

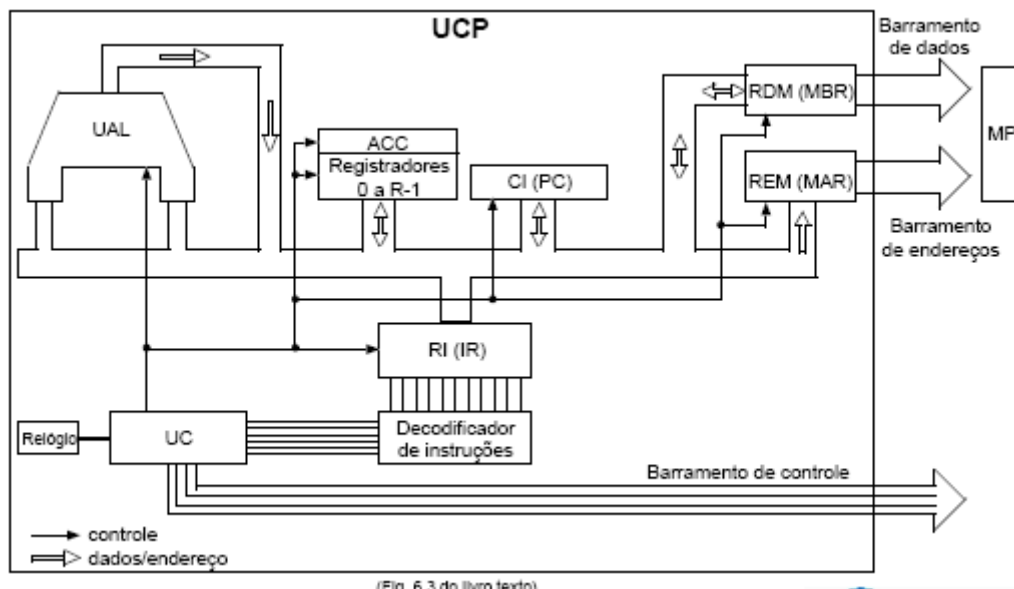
Nome –

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) Considere o sistema estudado em aula e mostrado na figura abaixo:



Descreva **detalhadamente** a execução das instruções **SUB Op.** e **JP Op.**, indicando como o Registrador de Instrução (RI), Contador de Instrução (CI), Acumulador (ACC), Registrador de Dados da Memória (RDM), Registrador de Endereços da Memória (REM), Unidade Aritmética Lógica (UAL) e Barramento de controle, de dados e de endereços são utilizados na execução destas instruções. Lembre-se que a instrução SUB Op., quando

executada, subtrai o conteúdo da memória cujo endereço é Op. do conteúdo do acumulador e a instrução JP Op., quando executada, carrega CI com o valor de Op. se o conteúdo do Acumulador é maior que zero, e caso contrário carrega CI com CI+1.

**Resposta:**

SUB Op

- a)  $RI \leftarrow (CI)$
- b)  $CI \leftarrow CI + 1$
- c) *Decodificação do código de operação*
- d) *Busca do operando na memória*
  - A UC emite sinais para que o valor do campo operando (Op) seja transferido para o REM
  - A UC ativa a linha READ do barramento de controle
  - Conteúdo de memória do endereço (Op) é transferido para o RDM
- e) *Execução da operação*
  - Dados para a operação de subtração são transferidos para a UAL:  
 $UAL(a) \leftarrow ACC$  e  $UAL(b) \leftarrow RDM$  (a) e (b) são as entradas da UAL
  - UAL executa a operação de subtração:  $UAL(a) - UAL(b)$ , ou seja,  $(ACC) - (RDM)$
  - ACC recebe o resultado:  $ACC \leftarrow ACC - RDM$

JP Op

- a)  $RI \leftarrow (CI)$
- b)  $CI \leftarrow CI + 1$
- c) *Decodificação do código de operação*
- d) *UC emite sinal para transferir conteúdo do acumulador para UAL*  
 $UAL \leftarrow (ACC)$
- e) *UAL executa operação de comparação*
  - e.1) Se Resultado = verdadeiro, isto é,  $(ACC) > 0$   
 $CI \leftarrow Operando \ (CI \leftarrow Op)$
- f) *Inicia o procedimento de leitura da instrução contida no endereço que consta em CI*

2. (2,0) 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 **X** ns e cada estágio DI e CO tem duração de **(X-2)** 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**. Na execução de um determinado programa P, verificou-se que foram executadas 1000 instruções e o programa foi executado em 16000 ns. Calcule **a duração do ciclo de relógio** que esta implementação possui.

