

1. (2,0) Suponha que você deve projetar uma máquina com as seguintes especificações:

- Capaz de endereçar 16 M 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, um operando e um registrador como mostrado abaixo:

Cód. Oper	Operando	Reg.
-----------	----------	------

onde Operando é um endereço da memória principal e Reg. é o identificador de um Registrador, sendo que a máquina possui 8 registradores.

- Deve poder ter um máximo de 32 códigos de operação diferentes.

a) (0,3) Indique qual deve ser o tamanho mínimo em bits do REM

*Memória com 16Mcélulas =>  $N = 16M$  células  
tamanho mínimo do REM será o tamanho do barramento de endereços necessário  
para endereçar toda a memória.  
 $\text{Barramento de endereços (BE)} = \log_2 N = \log_2 16M = 24 \text{ bits}$   
 $\text{REM} = \text{tamanho do BE} = 24 \text{ bits}$*

b) (0,3) Indique qual deve ser o tamanho mínimo em bits do barramento de endereços.

*tamanho do BE = REM = 24 bits*

c) (0,6) Calcule o número de células que uma instrução necessita para ser armazenada.

*Cada instrução = código de operação + 2 operandos  
1o. operando = endereço de uma célula = 24 bits  
2o. operando = endereço de um registrador = 3 bits (total de  $2^3 = 8$  registradores)  
cod.operação = tamanho necessário para 32 códigos diferentes = 5 bits  
tamanho da instrução =  $5 + 24 + 3 = 32 \text{ bits}$   
1 célula = 1 byte  $\rightarrow$  quantidade de células =  $32 \text{ bits} / 8 \text{ bits} = 4 \text{ células}$*

- d) (0,6) Indique o **tamanho do RDM e do barramento de dados** de modo que a Unidade Central de Processamento obtenha uma instrução da memória principal realizando somente um acesso à memória principal.

*RDM = barramento de dados = tamanho necessário para transferir uma instrução*  
*RDM = barramento de dados = 32 bits.*

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

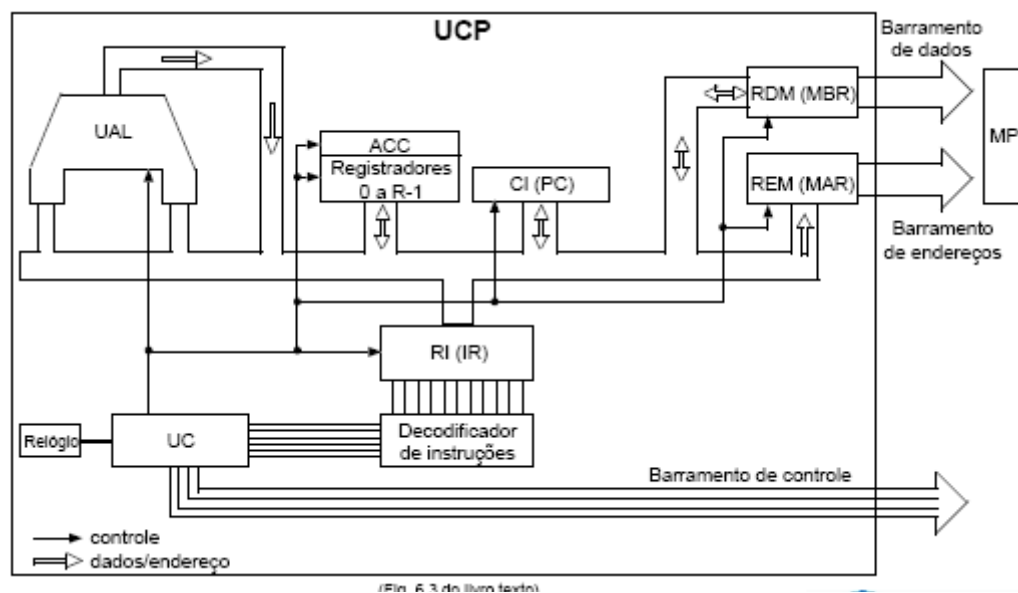
*CI = tamanho necessário para endereçar toda a memória = 24 bits*  
*RI = tamanho necessário para uma instrução = 32 bits*

2. (1,8) Explique como funcionam os barramentos síncronos e assíncronos.

*Nos barramentos que possuem operação síncrona, a ocorrência e duração de todos os eventos que acontecem 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.*

*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. .*

3. (1,8) Considere o sistema apresentado em aula mostrado na figura abaixo.



Descreva **detalhadamente** a execução do trecho de código abaixo, indicando como o Acumulador (ACC), RDM, REM, Unidade Aritmética Lógica (UAL) e Barramento de controle, de dados e de endereços são utilizados na execução de cada instrução.

