

Esta ficha foi elaborada para acompanhar o estudo da disciplina de Arquitetura de Computadores Avançada (a4s1) ou para complementar a preparação para os momentos de avaliação finais da mesma. Num segundo ficheiro poderás encontrar um conjunto de propostas de solução aos exercícios que estão nesta ficha. É conveniente relembrar que algum conteúdo destes documentos pode conter erros, aos quais se pede que sejam notificados pelas vias indicadas na página web, e que serão prontamente corrigidos, com indicações de novas versões.

**1.** Em relação ao paralelismo de instruções, qual é o interesse e a diferença do paralelismo de dados?

O paralelismo de instruções permite a exploração de instruções que, não sendo dependentes entre si, podem ser executadas em paralelo. O paralelismo de dados investiga a possibilidade de múltiplas fontes de dados serem processados em simultâneo.

**2.** O que é que caracteriza uma arquitetura vetorizada? Dá um exemplo de arquitetura vetorizada.

Uma arquitetura vetorizada caracteriza-se por manipular conjuntos de elementos de dados espalhados sobre a memória, transferindo-os para bancos de registos grandes e sequenciais dispersando, no fim, os resultados de volta à memória. Assim, uma única instrução opera em vetores de dados, permitindo dezenas de operações de registos para registos em elementos de dados independentes.

Um exemplo de aplicação é o VMIPS. Estas arquiteturas, especialmente no caso da VMIPS, possuem registos-vetor (8 registos de 64 elementos de 64 bits), unidades funcionais vetoriais e unidades de load/store vetoriais.

**3.** Compara o funcionamento das unidades de interlocking e de forwarding de uma arquitetura MIPS com uma VMIPS.

Uma diferença importante é a frequência de interlocks do pipeline que, enquanto que no MIPS ocorreriam uma vez por cada iteração, no VMIPS, cada instrução vetorizada só parará pelo primeiro elemento de cada vetor, sendo que os outros elementos fluirão normalmente pelo pipeline. Assim, os stalls no pipeline só são requeridos uma vez por instrução vetorial, ao invés de uma vez por elemento. Às operações de forwarding damos o nome de chaining, isto porque as operações dependentes estão encadeadas umas com as outras.

**4.** De que é que depende o tempo de execução das operações vetorizadas?

O tempo de execução das operações vetorizadas depende essencialmente em três fatores: o tamanho dos vetores-operando, o número de hazards estruturais e as dependências de dados entre operações sucessivas.

**5.** Dentro do contexto as arquiteturas vetorizadas, explica o que é um convoy (comboio) e qual é a sua importância a tornar um programa mais rápido.

Um convoy é um conjunto de instruções de vetor que podem ser executadas, potencialmente, todas juntas. As instruções contidas num convoy não devem assim conter quaisquer razões para a existência de hazards estruturais. Se tais hazards estivessem presentes, então necessitariam de ser serializados e iniciados em convoys. Este conceito torna um programa mais rápido porque permite uma maior paralelização de instruções e de dados num programa, principalmente com o chaining ativo, o que garante que as instruções sejam executadas todas juntas.

**6.** Que papel é que o registo-vetor máscara (vector mask register) tem em ciclos vetoriais?

Os ciclos vetoriais permitem que seja feita a paralelização. Por exemplo, se tivermos um ciclo for que itera 10 vezes, podemos executá-lo em 2 ciclos apenas, considerando que 5 unidades executam os ciclos originais de 5 em 5 iterações e que não há iterações que possuem condições que alterem o fluxo de execução. Considerando, pelo contrário, um ciclo de 10 iterações com uma condição if que altere o significado do ciclo na iteração  $n$  (sendo que  $0 \leq n \leq 10$ ), então não podemos paralelizar o ciclo normalmente. Esta é a razão principal pela qual se usam os vetores máscara, para identificar comportamentos diferentes de casos em casos, marcando o estado atual de paralelização no vector mask register.