

## AD1 - Organização de Computadores 2011.1

### Gabarito

1. (1,0) Um computador possui um RDM com 16 bits de tamanho e um REM com capacidade para armazenar números com 20 bits. Sabe-se que a célula deste computador armazena dados com 8 bits de tamanho e que ele possui uma quantidade  $N$  de células, igual à sua capacidade máxima de armazenamento. Pergunta-se:

$$RDM = 16bits$$

$$REM = 20bits$$

$$Tamanho\ da\ célula\ (M) = 8bits$$

- a) Qual é o tamanho do barramento de endereços?

$$Barramento\ de\ endereços = REM = 20\ bits$$

- b) Quantas células de memória são lidas em uma única operação de leitura?

$$Se\ 1\ célula = 8bits\ e$$

$$RDM = 16\ bits = tamanho\ do\ barramento\ de\ dados,$$

Portanto, para um barramento de dados = 16bits = 2 células, teremos o transporte de 2 células para cada operação de leitura.

- c) Quantos bits têm a memória principal?

$$Sabendo\ que\ o\ total\ de\ bits\ T = N \times M$$

E sendo,

$$N = 2^E, \text{ como } E \text{ corresponde a quantidade de bits do barramento de endereços}$$

$$N = 2^{20} = 1\ Mcélulas$$

$$M = tamanho\ da\ célula = 8\ bits/célula$$

Então,

$$T = 1\ Mcélulas \times 8\ bits/célula = \underline{8\ Mbits}$$

2. (1,0) Uma memória associativa por conjunto consiste em 64 quadros divididos em conjuntos de 4 quadros cada. A memória principal contém 4K blocos de 128 palavras cada um. Mostre o formato de um endereço de MP.

## Memória Principal

$$\Rightarrow K = 128 \text{ (quantidade de células/bloco)}$$
$$\Rightarrow B = 4 \text{ K blocos}$$
$$\Rightarrow N = 4 \text{ K blocos} \times 128 \text{ células} = 512 \text{ K células} = 2^{19} \text{ células}$$

## Memória Cache

OBS: A quantidade de células por bloco tem de ser igual ao da MP.

