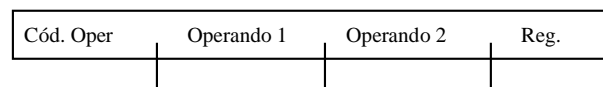


Curso de Tecnologia em Sistemas de Computação
Disciplina: Organização de Computadores
GABARITO - AP1 1º semestre de 2018 - Rocinha.

1. (3,5) Suponha que você deve projetar uma máquina com as seguintes especificações:

- Capaz de endereçar 128 M células de memória principal, sendo que cada célula armazena 4 bytes.
- Deve possuir os registradores RDM (utilizado para enviar e receber dados para/de o barramento de dados), REM (utilizado para enviar endereços no barramento de endereços), CI (utilizado para indicar o endereço da instrução a ser lida da memória) e RI (utilizado para armazenar uma instrução).
- Cada instrução deve conter um código de operação, dois operandos e um registrador como mostrado abaixo:



onde o Operando 1 e o Operando 2 são endereços da memória principal e Reg. é o identificador de um Registrador, sendo que a máquina possui 14 registradores.

- Deve poder ter um máximo de 58 códigos de operação diferentes.
- a) (0,5) Indique qual deve ser o tamanho mínimo em bits do REM
REM = Barramento de endereços, este terá a capacidade de endereçar 128Mcélulas = N
 $N = 128\text{Mcélulas} \Rightarrow N = 2^{27} \Rightarrow e = 27 \text{ bits}$
REM = barramento de endereços = 27 bits
- b) (0,4) Indique qual deve ser o tamanho mínimo em bits do barramento de endereços.
REM = barramento de endereços = 27 bits
- c) (0,6) Calcule o número de células que uma instrução necessita para ser armazenada.
Cada instrução = código de operação + 1o operando + 2o operando + 1 registrador
1o e 2o operando = endereço de uma célula = 27 bits
registrador = para atender pelo menos 14 regs. são necessários 4 bits
cod.operação = tamanho necessário para 58 códigos diferentes = necessários 6 bits
tamanho da instrução = 6 + 27 + 27 + 4 = 64 bits (2 células)
- d) (0,6) Indique o tamanho do RDM e do barramento de dados.
RDM = barramento de dados = tamanho necessário para transferir uma instrução
RDM = barramento de dados = 64 bits (2 células).
- e) (0,6) Quantos acessos a memória são necessários para que Unidade Central de Processamento obtenha uma instrução da memória, dado que cada acesso é lida uma célula da memória principal.
Serão necessários 2 acessos, já que uma instrução ocupa 2 células
- f) (0,8) Calcule a capacidade de armazenamento em bits dos registradores RI e CI, utilizando-se os valores calculados nos itens anteriores.
CI terá o tamanho mínimo necessário para endereçar toda a memória = REM = 27 bits
RI terá o tamanho mínimo para receber uma instrução = 64 bits

2. (2,5) Considere uma máquina que pode ter seu ciclo de busca e execução de uma instrução dividido em 5 estágios totalmente independentes: Busca de Instrução (BI), Decodificação (DI), Cálculo de Endereços de Operandos (CO), Execução (EX) e Escrita de Operandos (EO). Cada um dos estágios BI, EX e EO possui a duração de 7 ns e cada estágio DI e CO tem duração de 4 ns. Cada instrução desta máquina precisa executar os 5 estágios que serão sempre executados na sequência BI, DI, CO, EX e EO.

- a) (0,2) Uma implementação desta máquina foi realizada de modo que cada instrução deve ser completamente realizada em um único ciclo de relógio. Calcule a duração do ciclo de relógio que esta implementação deve possuir. Lembre-se que todas as instruções necessitam dos 5 estágios.

<i>BI</i> 7ns	<i>DI</i> 4ns	<i>CO</i> 4ns	<i>EX</i> 7ns	<i>EO</i> 7ns
------------------	------------------	------------------	------------------	------------------

O ciclo de relógio deverá ter o tempo de 29ns

- b) (0,5) Como cada estágio é independente um do outro, deseja-se implementar uma nova arquitetura utilizando-se um pipeline de 5 estágios. Nesta nova implementação cada estágio do pipeline deve ser executado em um ciclo de relógio. Calcule a duração do ciclo de relógio que esta implementação pipeline deve possuir.

1°. Estágio	2°. Estágio	3°. Estágio	4°. Estágio	5°. Estágio
<i>BI</i> 7ns	<i>DI</i> 7ns	<i>CO</i> 7ns	<i>EX</i> 7ns	<i>EO</i> 7ns

Ciclo de relógio será igual ao tempo para execução do estágio com maior tempo de execução = 7ns

- c) (0,5) Considere um programa que necessita executar 1.000 instruções. Calcule o tempo de execução deste programa na máquina do item a e na máquina do item b.

