

# DESENVOLVIMENTO E SIMULAÇÃO DOS MÓDULOS DE UM PROCESSADOR DE 16 BITS. Renan Salenave Troiano, João Gabriel Corsi Mansano, José Aparecido de Aguiar Viana. – Exatas – Ciência da Computação – Departamento de Exatas – Centro Universitário de Rio Preto – Unidade Universitária 3.

O limite entre o hardware (parte física) e o software (programas), ainda hoje não está bem definido, está constantemente se alterando.

Os estudos dos processadores (microarquitetura) atualmente é muito importante e complexo de se entender pela sua estrutura física, lógica e de como é fabricado. Os componentes CI's (Circuito Integrado), os tipos de barramentos de dados, os tipos de memórias, registradores, a Unidade Lógica e Aritmética (ULA) são imprescindíveis na construção de uma microarquitetura eficiente e ao mesmo tempo compatível com o mercado.

A construção final de uma microarquitetura é um problema muito complexo quanto ao tamanho físico, temperatura e eficiência, pois envolve uma quantidade muito grande de circuitos integrados, fios, barramentos, fonte de alimentação, etc. Este problema vem sendo tratado por várias áreas voltadas à Computação e Engenharia. Neste sentido, o conceito de processador está envolvido em vários blocos até chegar a sua construção final.

O foco principal deste estudo está sendo principalmente o desenvolvimento de todos os módulos e a simulação de cada um deles, sendo possível adquirir um entendimento detalhado de todos os processos determinantes na construção e simulações destes módulos com 16 bits de dados.

O desenvolvimento e a simulação dos módulos de uma arquitetura de um processador de 16 bits está sendo utilizado o software de construção lógica Circuit Maker 2000, uma vez, que a construção física dessa arquitetura se torne inviável devido ao custo, mas também por não permitir a visualização de cada bloco internamente, pois seu projeto já vem implementado em CI's com funções específicas.

O conjunto de métodos, regras e postulados utilizados no desenvolvimento deste projeto, foi principalmente adquirir um entendimento das primeiras arquiteturas existentes abrangendo assim todas as regras de combinações para a construção e simulação dos módulos separados de um processador de 16 bits. Todo esse conjunto de informações que foram coletadas sobre cada componente utilizado na microarquitetura tais como, registradores, barramentos, multiplexadores, decodificadores, ULA, deslocadores, relógios, memória principal, permitiu desenvolver módulos para cada tipo de componente. Os componentes por si só não têm sentido se não forem interligados por barramentos ou fios, ou seja, os componentes precisam estar interligados entre si desde as regras sejam seguidas para que eles possam trabalhar com uma informação.

Uma das peças principais e fundamentais na construção de um processador de 16 bits são os registradores, multiplexador, demultiplexador e a ULA.

A ULA por si só possui duas entradas e uma saída de dados, mas também possui algumas entradas e saídas de controle, executa a adição, subtração, e outras operações simples com suas entradas de dados, produzindo um resultado no seu registrador de saída. Este resultado pode ser armazenado em um registrador e, de lá, para a memória, se desejada, o armazenamento volta a um registrador ou novamente a memória, tudo isso é percorrido por um ou mais barramentos de interconexão com os módulos do processador e da memória. Na figura 1 tem-se um modelo de interconexão entre a ULA com registradores.

Os registradores como uns dos principais elementos estão conectados a um ou mais barramentos de entrada da ULA e a um ou mais barramentos de saída. O registrador consiste em dezesseis flip-flops tipo D, cada um conectado ao barramento de saída através de um buffer não-inversor, cada um armazena um bit.

O registrador possui dois sinais de controle: CK (Clock, que na realidade significa “carrega registrador”) e OE (**Output Enabled**, habilitação da saída), ambos conectados a todos os flip-flops. Normalmente ambos os sinais estão em seu estado desativado, mas ocasionalmente eles podem ser ativados, fazendo com que a ação correspondente dos valores armazenados aconteça.

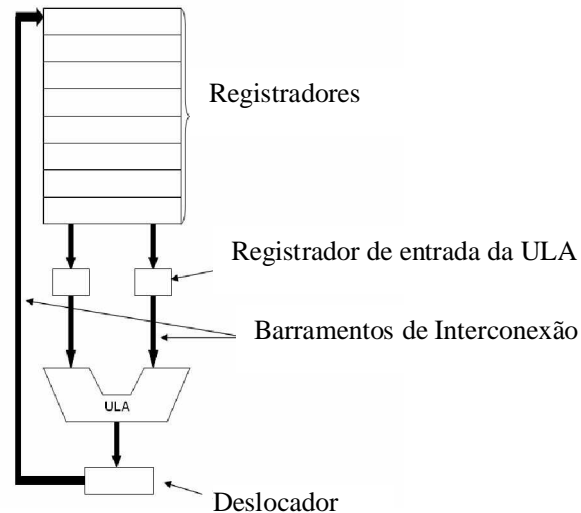


Figura 1: Esquema de interconexão entre os módulos centrais (Fonte: Tanenbaum, 1992).

Dois dos mais importantes circuitos combinatórios os multiplexadores, são responsáveis por receber uma ou mais linhas de entrada e calcular um ou mais valores de saída, são o multiplexador e o decodificador. Um multiplexador tem  $2^n$  entrada de dados (linhas individuais ou barramentos), uma saída de dados da mesma largura que as entradas e uma entrada que controlam  $n$  bits e seleciona uma das entradas e a direciona para a saída. Na figura 2 tem-se um modelo de decodificador de 4 bits.

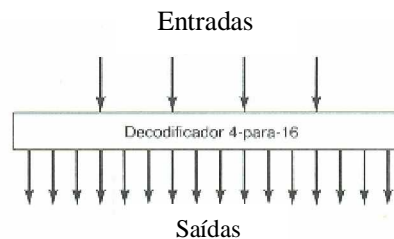


Figura 2: Modelo de decodificador (Fonte: Tanenbaum, 1992).

O inverso do multiplexador é o demultiplexador, que direciona sua entrada única para uma de suas  $2^n$  saídas, dependendo de um valor de suas  $n$  linhas de controle. O decodificador, que possui  $n$  linhas de entrada e  $2^n$  linhas de saída, numeradas de 0 a  $2^n - 1$ . O inverso de um decodificador é um codificador, que possui  $2^n$  de entrada e  $n$  saídas.

Os resultados já obtidos com a construção dos módulos como a ULA, deslocadores, multiplexadores, são resultados ainda que simbólicos, pois os testes estão sendo feitos separadamente, ou seja, cada módulo já construído são realizados diversos tipos de testes e simulações.

Portanto, como consequência das simulações dos módulos separados de uma arquitetura de 16 bits, será possível ter uma visualização mais detalhada do que está acontecendo com a informação, de como essa informação está sendo alocada nos registradores, ou seja, o que acontece afinal com os dados de entrada.

### Referências Bibliográficas

- 1 - TANENBAUM, A. S. Organização estruturada de computadores. 3a. ed. Rio de Janeiro: Prentice-Hall, 1992.
- 2 - STALLINGS, W. Arquitetura e organização de computadores. 5a. ed. São Paulo: Prentice-Hall, 2002.
- 3 - MURDOCCA, M. J. Introdução à arquitetura de computadores. São Paulo: Campus, 2001.