$$\Rightarrow Q = 64 \text{ quadros}$$

=> *Tamanho da memória cache* =  $64 \times 128 = 8K \text{ células}$

$\Rightarrow 1 \text{ conjunto} = 4 \text{ linhas (ou quadros)} \Rightarrow$

*Total de conjuntos  $\Rightarrow C = 64 / 4 \Rightarrow C = 16$  conjuntos*

### *Memória principal*

512K células:  $N$

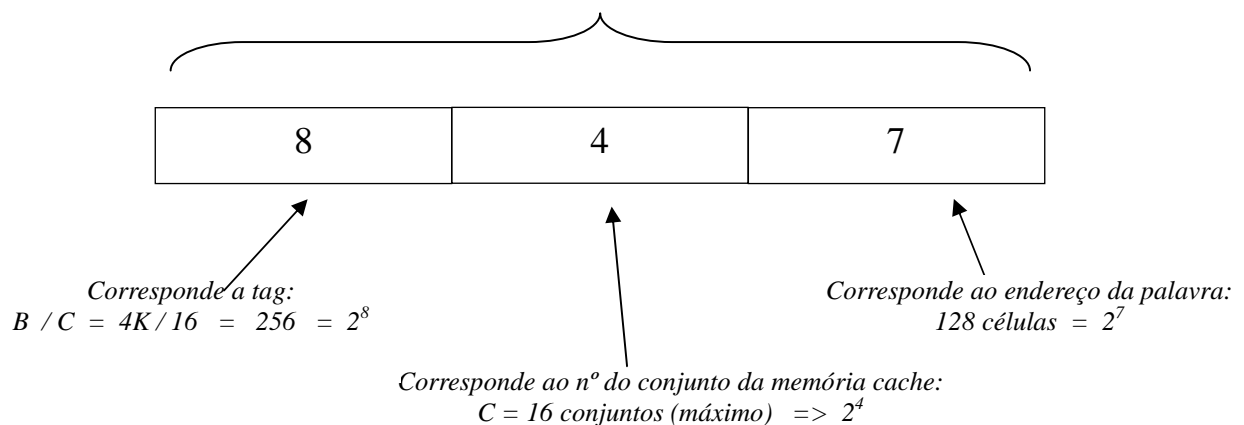
4 K blocos:  $B$

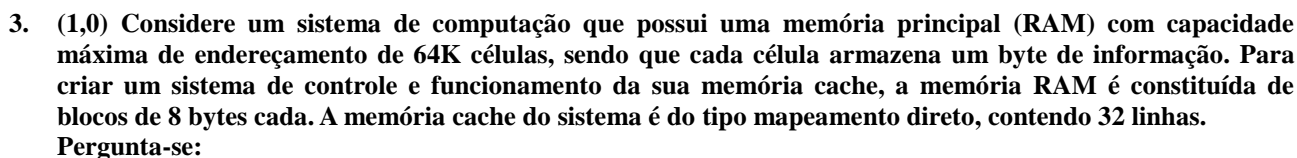
.....

Conjunto	linha	válido	tag	Conteúdo (bloco)
0	0	1 bit	8 bits	128 células
	1			
	2			
	3			
1	4			
	5			
	.....			
C - 1	Q - 2			
	Q - 1			

Para endereçarmos toda a MP precisamos da seguinte quantidade de bits:  $E = 19$  bits

Tamanho do endereço da MP = 19 bits





**c) Qual é capacidade da memória cache em bytes?**

*Capacidade da cache =  $Q \times K = 32 \text{ linhas} \times 8 \text{ palavras/linha}$ ,  
considerando neste problema, tamanho da palavra = tamanho da célula  
e tamanho da célula = 1 byte, então:*

*Capacidade da cache =  $32 \text{ linhas} \times 8 \text{ bytes} = 256 \text{ bytes}$  ou  $2^8 \text{ bytes}$*

**4. (1,0) Explique as diversas memórias que constituem a hierarquia de memória comparando-as em termos de tempo de acesso, capacidade, e temporariedade de armazenamento de dados. Explique esta hierarquia no contexto de processadores Intel Xeon.**

*O subsistema de memória é interligado de forma bem estruturada e organizado hierarquicamente em uma pirâmide com os níveis descritos a seguir.*

*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 à dos registradores e suficiente para armazenar uma apreciável quantidade de informações, sendo que, em geral, o tempo de permanência do dado é menor do que o tempo de duração do programa a que pertence.*

*Abaixo da memória cache, se encontra a memória básica de um sistema de computação (MP), que é a memória principal. Este dispositivo armazena o programa (e seus dados) que vai ser executado para que a UCP busque instrução por instrução para executá-las. As MPs são mais lentas que a cache e mais rápidas que a memória secundária, possui capacidade bem superior à da cache.*

*Finalmente, na base da pirâmide se encontra a memória secundária, memória auxiliar ou memória de massa, que fornece garantia de armazenamento permanente dos 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.*

**ARQUITETURAS MULTICORE XEON COMERCIAIS:**

**=> Xeon de 2 núcleos**

A tecnologia de dois núcleos traz dois processadores inteiros dentro de um mesmo invólucro. Como os processadores Xeon de núcleo duplo modelos 50xx, 70xx e 71xx têm a tecnologia HyperThreading – que simula a existência de dois processadores em cada núcleo – o sistema operacional reconhece cada processador Xeon de núcleo duplo como sendo quatro processadores. Assim, em um servidor com dois processadores Xeon de núcleo duplo, o sistema operacional reconhecerá oito processadores (quatro núcleos, dois por processador, e dois processadores lógicos por núcleo).

Os processadores Xeon de dois núcleos das séries 50xx, 70xx e 71xx são baseados na microarquitetura do Pentium 4 (NetBurst) e por isso possuem a tecnologia HyperThreading, que não está presente na microarquitetura Core.

Os processadores Xeon 31xx e 52xx, assim como os modelos 30xx, 51xx e 72xx, são baseados na microarquitetura Core. A principal diferença entre esses modelos é a tecnologia de fabricação. Enquanto os modelos 30xx, 51xx e 72xx usam o processo de fabricação de 65 nm, as séries 31xx e 52xx usam o novo processo de 45 nm.

