Lista 1

1. Ordem do endereço do desvio no caminho de dados
    - Faz-se a extensão de sinal do imediato, que representa o deslocamento a partir do PC+4
    - Multiplica-se o deslocamento por 4 deslocando-o dois bits à esquerda
    - Soma-se o deslocamento ao PC+4
2. Ordem dos passos de execução de uma instrução do tipo I num caminho de dados monociclo
    - Decodifica a instrução, determina, a partir do opcode, qual instrução deve ser executada
    - Acessa o banco de registradores e lê o registrador rs, que é o endereço base. Se for uma instrução de escrita (store), lê ainda o rt. Ao mesmo tempo, faz a extensão do imediato de 16 para 32 bits
    - Calcula o endereço da memória a ser acessado usando a ULA, passando como entrada o registrador rs e o imediato estendido (base+offset)
    - Acessa a memória de dados, fazendo a operação de leitura ou escrita
    - Se for uma instrução de leitura, escreve o dado lido no banco de registradores no registrador rt
3. Ordem dos passos de execução de uma instrução tipo R num caminho de dados monociclo
    - Decodifica a instrução, determinando, a partir do opcode, funct e shamt, qual instrução deve ser executada
    - Recupera o valor dos registradores rs e rt
    - Encaminha os dados dos registradores para a ULA a fim de executar a operação aritmética adequada
    - Envia o resultado da ULA para o banco de registradores no registrador rd
4. Associando atividades:
    - fetch -> Obtenha a instrução da memória no endereço armazenado em PC
    - decode -> A partir da palavra obtida, determine a instrução que deve ser executada e os dados que devem ser utilizados na execução
    - execute -> Use os elementos adequados para realizar a operação determinada pela instrução

6. Elementos de um caminho de dados: Contador de programa, banco de regs, memória de instruções, memória de dados e somadores
7. Os registradores usados nas operações, especificamente aqueles registradores a serem lidos (apenas lidos, não escritos), estão sempre nas mesmas posições de bit de uma instrução, independente da instrução.
8. Desvio condicional se classifica em tipo I
9. Analise com verdadeiro e falso
    - Instruções de acesso à memória não utilizam a ULA no caminho de dados, pois fazem apenas acesso à memória. - **FALSO**
    - A instrução beq utiliza a ULA para a operação de subtração. - **VERDADEIRO**
    - Instruções do tipo R sempre utilizam a ULA para alguma operação. - **VERDADEIRO**

10. O PC é incrementado em 4 unidades a cada ciclo de busca de instrução (fetch)
11. No modelo com suporte a pipeline, o recurso da bolha (nop) aumenta o desempenho do sistema, pois a utilização da bolha remove todos os riscos encontrados no fragmento de código. -> **VERDADEIRO**
12. No modelo uniciclo, o controle da ULA usa apenas as informações de Opcode (6 bits) para decidir qual será a operação matemática demandada à ULA. -> **FALSO**. Justificativa: Além do opcode, depende também do valor do funct para decidir a operação demandada
13. Um dos principais efeitos do pipeline é melhorar o desempenho aumentando a vazão das instruções, e não o tempo de execução de instruções individuais. -> **VERDADEIRO**
14. No modelo com suporte a pipeline, os efeitos adversos decorrentes da ocorrência de um risco de controle podem ser mitigados por meio de heurísticas de previsão de desvios. -> **FALSO**. Justificativa:
15. No modelo com suporte a pipeline, a unidade operativa lê uma nova instrução antes de terminar a execução de uma instrução que acabou de deixar o estágio de busca (instruction fetch). -> **VERDADEIRO**
16. No modelo com suporte a pipeline, a unidade operativa executa a instrução mais rapidamente, ou seja, diminui-se a latência de uma instrução. -> **FALSO**. Justificativa: Se os estágios forem balanceados, o tempo da instrução c/ pipeline ($Tempo\ da\ instrução\ sem\ pipeline\ \div\ numero\ de\ estagios$) tem que ser =

4, e a melhoria se deve a um maior vazão de instruções, não à diminuição do tempo de cada uma, ou seja, a latência não diminui
17. No modelo com suporte a pipeline, a unidade operativa executa um conjunto de instruções mais rapidamente sem diminuir a latência de uma instrução. -> **VERDADEIRO**
18. No modelo com suporte a pipeline, o risco de controle é uma situação em que uma instrução de desvio condicional que está entrando no estágio de fetch depende de uma informação/valor que vai ser gerado em consequência da execução de uma instrução anterior da listagem. -> **VERDADEIRO**.
19. No modelo com suporte a pipeline, o risco de dados é uma situação em que uma instrução que está entrando no estágio de fetch depende de uma informação/valor que vai ser gerado em consequência da execução de uma instrução posterior da listagem. -> **FALSO**. Justificativa:
20. No modelo com suporte a pipeline, o risco de estrutura é uma situação adversa que provoca a destruição da unidade operativa. -> **FALSO**. Justificativa: Os hazards estruturais ocorrem quando existe um conflito ao utilizar um recurso, onde ocorre um stall quando não é possível utilizar uma parte da memória no ciclo, mas não causa destruição da unidade operativa.
21. No modelo com suporte a pipeline, os riscos são situações adversas que não podem ser evitadas através da atuação do compilador ou de um programador experiente. -> **FALSO**. Justificativa: Errado, pois para cada tipo de Hazard existem maneiras diferentes de mitigar ou evitar o risco presente nos procedimentos e funções.
22. Numa arquitetura uniciclo, o tempo de execução de todas as instruções demora o mesmo tempo da mais lenta, que é a de acesso à memória, porque o tempo de clock do processador é o mesmo para qualquer instrução e deve abranger o tempo da mais demorada. **VERDADEIRO**.

