

KEYWORDS — MIMD, Taxonomia de Flynn, Arquitetura de Computadores, Processadores Paralelos

I. CLASSIFICAÇÃO TAXONÔMICA DE FLYNN

Em 1966, Michael J. Flynn em seu artigo “*Very high-speed computing systems*” propôs a primeira versão de um sistema de classificação de arquiteturas para computadores de “alta performance”, motivado pela diversidade dos computadores de alta performance voltados para aplicações científicas na época.

As quatro classificações iniciais definida por Flynn eram baseada na multiplicidade de recursos em hardware disponíveis para manipulação de fluxos de Dados e Instruções, sendo as quatro:

- Único Fluxo de Instruções-Único Fluxo de Dados (SISD)
- Único Fluxo de Instruções-Único Fluxo de Dados (SIMD)
- Múltiplos Fluxos de Instruções-Único Fluxo de Dados (MISD)
- Múltiplos Fluxos de Instruções-Múltiplos Fluxos de Dados (MIMD)

O sistema de classificação pode ser melhor entendido por meio da Figura 1.

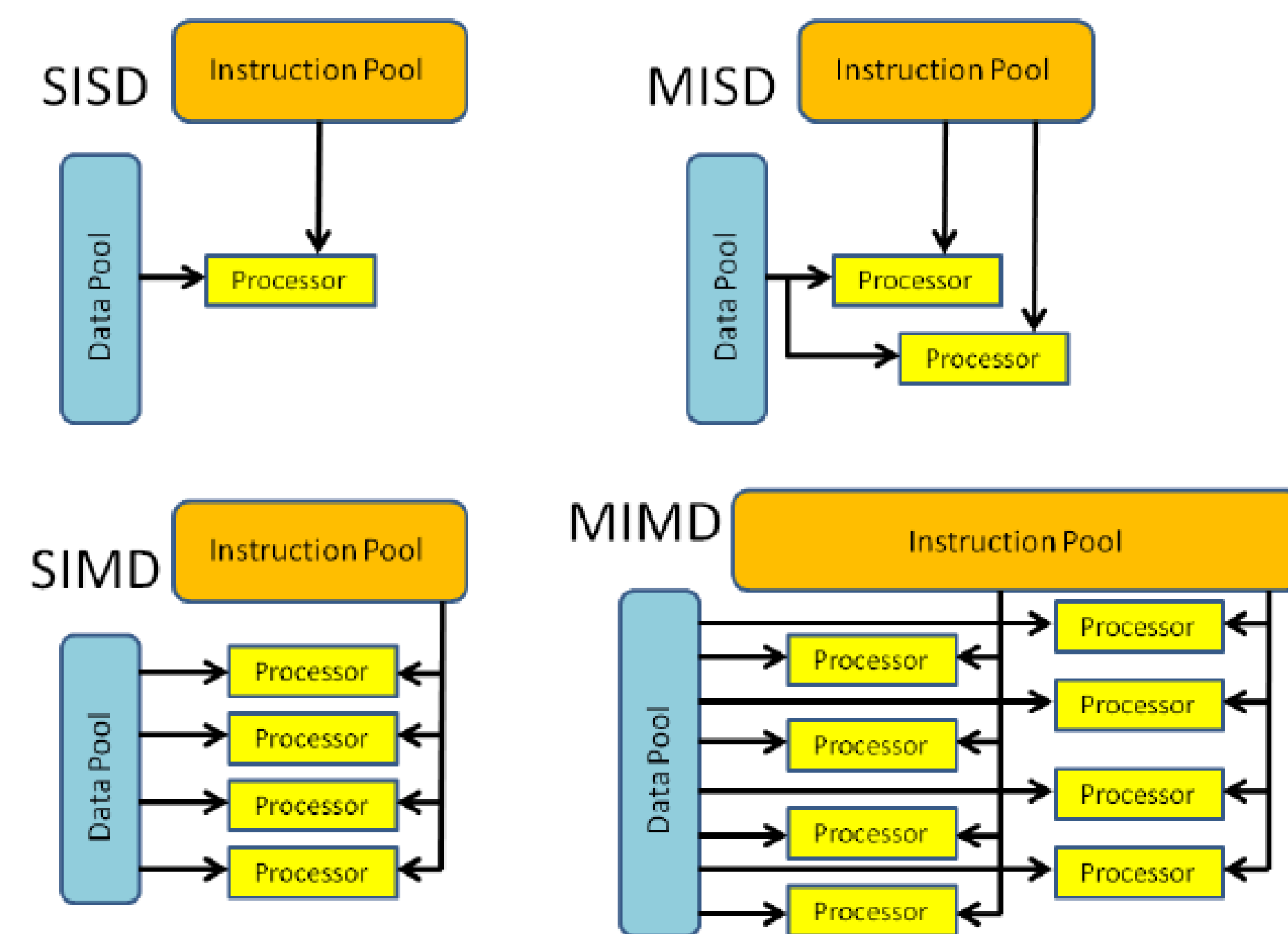


Figure 1: Diagrama esquemático das diferentes classes propostas por Flynn

Ressalta-se que o sistema proposto por Flynn, apesar da idade, ainda é utilizado como ferramenta para design de processadores modernos. Sua última alteração foi em 1972, no artigo “*Some Computer Organizations and Their Effectiveness*” onde Flynn propôs três sub-classificações para a Arquitetura SIMD: Processamento Matricial (SIMT), Processamento com Pipeline (*packed SIMD*), Processamento Associativo (*predicated/masked SIMD*).

II. MÚLTIPLOS FLUXOS DE INSTRUÇÕES, MÚLTIPLOS FLUXOS DE DADOS (MIMD)