*Principais características dos Xeon das séries 50xx, 70xx e 71xx:*

- => Tecnologia de dois núcleos*
- => Mesma arquitetura interna no Pentium 4 (NetBurst)*
- => Cache L1 de dados de 16 KB e cache de execução de 150 KB.*
- => Cache L2 de 2 ou 4 MB compartilhado*
- => Cache L3 interna de 4 MB, 8 MB ou 16 MB (somente nos modelos 71xx).*
- => Suporte a multiprocessamento simétrico com até dois processadores por placa-mãe.*

*Principais características dos Xeon 30xx, 51xx e 72xx Microarquitetura Core*

- 1. Tecnologia de dois núcleos*
- 2. Microarquitetura Core*
- 3. Tecnologia de fabricação de 65 nm*

4. Cache L1 dividido, sendo 32 KB para dados e 32 KB para instruções por núcleo
5. Cache L2 de 4 MB compartilhado entre os núcleos

Principais características dos Xeon 31xx e 52xx

- Tecnologia de dois núcleos
- Microarquitetura Core
- Tecnologia de fabricação de 45 nm
- Cache L1 dividido, sendo 32 KB para dados e 32 KB para instruções por núcleo
- Cache L2 de 6 MB compartilhado

=> **Xeon de 4 núcleos**

Os processadores Xeon Séries 33xx e 54xx, assim como os modelos 32xx, 53xx e 73xx, são baseados na microarquitetura Core, a mesma usada pelos processadores Core 2 Duo. Enquanto os modelos 32xx, 53xx e 73xx usam o processo de fabricação de 65 nm, as séries 33xx e 54xx usam o novo processo de 45 nm..

Os quatro núcleos dos processadores Xeon 32xx, 53xx e 73xx bem como as séries 33xx e 54xx são obtidos a partir de duas pastilhas de dois núcleos cada, assim como ocorre com os modelos descritos na página anterior. Com isso, o cache L2 desses processadores não é compartilhado entre todos os seus núcleos: os núcleos 1 e 2 compartilham um mesmo cache L2, enquanto que os núcleos 3 e 4 compartilham um outro cache L2. O valor divulgado é o valor total (soma dos dois caches). Leia o nosso artigo Visão Geral dos Futuros Processadores de Quatro Núcleos da Intel para uma explicação mais detalhada sobre a arquitetura usada por estes processadores.

Principais características dos processadores Xeon das séries 32xx, 53xx e 73xx:

- a) Microarquitetura Core
- b) Tecnologia de quatro núcleos
- c) Tecnologia de fabricação de 65 nm
- d) Cache L1 dividido, sendo 32 KB para dados e 32 KB para instruções por núcleo
- e) Cache L2 de 4 MB, 6 MB ou 8 MB, dependendo do modelo, dividido em dois

Principais características dos processadores Xeon 33xx e 54xx:

- Microarquitetura Core
- Tecnologia de quatro núcleos
- Tecnologia de fabricação de 45 nm
- Cache L1 dividido, sendo 32 KB para dados e 32 KB para instruções por núcleo.
- Cache L2 de 6 MB ou 12 MB dividido em dois

5. (1,0) Considere uma máquina que possa endereçar 512 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 8K blocos, sendo um bloco por linha. Mostre o formato da memória cache, indicando os campos necessários (válido, 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:

a) Mapeamento direto.

**Memória Principal**

⇒ Tamanho da memória (em bytes) = 512Mbytes, como 1 célula referencia a 1 byte, temos  $N = 512M$  células

⇒ Será organizada em blocos de 16 bytes, como 1 célula = 1 byte, temos cada bloco = 16 células,  $K = 16$

⇒ Sendo  $N$  o tamanho endereçável da memória e  $K$  que é a quantidade de células por blocos temos:

$N = 512M$  células e  $K = 16$  células/blocos o total de blocos da MP ( $B$ ) será:

Total de blocos:  $B = N / K \Rightarrow B = 512M \text{ células} / 16 \text{ células/bloco} \Rightarrow B = 32 M \text{ blocos}$

**Memória Cache**

OBS: A quantidade de células por bloco tem de ser igual a MP.

