



|  |              |                                     |   |
|--|--------------|-------------------------------------|---|
| <b>Universidade de Itaúna</b>              |              | <b>Curso:</b> Ciência da Computação | <b>Disciplina:</b> Arquitetura e Organização de Computadores II |
| <b>Professor (a):</b> Adriana Dornas       |              |                                     | Ano: 2021   |
| 3º Período                                 | Turno: Noite | <b>CIU:</b><br>82148                | Atividade relativa à aula de<br><b>22/03/2021</b>               |
| <b>Nome:</b> Davi Ventura Cardoso Perdigão |              |                                     |   |
| <b>Revisão AOC II</b>                      |              |                                     |   |

### 1 -

- **Simplicidade é favorecida pela regularidade:** Instruções de tamanho fixo; poucos formatos de instruções; opcode sempre utiliza os primeiros 6 bits.
- **Quanto menor, mais rápido:** Repertório de instruções limitados; quantidade de registradores limitados; número reduzido de modos de endereçamento.
- **Torne rápido o caso mais comum:** Existência de instruções que contém operandos.
- **Um bom projeto requer boas escolhas (compromissos):** Diferentes formatos de instruções complicam a decodificação, contudo permite instruções de tamanho fixo.

**2 -** A unidade de controle (UC) de um microprocessador assume toda a tarefa de controle das ações a serem realizadas pelo computador, comandando todos os demais componentes de sua arquitetura. É ela que deve garantir a correta execução dos programas e a utilização dos dados corretos nas operações que as manipulam.

**3 -** Por muito tempo falou-se que o clock era um indicativo soberano da velocidade dos processadores (quanto maior o clock, mais rápido o processador é). Porém, com o tempo, essa afirmação foi se mostrando desatualizada e sem embasamento nenhum do que ocorre, de fato, na prática. Visto que os processadores de gerações e fabricantes diferentes usam arquiteturas diversas, é impossível comparar a velocidade deles apenas com a frequência do clock. Devido a arquitetura, um processador com clock interno mais baixo pode ser mais rápido que um outro chip que apresente o clock mais alto.

Porém, é importante ressaltar que a frequência do clock é sim, parâmetro de comparação de velocidade, porém apenas se levarmos em conta processadores da mesma linha.

**4** - Pipeline de execução no contexto de microprocessadores, nada mais é do que uma técnica inicialmente usada em processadores RISC, que consiste em dividir o processador em vários estágios distintos. Quando é carregada uma nova instrução, ela primeiramente passa pelo primeiro estágio, que trabalha nela durante apenas um ciclo de clock, passando-a adiante para o segundo estágio. A instrução continua então sendo processada sucessivamente pelo segundo, terceiro, quarto e quinto estágios do processador.

**5** - Usando a analogia da lavanderia, supomos que cada etapa possa ser realizada em 30 minutos:

- 1) Colocar a roupa na máquina de lavar;
- 2) Depois de lavada, colocá-la na máquina de secar roupa;
- 3) Depois de seca, passar a ferro;
- 4) Depois de passada, arrumá-la no armário.

Para lavar, secar, passar e arrumar 5 cestos de roupa, sem uso do pipeline, levaríamos **10 horas** para concluir a tarefa. Contudo, podemos iniciar a lavagem de um cesto de roupas a cada 30 minutos, até que tenhamos 4 cestos sendo lavados simultaneamente, um em cada etapa do “pipeline”. Depois das primeiras **2 horas**, teremos um cesto de roupa lavada, passada e arrumada a cada 30 minutos. Ao final do dia teremos lavado muito mais cestos de roupa do que sem o uso de “pipeline”. Para aprontar 5 cestos de roupa, por exemplo, levaríamos apenas **4 horas** no total.

Com o uso da técnica de “pipeline”, a lavagem de um cesto de roupas continuará levando **2 horas** para ser realizada. Ou seja, a técnica de “pipeline” não melhora o tempo de execução de cada tarefa individualmente, mas melhora o rendimento ou a produtividade de todo o sistema.

**6** - Os Hazards de Dados ocorrem quando o pipeline precisa ser interrompido porque uma etapa precisa esperar até que outra seja concluída. A solução mais viável para esse tipo de risco é baseada na observação de que não precisamos esperar que a instrução termina antes de tentar resolver o hazard de dados. O acréscimo de hardware extra para ter o item que falta antes do previsto, diretamente dos recursos internos, é chamado de forwarding e bypassing.

**7** - Hazards Estruturais acontecem quando duas ou mais instruções sobrepostas no pipeline tentam usar o mesmo recurso de hardware ao mesmo tempo. Isso pode ser simplesmente resolvido com o uso de *bolhas* no pipeline. Isso significa que é possível inserir “*bolhas*” (instruções sem operações) no pipeline e atrasar a execução da instrução atual até que o operando requerido seja computado corretamente e escrito no registrador. Esse método diminui a eficiência do pipeline, que é totalmente o oposto da idéia do próprio pipeline, mas mesmo assim, é uma estratégia válida porque reduz o custo de memória quase que pela metade em uma máquina que suporta acesso à memória e às instruções a cada estágio do pipeline para evitar o hazard, por exemplo.

**8** - Como estudo em sala de aula, um dos mais comuns problemas de pipeline são os de desvio de fluxo. Esses problemas são os Hazards de Controle e são

causados por desvios e outras instruções que alteram o PC (Program Counter). Hazards de Controle podem ser tratados dos seguintes modos: Inserção de bolhas; nunca tomar o desvio ou sempre tomar o desvio. Desses três métodos de tratamento dos Hazards de Controle, o mais comum, assim como para qualquer outro hazard é o uso de *bolhas*. Nesse caso o pipeline é parado até que seja computado o estágio de decisão de fluxo, evitando quaisquer tipos de penalidades e perdas maiores de desempenho.