Lista 2

1. Nível mais alto da hierarquia é a memória SRAM (cache), o intermediário é a memória DRAM e o mais baixo é o Disco Magnético
2. Acerca do acesso de dados pelo processador:
    - O processador sempre requisita um endereço virtual -> **VERDADEIRO**
    - Caso o endereço virtual não esteja listado na TLB, então isso significa que o dado não está carregado na memória -> **FALSO,** pois
    - A memória cache armazena os endereços virtuais dos dados p/ acelerar o acesso -> **FALSO,** pois
3. Analise as asserções a seguir:
    - As unidades de memória de um computador organizam-se em primária e secundária -> **VERDADEIRO**
    - A memória principal é composta pelas unidades não voláteis -> **FALSO**
    - O processador acessa um dado diretamente no nível mais baixo da hierarquia, pela solicitação de um endereço da memória secundária -> **FALSO**
4. Analise as asserções a seguir:
    - As caches tiram proveito da localidade temporal -> **VERDADEIRO**
    - Em uma leitura, o valor retornado depende de quais blocos estão na cache -> **FALSO**
    - A maioria do custo da hierarquia de memória está no nível mais alto -> **FALSO**
    - A maioria da capacidade da hierarquia de memória está no nível mais baixo -> **VDD**
5. Elementos que compõem um bloco na memória cache: tag, bit de validade e dados
6. Principais tecnologias de memória utilizadas: SRAM, DRAM, Memória Flash e Discos Magnéticos
7. V e F
    - Um processador utiliza uma pequena proporção de dados em um instante de tempo -> **V**
    - Como a memória SRAM é cara, ela não é utilizada em computadores pessoais, pois seu uso tornaria o custo inviável para o usuário final -> **FALSO**, pois existe sim em pcs pessoais, porém em pouca quantidade
    - A tecnologia mais rápida que pode ser utilizada em computadores é a tecnologia DRAM -> **FALSO**, pois é a SRAM
8. Suponha que você tenha uma memória cache diretamente mapeada com 16 blocos, e uma memória RAM com capacidade de armazenamento de 1024 blocos. Qual seria a tag e o endereço na cache do bloco 542 da memória principal

posição 542 = 1000011110 em binario, 16 blocos = 2^4 blocos, 1024 de memoria RAM = 2^10, tag e endereço?

no mapeamento associativo temos que: tag tem t-n bits e endereço tem n bits
dado que n = 4 bits pois 16 = 2^4
dado que t = 10 bits pois 1024 = 2^10
temos que o endereço sao os ultimos 4 bits de 542
temos que a tag sao os primeiros 6 bits de 542
tag = 100001 e endereço = 1110
9.
512 blocos com 4 bytes cada, memória principal de 4GiB, tag?
cache tem 512 blocos = 2^9 blocos, n = 9
4 bytes cada bloco = 2^2, b = 2
memória principal 4GiB = 2^2 * 2^30 bytes = 2^32, t = 32
pelo mapeamento direto temos que: tag = t-n-b = 32-9-2 = 21
10. A escolha do tamanho do bloco numa memória cache é uma decisão de projeto. Neste contexto, analise as asserções a seguir:

- De acordo com o princípio de localidade espacial, quanto maior o tamanho do bloco, menor a taxa de falhas -> **VERDADEIRO**
- Em sistemas de concorrência de processos, quanto maior o tamanho do bloco, maior a taxa de falhas -> **VERDADEIRO** (não sei dizer pq)
- Quanto maior o tamanho do bloco, menor a penalidade de falha -> **FALSO**, pelo contrário, a penalidade de falha aumenta com o tamanho do bloco, podendo aumentar também o número de falhas por conflito e as falhas de capacidade, porém, reduz a taxa de falhas

11. Considere um computador com uma memória principal de 16 GiB e uma cache diretamente mapeada que seja capaz de armazenar 256 KiB de dados, em blocos de 8 palavras (ou seja, 32 B). Qual o tamanho real, em KiB, dessa memória cache?
memória principal = 16GiB = 2^4 * 2^30 = 2^34 bits, t = 34
bloco com 8 palavras = 8*4 = 32 = 2^5 bits, b = 5
cache com 256 KiB = 256/32 bits = (2^8 * 2^10) / 2^5 = 2^13, n = 13
tag = 34-13-5 = 16 bits
tam real = 2^13 * [1+16+2^(5+3)] = 2^13 * 273 bits
(2^13 * 273) / (2^3*2^10) = 273 KiB
12. Instruções em um laço é um exemplo que ilustra apenas a localidade temporal, mas não a espacial -> **FALSO**, um laço de instruções usa locais de memória usados recentemente, ilustrando a localidade temporal, porém também pode utilizar a localidade espacial, pois diz respeito à tendência de um processador acessar instruções sequencialmente
13. O objetivo na construção de uma hierarquia de memória é a minimização da taxa de falhas -> **VERDADEIRO**
14. SRAM (Static Random Access Memory) é uma tecnologia de memória cujos dados precisam ser periodicamente atualizados, pois tal tecnologia não consegue manter os dados por muito tempo -> **FALSO**, nenhuma atualização é necessária pois não são usados capacitores, e