⇒ Tamanho da memória cache (em blocos ou linhas)  $\Rightarrow Q = 8K \text{ blocos}$

⇒ Tamanho da memória cachê em células  $= Q \times K = 8K \text{ blocos} \times 16 \text{ células/blocos} = 128K \text{ células}$

• • •

linha	válido	tag	Conteúdo (bloco)
0	1 bit	12 bits	16 células de 8 bits cada = 128 bits
1			
2			
3			
4			
5			
.....			
Q - 2			
Q - 1			

Corresponde ao endereço da célula no bloco:  
 $16 \text{ células por bloco} = 2^4$

**b) Mapeamento totalmente associativo.**

**Memória Principal**

=>  $N = 512M$  células

=>  $K = 16$

=>  $B = 32 M$  blocos

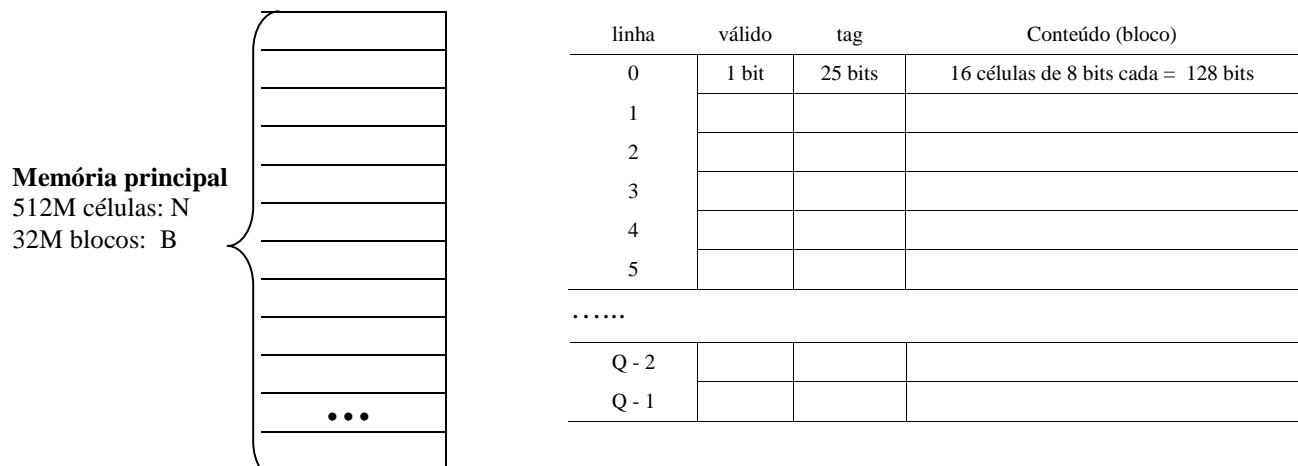
**Memória Cache**

*OBS: A quantidade de células por bloco tem de ser igual a MP.*

=>  $Q = 8K$  blocos

=> *Tamanho da memória cache = 128K células*

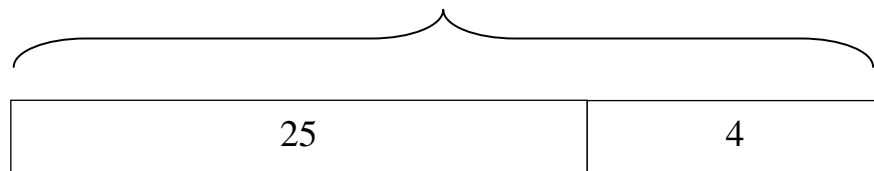
**Organização da cache**



Para endereçarmos toda a MP precisamos da seguinte quantidade de bits:  $E = 29$  bits