A classe MIMD de arquitetura paralela é a forma mais familiar e possivelmente a mais básica de processador paralelo: consiste em múltiplas unidades de processamento (núcleos), cada um com sua unidade lógico aritmética (ULA), conjunto de registradores (REG) e unidade de controle (UC), interconectadas, como pode ser visto na Figura 2. Ao contrário dos processadores SIMD, cada CPU executa de forma completamente independente (embora, tipicamente, o mesmo programa seja executado). Embora não haja um requisito para que todas os núcleos sejam idênticas, a maioria das configurações MIMD são homogêneas, com todas os núcleos sendo idênticas.

Quando a comunicação entre os núcleos é realizada por meio de um espaço de endereçamento de memória compartilhada (seja global ou distribuída entre os núcleos, chamada de memória compartilhada distribuída para diferenciá-la da memória distribuída), dois problemas significativos surgem. O primeiro é manter a consistência da memória—os efeitos de ordenação visíveis ao programador das referências de memória, tanto dentro de um núcleo quanto entre diferentes núcleos. O segundo é manter a coerência de cache—o mecanismo invisível ao programador para garantir que todas os núcleos vejam o mesmo valor para uma determinada posição de memória. O problema da consistência de memória é geralmente resolvido através de uma combinação de técnicas de hardware e software. O problema da coerência de cache é geralmente resolvido exclusivamente por meio de técnicas em hardware, como a implementação de algum protocolo de coerência de cache, como o protocolo MESI, por meio de estratégias como *Snooping* de Barramento ou Sistema de Diretório.

