

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. (2,0) Um computador possui uma capacidade máxima de memória principal de 128 K células, cada uma capaz de armazenar uma palavra de 8 bits.

- a) Qual é o maior endereço em decimal desta memória?
- b) Qual é o tamanho do barramento de endereços deste sistema?
- c) Quantos bits podem ser armazenados no RDM e no REM?
- d) Qual é o número máximo de bits que pode existir na memória?

- a) Qual é o maior endereço em decimal desta memória ?

Sabendo que a capacidade máxima de memória = 128K células

Teremos os endereços variando de 0 até $2^{17} - 1$, ou seja, de 0 até 131071

O maior endereço é portanto 131071

- b) Qual é o tamanho do barramento de endereços deste sistema ?

Como a máquina endereça 128K células = 2^{17} células, são necessários 17 bits para endereçá-las. Logo o barramento de endereços tem que ter tamanho igual a 17 bits.

- c) Quantos bits podem ser armazenados no RDM e no REM ?

RDM tem 8 bits, tamanho da palavra da memória principal;

REM tem a mesma quantidade de bits do barramento de endereços, 17 bits.

- d) Qual é o número máximo de bits que pode existir na memória ?

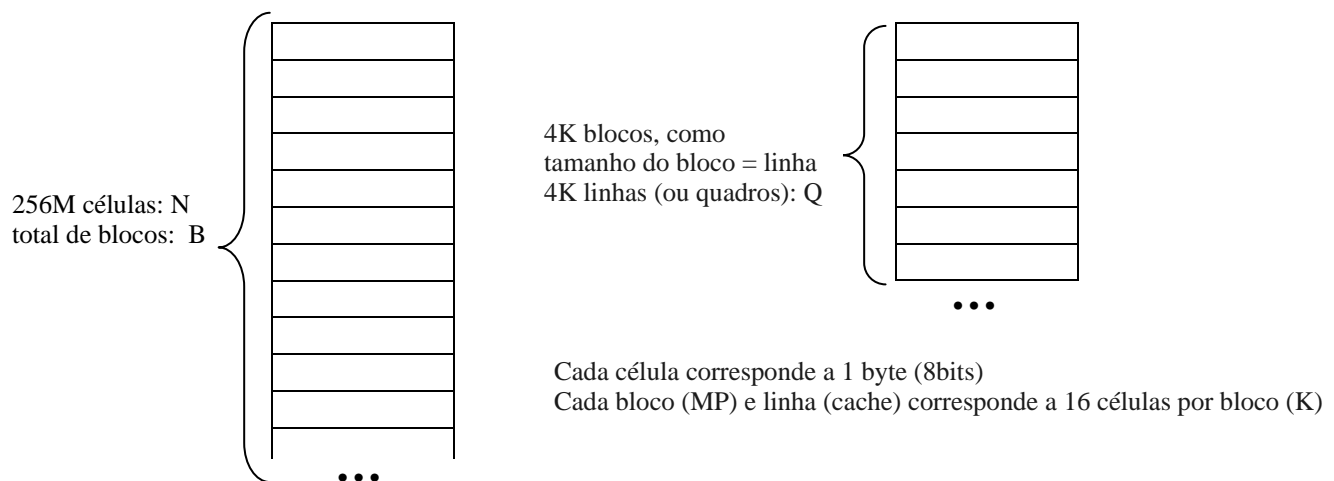
A capacidade da memória em bits é igual ao número de células multiplicado pelo tamanho da célula em bits, ou seja, 128K células \times 8 bits = 1024 K bits (ou 1 M bits)

2. (2,0) Considere uma máquina que possa endereçar 256 Mbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 16 bytes. Ela possui uma memória cache que pode armazenar 4K blocos, sendo um bloco por linha (ou quadro). Mostre o formato da memória cache, indicando os campos necessários (tag, bloco) e o número de bits para cada campo, e o formato de um endereço da memória principal, indicando os bits que referenciam os campos da cache, para os seguintes mapeamentos:

- Mapeamento direto.
- Mapeamento totalmente associativo.
- Mapeamento associativo por conjunto, onde cada conjunto possui duas linhas, cada uma de um bloco.