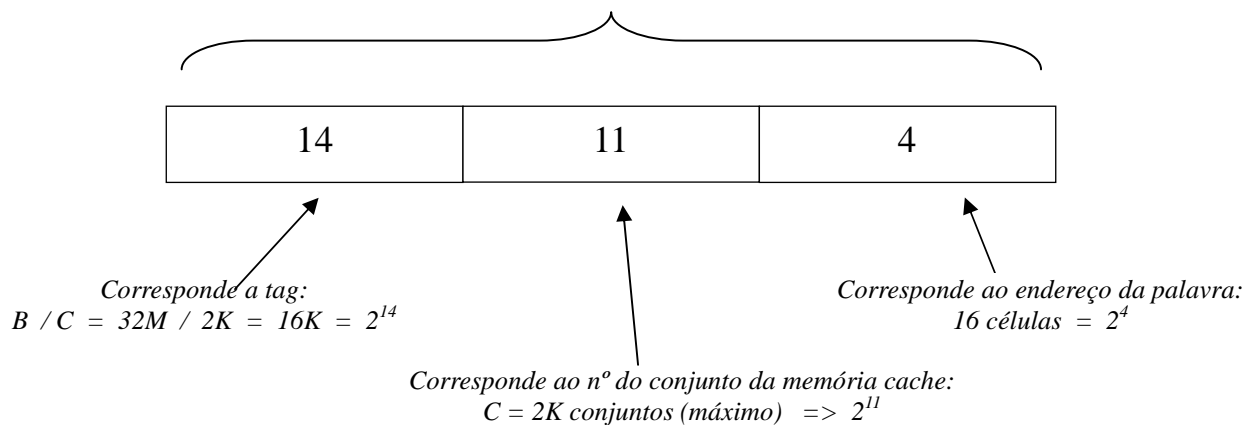
Como o bloco pode ser alocado em qualquer posição da memória cache a tag indicará qual dos blocos da MP está alocado naquela posição da memória cache

Tamanho do endereço da MP = 29 bits



Corresponde ao bloco da MP:  
 $tag = B = 32M = 2^{25}$

Corresponde ao endereço da palavra:  
 $16 \text{ células por bloco} = 2^4$





6. (1,0) Considere a máquina apresentada na aula 4. Descreva detalhadamente (do mesmo modo que é apresentado na aula 4) como é realizada a execução das seguintes instruções:

a) LDA 40

Passo 1:  $RI \leftarrow$  Instrução lida

Passo 2:  $CI \leftarrow CI + 1$

Passo 3: Decodificação do código de operação

Passo 3.1: recebe os bits do código de operação

Passo 3.2: produz sinais internos para a execução da operação de leitura em memória

Passo 4: Busca do operando na memória

Passo 4.1: A UC emite sinais para que o valor do campo operando = 40 seja transferido para a REM

Passo 4.2: A UC emite sinais para que o valor contido no REM seja transferido para o barramento de endereços

Passo 4.3: A UC ativa a linha READ do barramento de controle

Passo 4.4: Conteúdo da posição da memória com endereço contido no barramento de endereços (40) é transferido através do barramento de dados para o RDM

Passo 4.5: O conteúdo do RDM é transferido para o registrador acumulador ( $ACC \leftarrow RDM$ )

b) JP 150

Passo 1: A CPU verifica se o valor contido no ACC é positivo (maior que zero)

Passo 1.1: Caso seja verdadeira a verificação:  $CI \leftarrow Op$ . Sendo  $Op = 150$ .

Será executada a instrução do endereço 150.

Passo 1.2: Caso seja falsa a verificação:  $CI \leftarrow CI + 1$ .

Será executada a instrução do endereço seguinte

7. (1,0) Escreva um programa que utilize as instruções de linguagem de montagem apresentadas na aula 4 para executar o seguinte procedimento. O conteúdo da memória cujo endereço é 30 é lido e verifica-se se o seu valor é maior que 0. Caso seu valor seja maior que 0, ele é subtraído do conteúdo de memória cujo endereço é 40 e o resultado é armazenado no endereço 50. Caso contrário, ele é somado ao conteúdo de memória cujo endereço é 40 e o resultado é armazenado no endereço 50.

Além de apresentar seu programa escrito em linguagem de montagem, apresente também o programa traduzido para linguagem de máquina. Os endereços estão expressos em decimal e portanto devem ser convertidos para a base 2 utilizando 8 bits, quando o programa for convertido para linguagem de máquina.

Endereço	Instrução	Descrição	Linguagem Máquina (bin / hexa)
10	LDA 30	$ACC \leftarrow (30)$	( 000100011110 / 11E )
11	JP 16	$CI \leftarrow 16$ se $ACC > 0$	( 011000010000 / 610 )
12	LDA 40	$ACC \leftarrow (40)$	( 000100101000 / 128 )
13	SUB 30	$ACC \leftarrow ACC - (30)$	( 010000011110 / 41E )
14	STR 50	$(50) \leftarrow ACC$	( 001000110010 / 232 )
15	HLT	Encerra	( 000000000000 / 000 )
16	ADD 40	$ACC \leftarrow ACC + (40)$	( 001100101000 / 328 )
17	STR 50	$(50) \leftarrow ACC$	( 001000110010 / 232 )
18	HLT	Encerra	( 000000000000 / 000 )