**9** - Um computador RISC (Reduced Instruction Set Computer) parte do pressuposto de que um conjunto simples de instruções vai resultar numa Unidade de Controle simples, barata e rápida.

**10** - Os computadores CISC (Complex Instruction Set Computer) visam criar arquiteturas complexas o bastante a ponto de facilitar a construção dos compiladores, assim, programas complexos são compilados em programas de máquina mais curtos. Com programas mais curtos, os computadores CISC precisariam acessar menos a memória para buscar instruções e seriam mais rápidos.

**11** - Como citado anteriormente, as arquiteturas RISC visam Unidades de Controle mais simples, rápidas e baratas, elas geralmente optam por instruções mais simples possível, com pouca variedade e com poucos endereços. Pelo motivo dos processadores RISC trabalharem com menos instruções e dados por vez, é o que possibilita a utilização predominante de registradores.

Os registradores estão localizados no núcleo do processador. São caracterizados por um tempo de acesso menor que um ciclo de relógio e sua capacidade é da ordem de centenas de bytes. O controle de qual informação deve estar nos registradores é feita explicitamente pelo compilador, que determina quais variáveis serão colocadas no registrador. É o único nível da “hierarquia de memória” que permite movimentações iguais apenas ao tamanho da informação desejada.

**12** - A melhor resposta sobre os desempenhos que acho é de que depende do uso que se quer exigir do processador.

Processadores RISC geralmente resultam em projetos menores, mais baratos e que consomem menos energia. Isso torna-os muito interessante para dispositivos móveis e computadores portáteis mais simples. Já os processadores CISC trabalham com clock muito elevado, são mais caros e mais poderosos no que diz respeito a desempenho. Entretanto, eles são maiores e consomem mais energia, o que os torna mais indicados para computadores de mesa e notebooks mais poderosos, além de servidores e computadores profissionais.

Os processadores CISC iniciaram com processadores mais simples e depois foram incorporando mais funcionalidades. Os fabricantes, como a Intel e a AMD, precisavam sempre criar novos projetos, mas mantendo a compatibilidade com as gerações anteriores. Ou seja, os computadores líderes mundiais em competições de desempenho computacional utilizam processadores CISC. Já o foco dos processadores RISC está na simplicidade e previsibilidade. Além do benefício da previsibilidade do tempo de execução ao Pipeline, ele também é muito interessante

para aplicações industriais. Mesmo que os processadores RISC sejam mais lentos do que os CISC, eles são mais utilizados nessas aplicações críticas e de tempo real, como aplicações industriais, de automação e robótica.

**13** - Exceções são semelhantes a chamadas de procedimento, assim, o controle deve ser desviado para uma rotina de tratamento. Por exemplo, se uma instrução add \$1, \$2, \$1 produzir overflow aritmético. Após a instrução, o desvio será tomado e um flush no pipeline será causado nos estágios ID e EX (ao contrário dos branches que causam o flush no estágio IF).

**14** - Thread é um pequeno programa que trabalha como um subsistema, sendo uma forma de um processo se autodividir em duas ou mais tarefas. É o termo em inglês para Linha ou Encadeamento de Execução. Essas tarefas múltiplas podem ser executadas simultaneamente para rodar mais rápido do que um programa em um único bloco ou praticamente juntas, mas que são tão rápidas que parecem estar trabalhando em conjunto ao mesmo tempo.

**15** - A Unidade de Controle de um processador é responsável por gerar todos os sinais que controlam as operações no exterior do CPU, e ainda por dar todas as instruções para o correto funcionamento interno do CPU,. A unidade de controle executa três ações básicas, intrínsecas e pré programadas pelo próprio fabricante do processador. São elas : busca (fetch), decodificação e execução.

**16** - B) Falso.

**17** - A) uma única CPU (Central Processing Unit), pode conter múltiplas ALUs (Arithmetic Logic Unit).

**18** - B) Pipeline.

**19** - A) 5 estágios: busca de instruções, decodificação, execução, acesso à memória e gravação em registradores.

**20** - D) dividir a execução da instrução em várias partes, sendo cada uma delas manipuladas por unidades dedicadas do hardware que trabalham em paralelo.

**21** - A) Busca de instrução, decodificação de instrução e busca do operando.

**22** - A) Verdadeiro.

**23** - A) A técnica Pipelining permite ao processador executar múltiplas instruções paralelamente em estágios diferentes.

**24 - C)** Somente III é verdadeiro.

**25 - C)** O número e as funções de registradores variam de projetos de processadores.

**26 - E)** As afirmativas 1, 2 e 3 são verdadeiras.

**27 - B)** Geração de Clock.

**28 - A)** Podem ser executadas em paralelo pelo processador.

**29 - A)** Verdadeiro.

**30 - C)** I, II e III.

**31 - A)** Estrutural; dados; controle.

**32 - B)** Falso.

**33 - B)** Errado.

**34 - B)** Errado.

**35 - B)** Errado.

**36 - B)** Falso.

**37 - A)** Armazenar o código da instrução corrente em execução pelo processador.

**38 - B)** Controle.

**39 - B)** São pequenas áreas de armazenamento temporário usadas pela unidade de controle.

**40 - D)** Contador de programa (PC - program counter).

**41 - A)** Certo.

**42 - B)** Errado.

**43 - B)** propósito geral.

**44** - A) Arquitetura MIPS.

**45** - B) Executa cada instrução em um ciclo de clock.