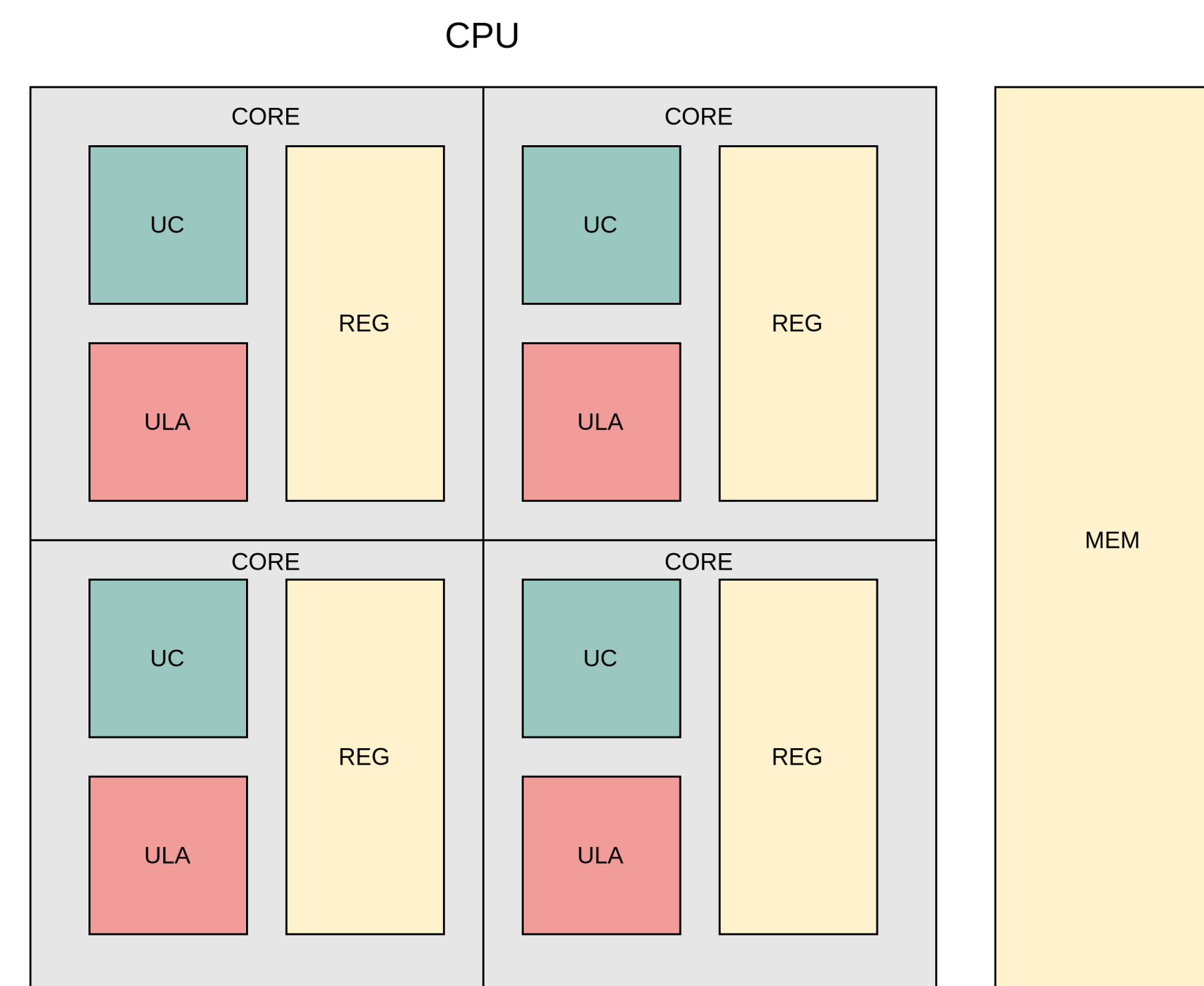


Figure 2: Diagrama esquemático da arquitetura MIMD - Autoral

III. USOS E APLICAÇÕES DE MÁQUINAS MIMD

- Exemplos de uso: simulações científicas, inteligência artificial, computação distribuída.
- Tendências futuras: maior uso em processamento paralelo e sistemas distribuídos.

IV. REFERÊNCIAS

- M. J. Flynn, “Very high-speed computing systems” in Proceedings of the IEEE, vol. 54, no. 12, pp. 1901-1909, Dec. 1966, doi: 10.1109/PROC.1966.5273.
- M. J. Flynn, “Some Computer Organizations and Their Effectiveness” in IEEE Transactions on Computers, vol. C-21, no. 9, pp. 948-960, Sept. 1972, doi: 10.1109/TC.1972.5009071.
- M. J. Flynn, K. W. Rudd, “Parallel architectures” in ACM Computing Surveys, vol. 28, pp. 67-70, 1996, doi: 10.1145/234313.234345