Outra solução,

Endereço	Instrução	Descrição	Linguagem Máquina (bin / hexa)
10	LDA 30	$ACC \leftarrow (30)$	( 000100011110 / 11E )
11	JP 15	$CI \leftarrow 15$ se $ACC > 0$	( 011000001111 / 60F )
12	LDA 40	$ACC \leftarrow (40)$	( 000100101000 / 128 )
13	SUB 30	$ACC \leftarrow ACC - (30)$	( 010000011110 / 41E )
14	JMP 16	$CI \leftarrow 16$	( 100000010000 / 810 )
15	ADD 40	$ACC \leftarrow ACC + (40)$	( 001100101000 / 328 )
16	STR 50	$(50) \leftarrow ACC$	( 001000110010 / 232 )
17	HLT	Encerra	( 000000000000 / 000 )

8. (1,0) Considere uma máquina com arquitetura semelhante àquela apresentada em aula. Pode-se endereçar no máximo 1G células de memória, sendo que cada célula tem tamanho igual a 8 bits. Em cada acesso à memória, obtém-se o conteúdo de uma célula. Esta máquina possui 31 registradores. Todas as instruções desta máquina têm tamanho de 40 bits e possuem três campos: o primeiro indica o código de operação, o segundo indica endereço de célula de memória onde se encontra o operando e o terceiro um identificador de registrador.

- a) Calcule a capacidade mínima de endereçamento em bits do REM, considerando que os bits armazenados no REM são utilizados para endereçar uma célula de memória.

$$REM = E = \text{tamanho em bits necessários para acessar toda a memória (N)}$$

$$N = 2^E = 1G \text{ células} = 2^{30} \text{ células} \Rightarrow E = 30 \Rightarrow REM = 30 \text{ bits}$$

- b) Calcule o número de bits que devem poder ser transmitidos no barramento de endereços em cada acesso à memória.

$$\text{Barramento de endereços} = REM = 30 \text{ bits}$$

- c) Indique o tamanho do RI (Registrador de Instruções).

$$\text{Tamanho mínimo de RI} = \text{tamanho da instrução}$$

$$\text{Tamanho da instrução} = 40 \text{ bits}$$

$$\text{Tamanho mínimo de RI} = 40 \text{ bits}$$

- d) Calcule o número de células que uma instrução ocupa.

$$\text{Tamanho mínimo de 1 instrução} = 40 \text{ bits}$$

$$\text{Tamanho de 1 célula} = 8 \text{ bits}$$

$$\text{Serão necessárias pelo menos 5 células para uma instrução}$$

- e) Calcule o número de códigos de operação diferentes que o conjunto de instruções desta máquina pode ter.

$$\text{Tamanho da instrução} = \text{cód.oper.} + 1 \text{ oper. (endereço memória)} + 1 \text{ oper. (registrador)}$$

$$\text{Tamanho da instrução} = 40 \text{ bits}$$

$$\text{Endereço de memória} = 30 \text{ bits}$$

$$\text{Endereço registrador} = 5 \text{ bits (necessário para endereçar cada um dos 31 registradores)}$$

$$40 = \text{cód.oper.} + 30 + 5$$

$$\text{Cód.oper.} = 5 \text{ bits, portanto poderemos ter até } 32 (2^5) \text{ códigos de operação diferentes}$$

- f) Calcule a capacidade máxima de armazenamento da memória deste sistema em bits.

$$T = N \times M \Rightarrow T = 1G \text{ células} \times 8 \text{ bits/célula} \Rightarrow T = 8 \text{ Gbits}$$

- g) Calcule a capacidade mínima em bits do CI (Contador de Instrução), considerando que os bits armazenados no CI são utilizados para endereçar a primeira célula de uma instrução armazenada na memória.

$$CI = \text{tamanho necessário para acessar toda a memória} = 30 \text{ bits}$$

9. (0,5) Considere uma máquina cujo relógio possui uma frequência de 2 GHz e um programa P1 no qual são executadas 5000 instruções.