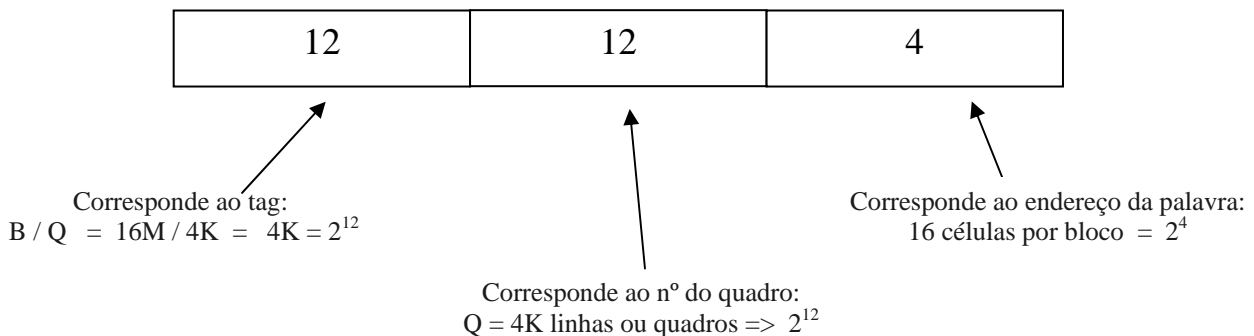
a) Mapeamento direto.

Tamanho da memória (em bytes) = 256Mbytes, como 1 célula = 1 byte, temos $N = 256M$ células



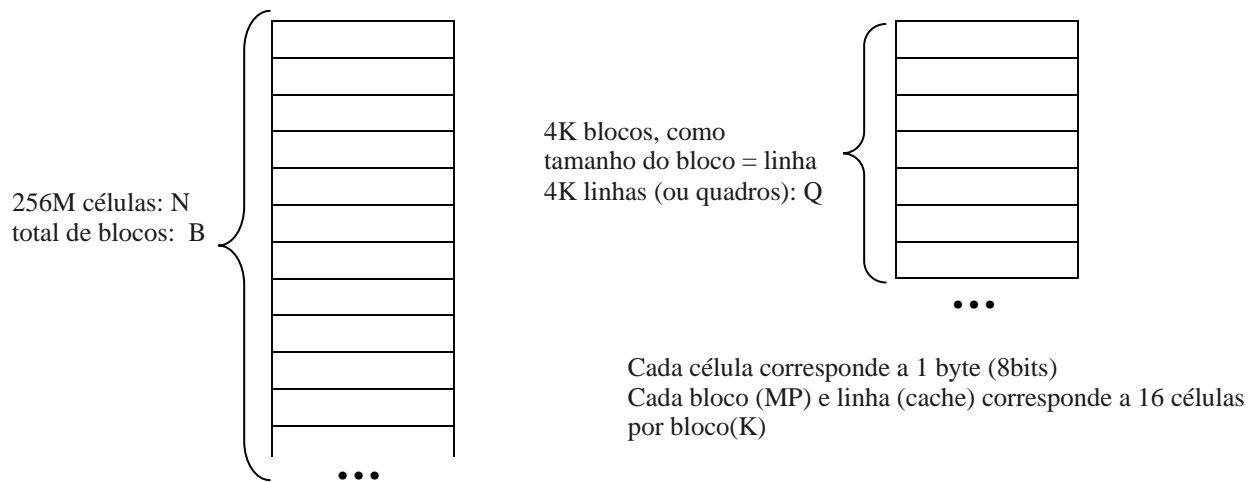
Total de blocos: $B = N / K \Rightarrow B = 256M \text{ células} / 16 \text{ células/bloco} \Rightarrow B = 16M \text{ blocos}$

Endereço da memória principal corresponde a E
sendo $N = 2^E \Rightarrow N = 256M \text{ células} \Rightarrow N = 2^{28} \Rightarrow E = 28 \text{ bits}$



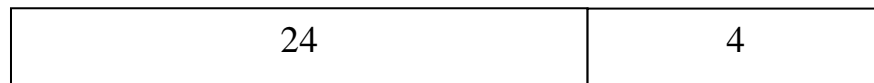
b) Mapeamento totalmente associativo.