- a) LDA 20  
b) STR 10

Lembre-se que (i) na execução da instrução LDA Op. o conteúdo da memória cujo endereço é Op. deve ser armazenado no acumulador e (ii) na execução de STR Op. o conteúdo do acumulador deve ser armazenado na memória no endereço Op.

a) LDA 20

- 1)  $RI \leftarrow (CI)$  , ou seja,  $RI \leftarrow$  recebe a Instrução contida no endereço contido no CI
- 2)  $CI \leftarrow CI + 1$
- 3) Decodificação do código de operação
  - recebe os bits do código de operação
  - produz sinais para a execução da operação de leitura em memória
- 4) Execução da operação
  - A UC emite sinais para que o valor do campo operando (20) seja transferido para a REM
  - A UC emite sinais para que o valor contido no REM seja transferido para o barramento de endereços
  - A UC ativa a linha READ do barramento de controle
  - Conteúdo da posição da memória, conforme endereço contido no barramento de endereços ( 20 ), é transferido através do barramento de dados para o RDM
  - O conteúdo do RDM é transferido para o registrador acumulador ( $ACC \leftarrow RDM$ )

b) STR 10

- a)  $RI \leftarrow (CI)$
- b)  $CI \leftarrow CI + 1$
- c) Decodificação do código de operação
  - recebe os bits do código de operação
  - produz sinais para a execução da operação de escrita
- d) Busca do operando na memória
  - A UC emite sinais para que o valor do campo operando = 10 seja transferido para a REM
  - Conteúdo do Acumulador (ACC) é transferido para RDM ( $RDM \leftarrow ACC$ )
  - A UC ativa a linha WRITE do barramento de controle
  - O REM passa o conteúdo para o barramento de endereços
  - O RDM passa o conteúdo para o barramento de dados
  - A memória grava o dado recebido no endereço que consta do barramento de endereços



b) Mapeamento totalmente associativo.

### Memória Principal

=>  $N = 256M$  células

=>  $K = 16$

=>  $B = 16M$  blocos

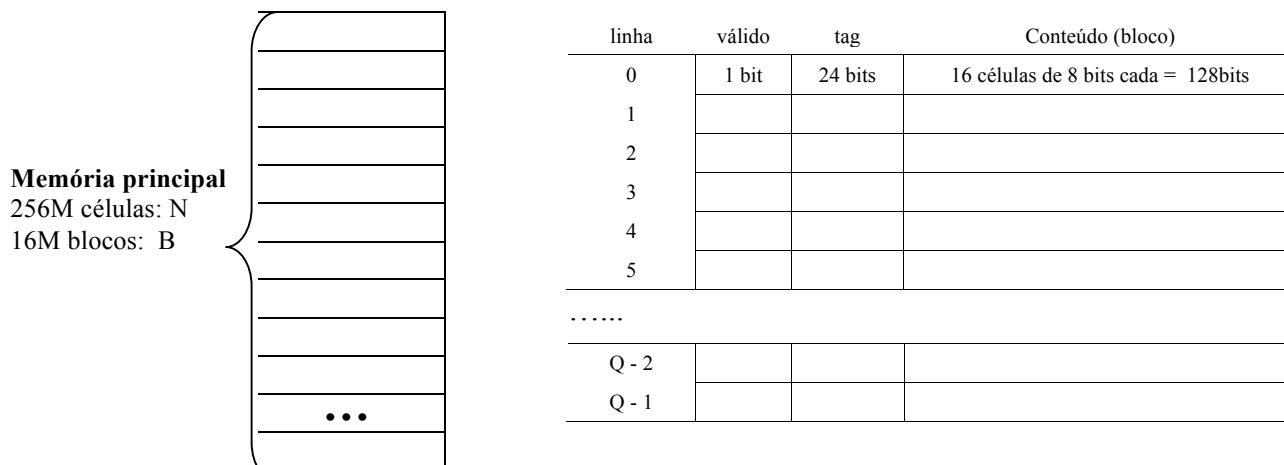
### Memória Cache

OBS: O  $K$  (quantidade de células/bloco) tem de ser igual a MP.

=>  $Q = 64K$  blocos

=> Tamanho da memória cache = 64Kbytes

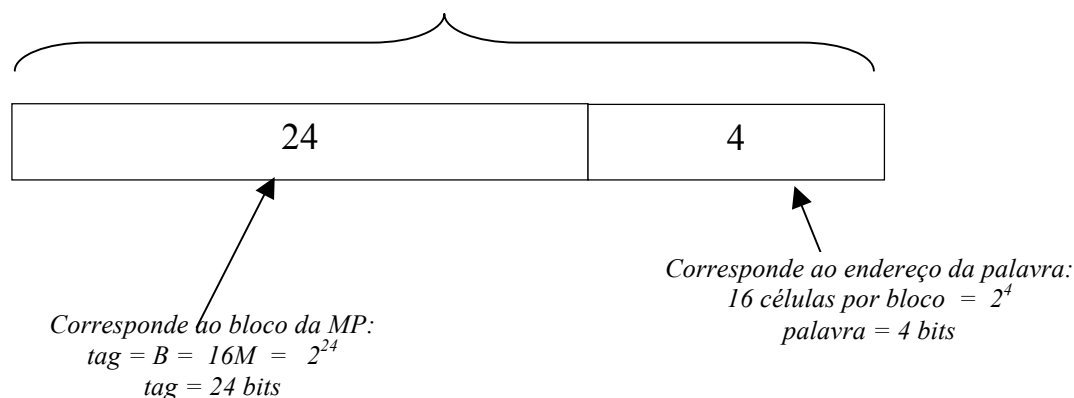
### Organização da cache



Para endereçarmos toda a MP precisamos da seguinte quantidade de bits:  $E = 28$  bits

Como o bloco pode ser alocado em qualquer posição da memória cache a tag indicará qual dos blocos da MP está alocado naquela posição da memória cachê

Tamanho do endereço da MP = 28 bits



5. (2,0) Explique os níveis que compõem 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.*