- a) Calcule o tempo para executar o programa P1, considerando que cada instrução é executada em 12 ciclos de relógio e a execução de uma instrução só se inicia quando a execução da instrução anterior é finalizada.

$$\text{Tempo de um ciclo de relógio} = 1/2.000.000.000 = 0,000\ 000\ 000\ 5 \text{ seg ou } 0,5 \text{ ns (nanossegundos)}$$

$$\text{Tempo de execução de 1 instrução} = 12 \text{ ciclos de relógio} = 12 \times 0,5 \text{ ns} = 6 \text{ ns}$$

$$1000 \text{ instruções executadas sequencialmente} = 5000 \times 6 \text{ ns} = 30.000 \text{ ns ou } 30 \mu\text{s}$$

- b) Uma nova implementação dessa máquina utiliza um pipeline de 2 estágios, todos de duração igual a 8 ciclos de relógio. Calcule o tempo para executar uma instrução nesta nova máquina e o tempo para executar o programa P1, considerando que não existem conflitos de qualquer tipo.

$$\text{Tempo para um estágio} = 8 \text{ ciclos de relógio} = 8 \times 0,5 \text{ ns} = 4 \text{ ns}$$

$$\text{Para execução da 1ª instrução} = 2 \text{ estágios} \times 4 \text{ ns} = 8 \text{ ns}$$

$$\text{Para execução das instruções posteriores} = \text{tempo de 1 estágio devido ao pipeline} = 4 \text{ ns}$$

$$\text{Tempo total para execução das 5000 instruções} = 8 \text{ ns} + 4\ 999 \times 4 \text{ ns} = 20.004 \text{ ns}$$

- c) Uma nova implementação dessa máquina utiliza um pipeline de 6 estágios, todos de duração igual a 3 ciclos de relógio. Calcule o tempo para executar uma instrução nesta nova máquina e tempo para executar o programa P1, considerando que não existem conflitos de qualquer tipo.

*Tempo para um estágio = 3 ciclos de relógio =  $3 \times 0,5ns = 1,5ns$*

*Para execução da 1ª instrução =  $6 \text{ estágios} \times 1,5ns = 9ns$*

*Para execução das instruções posteriores = tempo de 1 estágio devido ao pipeline =  $1,5ns$*

*Tempo total para execução das 5000 instruções =  $9ns + 4\,999 \times 1,5ns = 7.507,5ns$*

- d) Indique qual das 3 máquinas executa uma instrução em menos tempo e qual das três executa o programa P1 em menos tempo.

*A arquitetura do item a executa uma instrução em menos tempo. Já a arquitetura do item c executa o programa P1 em um tempo total menor.*

**10. (0,5) Explique a arquitetura pipeline do Pentium 4 (dica: consulte os artigos do site Clube do Hardware (<http://www.clubedohardware.com.br/>))**

*Texto base retirado do site [www.clubedohardware.com.br](http://www.clubedohardware.com.br/)*

*O pipeline do Pentium 4 possui 20 estágios, bem maior que seu antecessor, o Pentium III com 11 estágios. Este aumento no número de estágios do pipeline nos processadores da Intel foi para que os processadores rodassem com clocks mais elevados. Com mais estágios, cada um deles poderia ser construído com poucos transistores o que facilitaria a obtenção de clocks maiores.*

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
TC	Nxt IP	TC	Fetch	Drive	Alloc	Rename	Que	Sch	Sch	Sch	Disp	Disp	RF	RF	Ex	Flgs	Br	Ck	Drive

*Fig 1. Pipeline do Pentium IV (fonte: [www.clubedohardware.com](http://www.clubedohardware.com))*

*A Fig1 mostra os 20 estágios do pipeline do Pentium IV e logo abaixo está uma explicação básica de cada um dos estágios do pipeline.*