Tamanho da memória (em bytes) = 256Mbytes, como 1 célula = 1 byte, temos $N = 256M$ células



Total de blocos: $B = N / K \Rightarrow B = 256M \text{ células} / 16 \text{ células/bloco} \Rightarrow B = 16M \text{ blocos}$

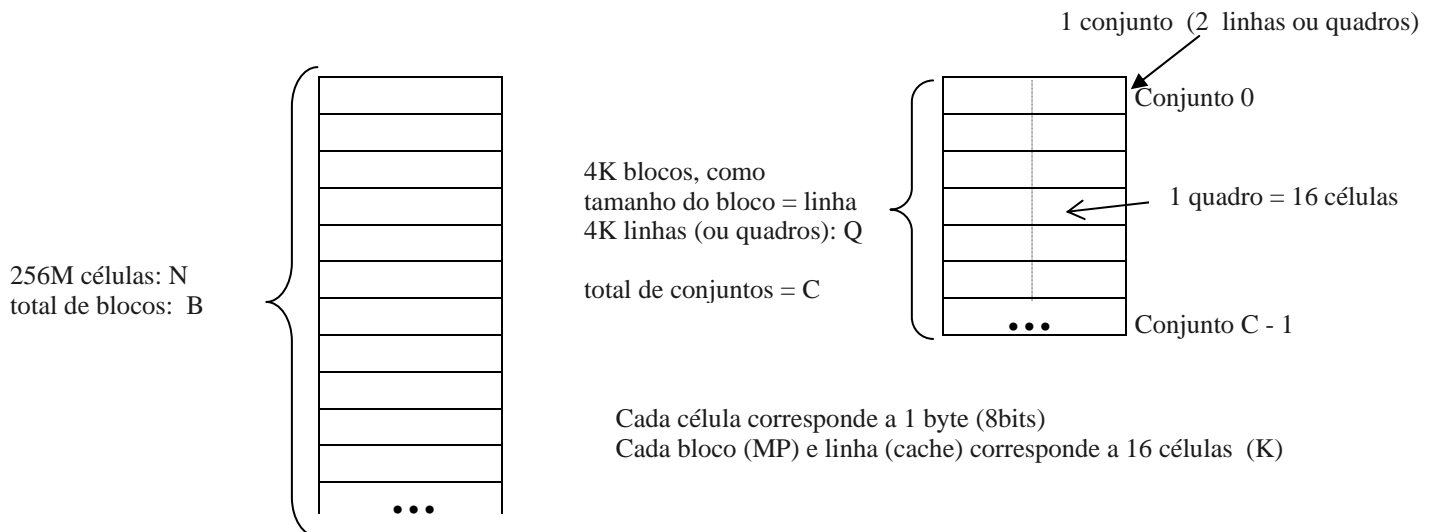
Endereço da memória principal corresponde a E
sendo $N = 2^E \Rightarrow N = 256M \text{ células} \Rightarrow N = 2^{28} \Rightarrow E = 28 \text{ bits}$



Corresponde ao tag = $B = 16M = 2^{24}$

Corresponde ao endereço da palavra:
16 células por bloco = 2^4

- c) Mapeamento associativo por conjunto, onde cada conjunto possui duas linhas, cada uma de um bloco.



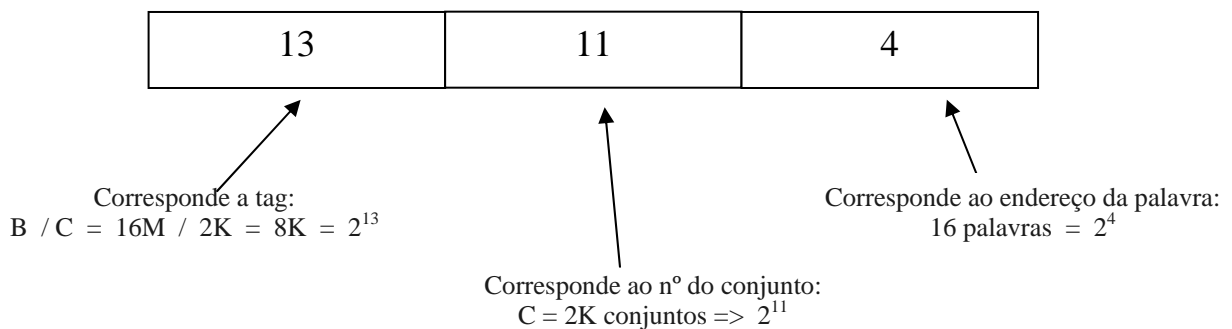
Na memória cache, a quantidade de linhas (ou quadros): $Q = C \times D$ sendo,
 Q a quantidade de linhas (ou quadros)
 C o total de conjuntos
 D a quantidade de linhas (ou quadros) por cada conjunto.