Seja T_{ex} = tempo de execução de uma instrução = número de estágios \times ciclo de relógio (determinado nos itens anteriores)

Para o item a (sem pipeline) :

$$T_{ex} = 1 \text{ estágio de } 29ns = 29ns$$

$$T_{total} = 1000 \text{ instruções} \times T_{ex} = \mathbf{29.000 \text{ ns}}$$

Para o item b (pipeline: 5 estágios) :

$$T_{ex} = 5 \text{ estágios de } 7ns \text{ cada} = 35ns$$

$$T_{total} = T_{ex} + 999 \times \text{tempo de 1 estágio}$$

$$T_{total} = 35ns + 999 \times 7ns = \mathbf{7.028 \text{ ns}}$$

- d) (0,5) Considere um programa que necessita executar 100.000 instruções. Calcule o tempo de execução deste programa na máquina do item a e na máquina do item b.

Seja T_{ex} = tempo de execução de uma instrução = número de estágios \times ciclo de relógio (determinado nos itens anteriores)

Para o item a (sem pipeline) :

$$Tex = 1 \text{ estágio de } 29ns = 29ns$$

$$Ttotal = 100.000 \text{ instruções} \times Tex = \underline{2.900.000 \text{ ns}}$$

Para o item b (pipeline: 5 estágios) :

$$Tex = 5 \text{ estágios de } 7ns \text{ cada} = 35ns$$

$$Ttotal = Tex + 99.999 \times \text{tempo de 1 estágio}$$

$$Ttotal = 35ns + 99.999 \times 7ns = \underline{700.028 \text{ ns}}$$

- e) (0,8) Compare as diferenças dos tempos de execução obtidos pela máquina do item a e a do item b para os programas dos itens c e d.

O uso do pipeline permite que mais de uma instrução seja executada ao mesmo tempo reduzindo, assim, o tempo total na execução dos programas. Esta afirmação pode observada nos tempos de execução calculados nos itens c e d para a máquina do item b (com pipeline) que é bem inferior quando comparada com a máquina do item a (sem pipeline).

3. (2,0) Considere uma máquina que possa endereçar 1 Giga bytes de memória física, sendo que cada endereço referencia uma célula de 8 bytes. Ela possui uma memória cache que pode armazenar 1 Mega blocos, sendo um bloco por linha e cada bloco possui 1 célula. Mostre o formato da memória cache, indicando os campos necessários (tag, bloco) e o número de bits para cada campo, o formato de um endereço da memória principal, indicando os bits que referenciam os campos da cache, e a capacidade em bits que a memória cache deve possuir (pode deixar a conta indicada) para os seguintes mapeamentos:

- a) Mapeamento direto.

Memória Principal

=> Tamanho da memória (em bytes) = 1GBytes, cada 1 célula armazena 8 bytes, temos $N = 128M$ células

=> Será organizada em blocos de 1 célula (cada com 8bytes), $K = 1$

=> $N = 128M$ células e $K = 1$ células por bloco, o total de blocos da MP (B) será:

Total de blocos: $B = N / K \Rightarrow B = 128M \text{ células} / 1 \text{ células por bloco} \Rightarrow B = 128M \text{ blocos}$

Memória Cache

=> O K (quantidade de células/bloco) tem de ser igual a MP.

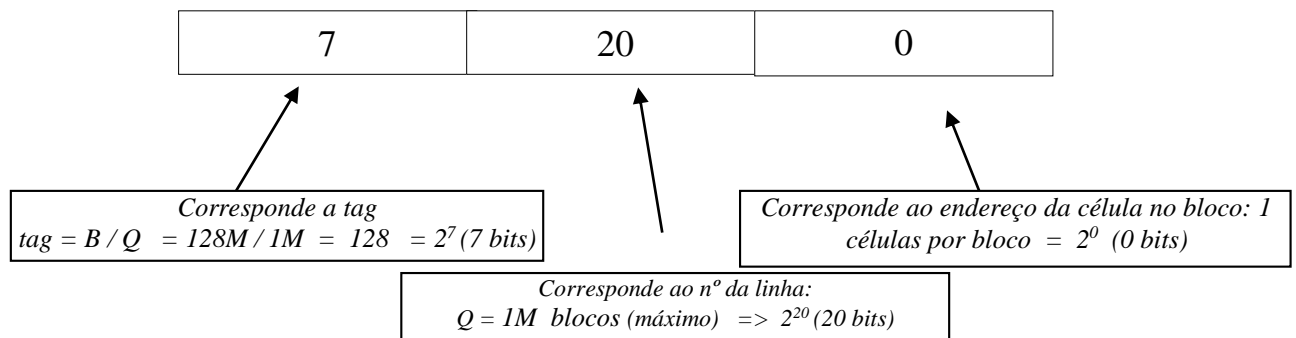
=> Tamanho da memória cache (em blocos ou linhas) => $Q = 1M$ blocos ou linhas