- *TC Nxt IP, Trace cache next instruction pointer (Ponteiro da próxima instrução do cache de microinstruções): Este estágio obtém no buffer de destino de desvio (BTB) qual é a próxima microinstrução a ser executada. Esta etapa demora 2 estágios.*
- *TC Fetch, Trace cache fetch (Busca da próxima microinstrução no cache de microinstruções): Carrega a microinstrução do cache de microinstrução. Esta etapa demora 2 estágios.*
- *Drive: Envia a microinstrução a ser executada à unidade de alocação de recursos e de renomeamento de registradores.*
- *Alloc (Alocação de recursos): Verifica quais recursos serão alocados pela microinstrução, como, por exemplo, o buffer de carga (load) ou o buffer de armazenamento (store).*
- *Rename (Renomeamento de Registradores): Renomeia os oito registradores padrão x86 em um dos 128 registradores internos presentes no Pentium 4. Esta etapa demora 2 estágios.*
- *Queue (Fila): As microinstruções são armazenadas em filas de acordo com o seu tipo (por exemplo, inteiro ou ponto flutuante). Elas são mantidas na fila até haver espaço no escalonador de execução correspondente ao tipo de instrução a ser executada.*
- *Sch, Schedule (Escalonamento): As microinstruções são escalonadas de acordo com o seu tipo (inteiro, ponto flutuante, etc). Antes de chegar nesta etapa, as microinstruções são entregues em ordem, ou seja, na mesma ordem em que apareciam no programa. Nesta etapa, o escalonador reordena as instruções de forma que estas sejam executadas da melhor forma possível, isto é, que não fique nenhuma unidade de execução vazia. Por exemplo, se uma unidade de ponto flutuante vai ficar disponível, o escalonador procurará por uma instrução de ponto flutuante para mandá-la para esta unidade, mesmo que a próxima instrução no programa seja uma instrução que opere com números inteiros. Os escalonadores são o coração da execução fora de ordem dos processadores de 7ª geração. Esta etapa demora três estágios.*
- *Disp, Dispatch (Envio): As microinstruções são enviadas às unidades de execução adequadas. Esta etapa é dividida em dois estágios.*
- *RF (Leitura dos registradores internos): Os registradores internos, armazenados no pool de instruções, são lidos. Esta etapa demora dois estágios.*
- *Ex (Execução): As microinstruções são executadas.*
- *Flgs (Flags): Atualiza os flags do processador.*
- *Br Ck, Branch check (Verificação de desvios): Verifica se o desvio tomado pelo programa é o mesmo que o*

*previsto pelo circuito de previsão de desvio.*

- 11. (1,0) Descreva o funcionamento de uma unidade de controle microprogramada e uma unidade implementada em hardware e indique uma vantagem e uma desvantagem de cada uma delas.**

***Unidade de controle microprogramada***

*A unidade de controle microprogramada é utilizada para se desenvolver a implementação de um conjunto de instruções que apresenta muita complexidade para ser implementado somente em hardware. A execução de uma instrução é composta da execução de microinstruções referentes a ela e a unidade de controle microprogramada é projetada de modo a executar estas microinstruções. Ela é composta por: Memória de controle, Contador de microprograma e Sequenciador.*

*A Memória de controle armazena as microinstruções que compõem uma instrução e o Contador de microprograma armazena a localização da próxima microinstrução a ser executada. O Sequenciador é o componente que controla a sequencia de execução das microinstruções, informando o local da próxima microinstrução que deve ser executada e armazenada no Contador de microprograma.*

*A vantagem de se utilizar uma unidade de controle microprogramada é a maior facilidade do seu projeto, da sua implementação e da verificação de erros, pois o número de expressões booleanas pode ser muito grande, o que pode tornar difícil a implementação por hardware de um conjunto de instruções que possua um grande número de instruções. A desvantagem de uma unidade de controle microprogramada é ser mais lenta que uma unidade de controle implementada por hardware.*

***Unidade de controle por hardware***

*Em uma unidade controlada por hardware, o seu desenvolvimento consiste essencialmente em projetar circuitos combinatórios. Os sinais lógicos de entrada na unidade devem ser transformados em um conjunto lógico de sinais que controlam a execução da instrução. Para implementar a unidade, necessita-se derivar, para cada sinal de controle a ser gerado para que cada instrução seja executada de forma correta, uma expressão booleana que defina esse sinal em função dos sinais de entrada referentes à instrução.*

*Nesta unidade de controle, as microinstruções serão executadas diretamente pelo hardware.*

*A desvantagem está na dificuldade de implementar de forma correta e segura um conjunto com um grande número de instruções porque é gerado um grande número de expressões booleanas que devem ser implementados em circuitos. A vantagem da implementação por hardware sobre microprogramação está na velocidade de execução das instruções.*