Total de blocos na MP: $B = N / K \Rightarrow B = 256M \text{ células} / 16 \text{ células/bloco} \Rightarrow B = 16 \text{ M blocos}$

Para sabermos a quantidade de conjuntos

$Q = C \times D \Rightarrow 4 \text{ K} = C \times 2 \text{ linhas/conjunto} \Rightarrow C = 2 \text{ K conjuntos}$

Endereço da memória principal corresponde a E
 sendo $N = 2^E \Rightarrow N = 256M \text{ células} \Rightarrow N = 2^{28} \Rightarrow E = 28 \text{ bits}$



3. (2,0) Suponha que você deve projetar uma máquina com as seguintes especificações:

- Capaz de endereçar 128M células de memória principal, sendo que cada célula armazena 1 byte.
- 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 e um operando como mostrado abaixo:

| | |
|-----------|----------|
| Cód. Oper | Operando |
|-----------|----------|

onde Operando é um endereço da memória principal.

- (0,6) Indique qual deve ser o tamanho mínimo em bits do REM e do barramento de endereços.
 - (0,4) Supondo que uma instrução ocupe 4 células de memória, indique o número de códigos de operação diferentes que podem existir no conjunto de instruções desta máquina.
 - (0,4) Indique o tamanho do RDM e o barramento de dados de modo que a Unidade Central de Processamento obtenha uma instrução da memória principal realizando somente um acesso à memória principal.
 - (0,6) Calcule a capacidade de armazenamento em bits dos registradores RI e CI, utilizando-se os valores calculados nos itens anteriores.
-
- (0,6) Indique qual deve ser o tamanho mínimo em bits do REM e do barramento de endereços.

O barramento de endereços deverá endereçar 128M células $= N$
Barramento de endereços = E, sendo $N = 128M \Rightarrow N = 2^{27} \Rightarrow E = 27$
Barramento de endereços = 27 bits = REM

- (0,4) Supondo que uma instrução ocupe 4 células de memória, indique o número de códigos de operação diferentes que podem existir no conjunto de instruções desta máquina.

Tamanho da instrução = 4 células, sendo 1 célula = 1 byte = 4 células x 8bits/célula = 32bits

Tamanho da instrução = tamanho do código de operação + tamanho do operando
Como 1 operando equivale ao tamanho para endereçamento da memória = 27bits

Temos,
32 bits = tamanho do código de operação + 27 bits

tamanho do código de operação = 5 bits

Quantidade de operações = $2^5 = 32$ instruções diferentes

- c) (0,4) Indique o tamanho do RDM e o barramento de dados de modo que a Unidade Central de Processamento obtenha uma instrução da memória principal realizando somente um acesso à memória principal.

Tamanho do RDM = barramento de dados = tamanho inteiro de uma instrução para que a mesma seja transferida de uma só vez em um só acesso

$$\text{RDM} = \text{barramento de dados} = \text{tamanho da instrução} = 32 \text{ bits}$$

- d) (0,6) Calcule a capacidade de armazenamento em bits dos registradores RI e CI, utilizando-se os valores calculados nos itens anteriores.

$$\text{RI} = \text{tamanho de uma instrução, portanto } \text{RI} = 32 \text{ bits}$$

$$\begin{aligned} \text{CI} &= \text{endereço de memória} = \text{tamanho do barramento de endereços;} \\ \text{CI} &= 27 \text{ bits} \end{aligned}$$

4. (2,0) Considere uma máquina que pode ter seu ciclo de busca e execução de uma instrução dividido em 5 estágios totalmente independentes: Busca (B), Decodificação/Leitura de registradores (D), Operação da ULA (U), Acesso à memória de dados (M) e Escrita nos registradores (E). Cada um dos estágios B e M possui a duração de 2 ns e cada estágio D, U e E tem duração de 1 ns. Cada instrução desta máquina precisa executar os 5 estágios.