**Resposta:**

<i>BI</i> <i>X ns</i>	<i>DI</i> <i>(X-2) ns</i>	<i>CO</i> <i>(X-2) ns</i>	<i>EX</i> <i>X ns</i>	<i>EO</i> <i>X ns</i>
--------------------------	------------------------------	------------------------------	--------------------------	--------------------------

$$| \text{-----} (5X - 4) \text{ ns} \text{-----} |$$

*Ciclo de relógio para execução de uma instrução (sem pipeline) =*  
 $X \text{ ns} + (X-2) \text{ ns} + (X-2) \text{ ns} + X \text{ ns} + X \text{ ns} = (5X - 4) \text{ ns}$

*Total de instruções = 1000, e tempo total para execução = 16.000ns,*

*Tempo para execução de 1 instrução = 16.000 / 1.000 = 16ns*

*Cada instrução é executada em um ciclo de relógio,*

*então  $(5X - 4) \text{ ns} = 16 \text{ ns} \Rightarrow X = 4 \text{ ns}$*

*Concluindo, os estágios BI, EX, EO terão duração, cada um de 4ns, e DI e CO de 2ns*

- b) (0,6) 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. Lembre-se que todas as instruções necessitam dos 5 estágios.

**Resposta:**

*O ciclo de relógio será igual ao tempo para execução do estágio com maior tempo de execução = 4 ns.*

- c) (0,6) Com o valor encontrado no item **b**, calcule o tempo de execução do programa P do item **a** na máquina com pipeline do item **b**.

**Resposta:**

*Seja  $T_{\text{ex}}$  = tempo de execução de uma instrução = número de estágios  $\times$  ciclo de relógio*

*Para o item b (pipeline: 5 estágios):*

$T_{\text{ex}} = 5 \text{ estágios} \times 4 \text{ ns} = 20 \text{ ns}$

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

$T_{\text{total}} = 20 \text{ ns} + 999 \times 4 \text{ ns} = \underline{\underline{4016 \text{ ns}}}$

- d) (0,6) Indique em qual das duas máquinas o programa P é executado mais rapidamente e calcule o ganho de desempenho obtido calculando a divisão do maior tempo de execução pelo menor tempo de execução.

**Resposta:**

*O programa P será executado mais rapidamente na máquina do item b (pipeline de 5 estágios).*

*Ganho de desempenho = 16.000 / 4.016 ns. O tempo da máquina sem pipeline (item a) será 3,98 vezes maior em relação àquela que tem pipeline (item b).*

3. (2,0) Explique **detalhadamente** como funciona uma Unidade Central de Processamento (UCP) com controle por microprograma, e explique os dois métodos utilizados para

formatar e usar uma microinstrução: microinstruções verticais e microinstruções horizontais.

**Resposta:**

*Em uma arquitetura microprogramada, a unidade de controle é especificada por um microprograma que consiste de uma seqüência de microinstruções de uma linguagem de microprogramação. Estas instruções são muito simples e especificam microoperações. Uma unidade de controle microprogramada é implementada com circuitos lógicos e é capaz de seguir uma seqüência de microinstruções gerando sinais de controle para que cada instrução de linguagem de máquina seja executada corretamente. Os sinais de controle gerados por uma microinstrução são usados, por exemplo, para causar transferências de dados entre registradores e memória e execução de operações pela UAL.*

*As microinstruções horizontais possuem um bit para gerar cada um dos sinais de controle, como por exemplo, sinal para controlar uma linha de controle interna da UCP, sinal para controlar uma linha de barramento externo de controle, sinal para definir condição de desvio e endereço de desvio.*

*Em microinstruções verticais, não utiliza-se um bit para cada sinal de controle, mas uma combinação de bits da instrução indica a ação que deve ser efetuada. Para isso, deve existir um decodificador extra que traduz este código em sinais de controle individuais que serão efetivamente ativados.*

4. (2,0) Um computador possui uma capacidade máxima de memória principal com 2 Giga células, cada uma capaz de armazenar uma palavra de 32 bits.

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

**Resposta:**

$$N = 2G\text{células} \Rightarrow N = 2^{31}$$
$$\text{Maior endereço} = N - 1 = 2^{31} - 1 = 2.147.483.647$$

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

**Resposta:**

*O tamanho deste barramento será o suficiente para endereçar todas as células da memória (N).  
O tamanho do barramento corresponderá ao valor de  $e$  em  $2^e = N$   
 $2^e = N \Rightarrow 2^e = 2^{31} \Rightarrow e = 31$ , portanto,  
**barramento de endereços = 31 bits***

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

**Resposta:**

*Tamanho do REM = tamanho do barramento de endereço. **REM = 31 bits**  
Tamanho do RDM = tamanho do barramento de dados e terá de ser no mínimo o tamanho de uma célula. **RDM = 32 bits***