=> Tamanho da memória cachê em células = $Q \times K = 1M \text{ blocos} \times 1 \text{ células/linha} = 8M\text{Bytes}$

Memória principal

=> Para endereçarmos toda a MP precisamos da seguinte quantidade de bits (E)

sendo $N = 2^E \Rightarrow N = 128M \text{ células} \Rightarrow N = 2^{27} \Rightarrow E = 27 \text{ bits}$



b) Mapeamento totalmente associativo

Memória Principal

=> Tamanho da memória (em bytes) = 1GBytes, cada 1 célula armazena 8 bytes, temos $N = 128M$ células

=> Será organizada em blocos de 1 célula (cada com 8bytes), $K = 1$

=> $N = 128M$ células e $K = 1$ células por bloco, o total de blocos da MP (B) será:

Total de blocos: $B = N / K \Rightarrow B = 128M \text{ células} / 1 \text{ células por bloco} \Rightarrow B = 128M \text{ blocos}$

Memória Cache

=> O K (quantidade de células/bloco) tem de ser igual a MP.

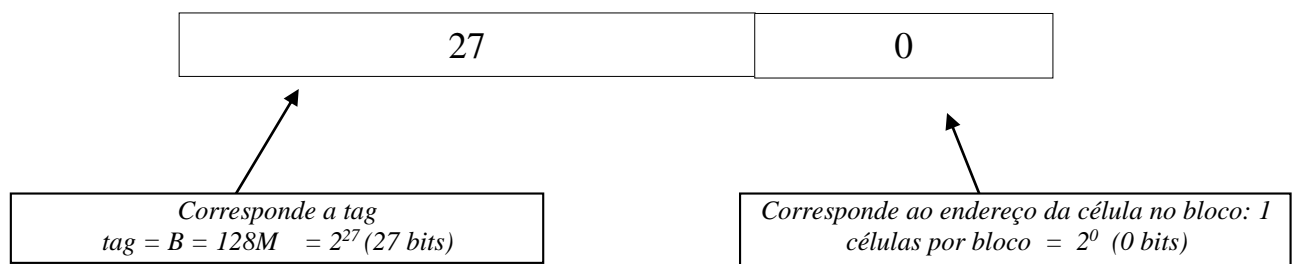
=> Tamanho da memória cache (em blocos ou linhas) => $Q = 1M \text{ blocos ou linhas}$

=> Tamanho da memória cache em células = $Q \times K = 1M \text{ blocos} \times 1 \text{ células/linha} = 8M\text{Bytes}$

Memória principal

=> Para endereçarmos toda a MP precisamos da seguinte quantidade de bits (E)

sendo $N = 2^E \Rightarrow N = 128M \text{ células} \Rightarrow N = 2^{27} \Rightarrow E = 27 \text{ bits}$



4. (2,0) Explique em detalhes a hierarquia de memória dos computadores atuais.

Podemos ilustrar essa hierarquia de memória na forma de uma pirâmide dividida em 4 níveis. No topo da pirâmide teríamos os registradores, que são pequenas unidades de memória que armazenam dados na UCP. São dispositivos de maior velocidade com tempo de acesso em torno de 1 ciclo de memória, menor capacidade de armazenamento além de armazenar as informações por muito pouco tempo.

Em um nível abaixo teríamos a memória cache, cuja função é acelerar a velocidade de transferência das informações entre UCP e MP e, com isso, aumentar o desempenho do sistema. A UCP procura informações primeiro na Cache. Caso não as encontre, as mesmas são transferidas da MP para a Cache. A cache possui tempo de acesso menor que a da Memória principal, porém com capacidade inferior a esta, mas superior ao dos registradores e o suficiente para armazenar uma apreciável quantidade de informações, sendo o tempo de permanência do dado menor do que o tempo de duração do programa a que pertence.

Abaixo da memória cache teríamos a memória básica de um sistema de computação, que é a memória principal. Dispositivo onde o programa (e seus dados) que vai ser executado é armazenado para que a UCP busque instrução por instrução para executá-las. A MP são mais lentas que a cache e mais rápidas que a memória secundária, possui capacidade bem superior ao da cache e os dados ou instruções permanecem na MP enquanto durar a execução do programa.

Finalmente, na base da pirâmide teríamos a memória secundária, memória auxiliar ou memória de massa, que fornece garantia de armazenamento mais permanente aos dados e programas do usuário. Alguns dispositivos são diretamente ligados: disco rígido, outros são conectados quando necessário: disquetes, fitas de armazenamento, CD-ROM. São os mais lentos em comparação com os outros níveis de memória, mas possuem a maior capacidade de armazenamento e armazenam os dados de forma permanente.