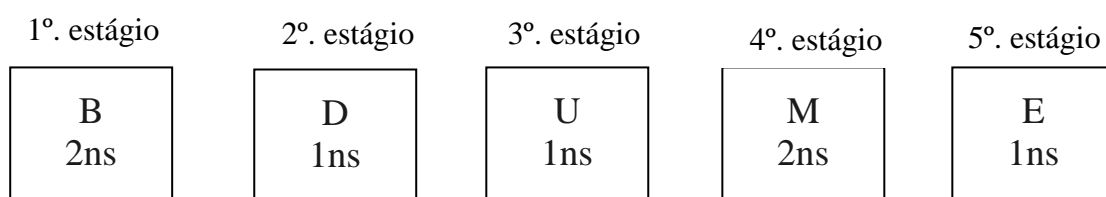
- a) (0,5) Uma implementação desta máquina foi realizada de modo que cada instrução deve ser completamente realizada em um único ciclo de relógio e uma instrução só começa a ser realizada após o término da anterior. Calcule a duração do ciclo de relógio que esta implementação deve possuir. Lembre-se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

| | | | | |
|----------|----------|----------|----------|----------|
| B 2ns | D 1ns | U 1ns | M 2ns | E 1ns |
|----------|----------|----------|----------|----------|



$$\begin{aligned} \text{Ciclo de relógio para execução de uma instrução (sem pipeline)} &= 2\text{ns} + 1\text{ns} + \\ &1\text{ns} + 2\text{ns} + 1\text{ns} = 7\text{ns} \end{aligned}$$

- b) (0,5) Como cada estágio é independente um do outro, implementou-se uma **nova** arquitetura utilizando-se um pipeline de 5 estágios. Calcule a duração do ciclo de relógio que a implementação pipeline deve ter. Considere que qualquer estágio do pipeline deve poder ser realizado em um único ciclo de relógio.



Ciclo de relógio será igual ao tempo para execução do estágio de maior tempo de execução = 2ns.

- c) (1,0) Mostre o tempo em que um programa que contenha 10 instruções será executado pela **implementação do item a e do item b.**

Seja T_{ex} = tempo de execução de uma instrução

= número de estágios x ciclo de relógio (determinado nos itens anteriores)

Item a (sem pipeline) :

$$T_{ex} = 1 \text{ estágio} \times 7ns = 7ns$$

$$T_{total} = 10 \text{ instruções} \times T_{ex} = \underline{70ns}$$

Item b (pipeline: 5 estágios)

$$T_{ex} = 5 \text{ estágios} \times 2ns = 10ns$$

$$T_{total} = T_{ex} + 9 \times (\text{tempo de 1 estágio})$$

$$T_{total} = 10ns + 9 \times 2ns = \underline{28ns}$$

5. (2,0) Explique como funcionam barramentos síncronos e assíncronos.

Nos barramentos que possuem operação síncrona, a ocorrência e duração de todos os eventos que ocorrem nas diversas linhas do barramento são guiados por pulsos de um relógio. Existe uma linha no barramento por onde circulam os pulsos gerados pelo relógio e todos os acontecimentos nas linhas de barramento, como, por exemplo, envio de endereço e envio de sinal de leitura, tem sua inicialização e duração de ocorrência determinadas por estes pulsos. Por exemplo, em uma operação de leitura realizada em um barramento síncrono, os passos serão tomados por base em intervalos de tempo iguais a T , baseados nos ciclos do relógio. O processo de leitura inicia no intervalo T_1 , onde a linha de início é ativada indicando que o endereço foi colocado no barramento de endereços; no próximo intervalo de tempo T_2 , os dados serão colocados no barramento de dados pelo dispositivo solicitado e no intervalo de T_3 é ativada a linha de confirmação.

Nos barramentos que operam de forma assíncrona, não existe um relógio sincronizador. Os eventos ocorrem no barramento de acordo com um protocolo de aperto de mão (handshaking). Cada evento no barramento não depende dos pulsos do relógio, mas sim de algum evento que deve ocorrer anteriormente a ele e que pode ter qualquer duração de tempo. Podemos exemplificar da seguinte forma. Quando uma CPU deseja realizar uma operação de leitura, a unidade de controle coloca o endereço da célula de memória (ou ativação do dispositivo) no barramento e ativa o sinal de leitura no barramento de controle (READ). Logo após estes estarem disponíveis no barramento é ativado o sinal de execução da tarefa (MSYN – ativação do “mestre”), e assim que o dispositivo “escravo” detectar este sinal, ele inicia de forma imediata a operação, com base no endereço, colocando os dados no barramento de dados. Concluindo esta etapa, o dispositivo “escravo” informa que os dados estão disponíveis ativando a linha SSYN. Logo após, o barramento fica disponível para início de outra operação da CPU.