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

**Resposta:**

*O total de bits da memória será igual a T*  
$$\text{Tamanho da memória (T)} = N \times M = 2G \text{ células} \times 32\text{bits} = 2^{31} \times 2^5\text{bits} = 2^{36} \text{ bits}$$
$$= 68.719.476.736 \text{ bits}$$

5. (2,0) Considere uma máquina que possa endereçar 2 Giga bytes de memória física, sendo que cada endereço referencia uma célula de 2 bytes. Ela possui uma memória cache que pode armazenar 1 Mega blocos, sendo um bloco por linha e cada bloco possui 2 células. 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**

$\Rightarrow$  Tamanho da memória (em bytes) = 2Gbytes, como 1 célula referencia a 2 bytes,

temos então,  $N = 1G$  células

$\Rightarrow$  Será organizada em blocos de 2 células, temos cada bloco = 2 células,  $K = 2$

$\Rightarrow$  Sendo  $N$  o tamanho endereçável da memória, e  $K$  que a quantidade de células por blocos,

$N = 1G$  células e  $K = 2$  células / blocos, o total de blocos da MP (  $B$  ) será:  
 $B = N / K \Rightarrow B = 1G \text{ células} / 2 \text{ células/bloco} \Rightarrow B = 512 \text{ M blocos}$

**Memória Cache**

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

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

$\Rightarrow$  Tamanho da memória cache em células =  $Q \times K =$

$1M \text{ blocos} \times 2 \text{ células/bloco} = 2M \text{ células}$

Cada célula possui 2 bytes, então, cache possui  $2M \text{ células} \times 2 \text{ bytes}$ , que totaliza 4M bytes

Endereço da MP: Para endereçarmos toda a MP precisamos da seguinte quantidade de bits (  $E$  )  
 sendo  $N = 2^E \Rightarrow N = 1G \text{ células} \Rightarrow N = 2^{30} \Rightarrow E = 30 \text{ bits}$

**Composição do endereço em função da memória cache**

$\Rightarrow \text{tag} = B / Q = 512M / 1M = 512 = 2^9 = 9 \text{ bits}$

$\Rightarrow n^\circ \text{ da linha: } Q = 1M \text{ linhas ou quadros (máximo)} \Rightarrow 2^{20} \Rightarrow 20 \text{ bits}$

$\Rightarrow \text{células por bloco: } 2 \text{ células por bloco} = 2^1 \Rightarrow 1 \text{ bit}$

30bits		
Tag 9 bits	No. Linha 20 bits	Célula no bloco 1 bit

b) Mapeamento totalmente associativo.

**Memória Principal**

$\Rightarrow N = 1G \text{ células}$

$\Rightarrow K = 2 \text{ células/bloco}$

$\Rightarrow B = 512 \text{ Mblocos}$

**Memória Cache**

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

$\Rightarrow Q = 1M \text{ blocos}$

$\Rightarrow$  Tamanho da memória cache = 4Mbytes

Endereço da MP = 30 bits

*Composição do endereço em função da memória cache*

$\Rightarrow \text{tag} = B = 512M \text{ blocos} = 2^{29} \Rightarrow \text{tag} = 29\text{bits}$

$\Rightarrow \text{células por bloco: } 2 \text{ células por bloco} = 2^1 \Rightarrow 1 \text{ bit}$

30bits	
Tag 29bits	Célula no bloco 1 bit

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

*Memória Principal*

$\Rightarrow N = 1G \text{ células}$

$\Rightarrow K = 2 \text{ células/bloco}$

$\Rightarrow B = 512M \text{ blocos}$

*Memória Cache*

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

$\Rightarrow Q = 1M \text{ blocos}$

$\Rightarrow \text{Tamanho da memória cache} = 4M\text{bytes}$

$\Rightarrow 1 \text{ conjunto} = 2 \text{ linhas (ou quadros)} \Rightarrow$

$\text{Total de conjuntos} \Rightarrow C = 1M \text{ blocos} / 2 \Rightarrow C = 512K \text{ conjuntos}$

*Endereço da MP = 30 bits*

*Composição do endereço em função da memória cache*

$\Rightarrow \text{tag} = B / C = 512M / 512K = 1024 = 2^{10} \Rightarrow \text{tag} = 10 \text{ bits}$

$\Rightarrow \text{nº do Conjunto: } Q = 512K \text{ linhas ou quadros (máximo)} \Rightarrow 2^{19} \Rightarrow 19 \text{ bits}$

$\Rightarrow \text{células por bloco: } 2 \text{ células por bloco} = 2^1 \Rightarrow 1 \text{ bit}$

30 bits		
Tag 10 bits	No. Conjunto 19 bits	Célula no bloco 1 bit