Lista 2 – Arquitetura e Organização de Computadores

1. **Quais as diferenças de um processador multiciclo em relação a um uniciclo?**

O processador Uniciclo é mais simples, possui dois somadores/ULAs e duas memórias. Faz com que todas as instruções funcionem em um único clico de clock limitado a mais longa instrução(lw) , ou seja todas as instruções levarão o mesmo tempo para serem executadas, inclusive as mais rápidas como, isso de desconsiderarmos o tempo das unidades funcionais.

O processador multiciclo trabalha com as execuções em passos onde cada passo é executado em um ciclo de clock podendo levar menos tempo para executar as instruções, porém necessita de mais multiplexadores e usa mais registradores para que sejam armazenados os valores intermediários.

2. **Quais as modificações necessárias em um processador multiciclo simples para que se introduza a função de pipeline?**

Para que um processador multiciclo tenha a função de pipeline ele precisa que o processador tenha incluído em si os 5 passos para implementação do pipeline que são:

1. Instruction Fetch – Onde é feita a leitura da instrução;

2. Intruction decode and register fetch – Onde é feita a decodificação e os registradores a serem usados são identificados;

3. Execute – Onde é feita a execução da instrução;

4. Memory acess – Se é necessário o acesso da memória em algum momento ele é feito nessa etapa;

5. Write Back – Nessa etapa o resultado do processo é devolvido a memória, onde é feita a escrita do resultado.

3. **No programa abaixo, relacione as dependências (dados, WAR, WAW e outros) e conflitos existentes para execução em um processador MIPS usando pipeline.**

1.div.d F1, F2, F3 RAW – Só será feita a leitura e o armazenamento da resposta em F1 da divisão após a escrita e o resultado da subtração na linha 2.  
2.sub.d F4, F5, F1 WAW – Só será feita a escrita após a escrita nas linhas 3 e 5.   
3.s.d F4, 4(F10) WAW - Só será feita a escrita após a escrita da linha 5.  
4.add.d F5, F6, F7 RAW – Primeiro será feito o cálculo usando F5 na linha 5  
5.div.d F4, F5, F6

4.

a) **Write trough:** Quando o sistema escreve para um zona de memória que está contida no cache, escreve a informação, tanto na linha específica do cache como na memória ao mesmo tempo. Possui um desempenho inferior ao processo de write back devido à grande largura de banda que utiliza, o constante acesso a memória e a escrita lenta. Porém é mais simples de implementar e mais consistente pois o cache nunca está dessincronizado com a memória, diferente do write back.

b) **Write back:** A CPU escreve os dados diretamente no cache, cabendo ao sistema a escrita posterior da informação da memória principal. Dessa maneira , a CPU fica livre mais rapidamente para executar outras operações. Em contrapartida, a latência do controlador pode induzir problemas na consistência de dados na memória principal, em sistemas multiprocessados com memória compartilhada. As vantagens de usar o write back são: a escrita ocorre na velocidade do cache, escritas múltiplas de um endereço requerem apenas uma escrita na memória e consome menos largura de banda, porém em contraposto suas desvantagens são: a dificuldade de implementação, o fato de nem sempre haver a coexistência dos arquivos na memória e no cache e a leitura de blocos de endereço no cache podem resultar em escritas de blocos de endereços “dirty” na memória.

c) **Localidade temporal:** se em um ponto uma localização de memória específica for referenciada, é provável que a mesma localização seja referenciada novamente em um futuro próximo. Há uma proximidade temporal entre as referências adjacentes ao mesmo local de memória. Nesse caso, é comum fazer esforços para armazenar uma cópia dos dados referenciados em um armazenamento de memória mais rápido, para reduzir a latência das referências subsequentes. Localidade temporal é um caso especial de localidade espacial (veja abaixo), a saber, quando a localização potencial é idêntica à localização atual.

d) **Localidade espacial:** se um local de armazenamento específico for mencionado em um horário específico, é provável que locais de memória próximos sejam referenciados em um futuro próximo. Nesse caso, é comum tentar adivinhar o tamanho e o formato da área em torno da referência atual para a qual vale a pena preparar um acesso mais rápido para as referências subsequentes.