, resultantes dos procedimentos necessários nesta lógica, resultantes dos procedimentos de teste, são facilmente realizáveis.

Esta característica torna o S-MICRO um sistema de desenvolvimento e depuração de sistemas microprogramados diferente dos sistemas de desenvolvimento convencionais propostos até então. No entanto surge uma restrição de corrente deste enfoque, ou seja, a taxa de carregamento das microinstruções no registro de "pipeline" não é constante e, na maioria das vezes, não atende às necessidades de velocidade dos sistemas microprogramados em teste, o que impede a sua validação em tempo real. Deve-se ressaltar também que a flexibilidade conseguida através deste enfoque incorre na perda de paralelismo de microoperações obtido com a implementação em "hardware" do sequenciador.

3.1 - DESCRIÇÃO DOS MÓDULOS FUNCIONAIS BÁSICOS DO SS

O SS compõe-se basicamente dos seguintes módulos funcionais:

1 - Aquisição de Macroinstrução: realiza o "fetch" da macroinstrução com base nas informações lidas no REGISTRO DE ENTRADA e carrega-a no registro RI da IAA. A área de dados manipulada por este módulo é fundamentalmente a Memória de Programa (Figura 2.4).

2 - Decodificação e Mapeamento: realiza a decodificação do "opcode" da macroinstrução e utiliza o resultado desta decodificação para acessar o endereço de início do microprograma que interpreta esta macroinstrução. A estrutura de dados associada a este módulo é uma tabela que contém os endereços citados acima.

3 - Sequenciamento de Microinstruções: realiza repetitivamente a carga do registro PIPELINE com a microinstrução corrente, além de decodificar o tipo de microinstrução e buscar a próxima microinstrução. As áreas de dados associadas a este módulo são a Memória de Controle e os parâmetros lidos através do REGISTRO DE ENTRADA da IAA.

3.2 - FACILIDADES DE DEPURAÇÃO OFERECIDAS PELO SS.

Em todas as etapas envolvidas no processo de desenvolvimento e depuração de sistemas microprogramados, o S-MICRO oferece facilidades de monitoração através de amostragem de resultados e informações na tela do console. Estes resultados e informações podem ser especificados pelo usuário, bem como o próprio formato (modos de leitura) em que eles deverão ser fornecidos.

Visando fornecer maior potencial de depuração, foram definidos "Modos de Operação" selecionáveis pelo usuário. Estes "modos" diferem quanto à quantidade de microinstruções a serem executadas cada vez. Existem três modos, descritos a seguir:

1 - "Por microinstrução": Após a carga de uma microinstrução no registro PIPELINE, o operador poderá requerer a amostragem de resultados a partir dos modos de leitura oferecidos.

2 - "Por macroinstrução": O operador poderá solicitar a amostragem dos resultados ao final da execução da sequência de microinstruções que interpreta uma dada macroinstrução.

3 - "Por sequência de macroinstruções": Os dados serão listados na tela de acordo com o formato especificado pelo usuário, após a execução de um conjunto predefinido de macroinstruções.

Os "Modos de Leitura" que podem ser selecionados pelo usuário são:

1 - Listar de uma só vez todo o conteúdo do REGISTRO DE ENTRADA da IAA em formato predefinido, que contém informações gerais do contexto do sistema microprogramado.

2 - Informar o estado de um único bit do REGISTRO DE ENTRADA.

3 - Listar uma sequência de bits do REGISTRO DE ENTRADA, de tamanho variável especificado pelo usuário.

4. CONCLUSÃO

O S-MICRO encontra-se atualmente em fase de desenvolvimento, com o "hardware" totalmente configurado e com o SS já especificado numa primeira versão e programado na linguagem de alto nível, ALGOL-M.

A primeira aplicação deste sistema será no desenvolvimento do Processador PISB (emulação domicroprocessador SBP9900 da Texas Instruments). O "software" que simula a unidade de controle deste processador foi especificado e incorporado ao "Software de Simulação." O microprograma desta unidade está sendo desenvolvido com o auxílio da linguagem LMP e espera-se que esse sistema forneça meios suficientes para a validação e especificação conjuntas da arquitetura da unidade de controle, do microprograma e dos recursos de "hardware" do Processador PISB. Pode-se dizer que, sem o apoio de um sistema semelhante, o grau de dificuldade para a validação do Processador PISB seria muito maior.

Com a experiência já adquirida no desenvolvimento do S-MICRO, notou-se que um possível aperfeiçoamento do sistema proposto poderia ocorrer com o desenvolvimento de uma linguagem que possibilitasse a definição, de forma flexível, da arquitetura da unidade de controle através da definição, especificação e integração de módulos funcionais básicos que compõem normalmente uma unidade de controle microprogramada.

Os sistemas de desenvolvimento para sistemas microprogramados existentes hoje comercialmente (como o AmSYSTM 29 da Advanced Micro Devices ou os equipamentos da Step Engineering) diferem em essência do S-MICRO, uma vez que nestes sistemas a unidade de Con

trole Microprogramada é emulada ao nível do "Hardware" quase em sua totalidade, enquanto que no S-MICRO toda a lógica de sequenciamento é simulada ao nível do "Software" como já foi dito.

Não temos informação alguma sobre similares nacionais.

5. REFERÊNCIAS BIBLIOGRÁFICAS

Amaral, P.F.S., (1979). Emulador de Memórias de Microcontrole Auxiliado por Computador, INPE, São José dos Campos. Tese de Mestrado em Eletrônica e Telecomunicações. (INPE-1489-TDL/009).

Yamaguti, W., (1981). LMP, Uma Linguagem de Microprogramação, INPE, São José dos Campos. Tese de Mestrado em Eletrônica e Telecomunicações. (INPE-2031-TLD/049).

Martins, R.C.O., (1976). Algol-M - Uma Linguagem de Alto Nível para microcomputadores, INPE, São José dos Campos. Tese de Mestrado em Computação Aplicada.