

1. **Descreva as funções de uma UCP.**

- Executar os programas armazenados na memória principal, buscando suas instruções, examinando as, e então executando uma após a outra.
 - Busca de instruções e dados na memória.
 - Programa a transferência de dados entre a memória e os dispositivos de entrada/saída.
 - Decodifica as instruções.
 - Realiza as operações lógicas e aritméticas.
 - Responde a sinais enviados por dispositivos de entrada/saída como RESET ou interrupções.
-

2. **Indique os dois atributos da memória e quais são as operações que a UCP pode solicitar.**

- Memória: Primária e Secundária, Volátil e não volátil
 - Operações da UCP: Aritmética e Lógica
-

3. **Para que servem os dispositivos de entrada e saída de um computador? Cite alguns exemplos.**

- Os dispositivos de entrada e saída (E/S) permitem a comunicação entre o usuário e o computador, bem como entre o computador e outros dispositivos. Exemplos de dispositivos de entrada incluem teclado e mouse, enquanto dispositivos de saída incluem monitores e impressoras. Dispositivos como discos rígidos e pen drives podem funcionar como entrada e saída.
-

4. **Formalize o conceito de bit, byte e palavra.**

- **Bit:** A menor unidade de informação em um computador, que pode representar dois estados (0 ou 1).
 - **Byte:** Conjunto de 8 bits. É a unidade básica de armazenamento usada para representar um caractere, como uma letra ou número.
 - **Palavra:** Conjunto de bits que o processador manipula como uma unidade única. O tamanho da palavra varia conforme a arquitetura do sistema, podendo ser, por exemplo, 16, 32 ou 64 bits.
-

5. **Qual é a diferença entre linguagem de alto nível e linguagem de máquina?:**

- **Linguagem de Alto Nível:** Linguagens como Python, C ou Java, que são mais fáceis para humanos entenderem, utilizando sintaxes semelhantes a palavras e frases do cotidiano.
 - **Linguagem de Máquina:** Conjunto de instruções em formato binário que o processador entende diretamente. Estas instruções são mais difíceis para os humanos escreverem e interpretar, mas são necessárias para a execução direta no hardware.
-

6. **Considere um sistema de computador que possua u processador capaz de endereçar, no máximo, 32 M endereços de memória principal. Qual deverá ser o tamanho, em bits, do barramento de endereços?**

Número de endereços = 2^n

Onde:

- n é o número de bits do barramento de endereços;
- O número de endereços é 32M, que significa (32×10^6) endereços.

Primeiro, transformamos ($32M$) em potência de 2:

$$32M = 32 \times 2^{20} = 2^5 \times 2^{20} = 2^{25}$$

Agora, igualamos:

$$2^n = 2^{25}$$

Logo:

$$n = 25$$

Portanto, o tamanho do barramento de endereços deverá ser **25 bits**.

7. **Um computador, cuja memória RAM (MP) tem uma capacidade máxima de armazenamento de 2K palavras de 16 bits cada, possui um REM e um RDM. Qual é o tamanho desses registradores? Qual é o valor do maior endereço dessa MP e qual é a quantidade total de bits que nela podem ser armazenados?**

Tamanho do REM (Registrador de Endereços de Memória): O REM armazena o endereço de uma palavra na memória. A memória tem 2K palavras, onde $K = 1024$, então:

$$2K = 2 \times 1024 = 2048 \text{ palavras}$$

Para endereçar 2048 palavras, o número de bits necessários no REM é dado por $2^n = 2048$, onde n é o número de bits:

$$2^{11} = 2048$$

Logo, o **REM deve ter 11 bits** para endereçar todas as palavras da memória.

Tamanho do RDM (Registrador de Dados de Memória): O **RDM** armazena os dados de uma palavra. Cada palavra na memória tem 16 bits, então o **RDM deve ter 16 bits**, já que ele precisa ser capaz de armazenar uma palavra inteira.

Valor do maior endereço da MP: O maior endereço possível é o último endereço da memória, que é o endereço 204720472047 (em decimal), pois os endereços começam do 0. Portanto, o valor do maior endereço é **2047**.

Quantidade total de bits que podem ser armazenados na memória: A memória tem 2048 palavras, e cada palavra tem 16 bits. Assim, a quantidade total de bits que podem ser armazenados é:

$$2048 \times 16 = 32768 \text{ bits} = 32K \text{ bits}$$

8. **Um processador possui RDM com capacidade de armazenar 32 bits e um REM com capacidade de armazenar 24 bits. Sabendo-se que em cada acesso são lidas duas células da memória RAM (MP) e que o barramento de dados em tamanho igual ao da palavra, pergunta-se:**

- a) **Qual é a capacidade máxima de endereçamento do microcomputador em questão?**

A capacidade máxima de endereçamento depende do número de bits do **REM** (Registrador de Endereços de Memória). O REM tem 24 bits, então o número total de endereços que o microcomputador pode acessar é:

$$2^{24} = 16.777.216 \text{ endereços}$$

Portanto, o microcomputador pode endereçar **16.777.216 (ou 16M) endereços**.

- b) **Qual é o total máximo de bits que podem ser armazenados na memória RAM?**

Sabemos que em cada acesso à memória são lidas **duas células**. Como o **RDM** (Registrador de Dados de Memória) tem 32 bits, cada palavra tem 32 bits. Isso significa que cada célula armazena metade de uma palavra, ou seja, **16 bits** por célula.

Assim, a memória tem 2^{24} endereços, e cada endereço (ou célula) armazena 16 bits. O total de bits que podem ser armazenados na memória RAM é:

$$16 \times 2^{24} = 16 \times 16.777.216 = 268.435.456 \text{ bits}$$

Portanto, o total máximo de bits que podem ser armazenados na memória RAM é **268.435.456 bits** ou **256 Megabits (Mb)**.

- c) **Qual é o tamanho da palavra e de cada célula da máquina?**

O tamanho da **palavra** é igual à capacidade do **RDM**, ou seja, **32 bits**.

O tamanho de cada **célula** é metade de uma palavra, já que o enunciado menciona que em cada acesso são lidas duas células. Portanto, o tamanho de cada célula é **16 bits**.

9. **Um processador possui um barramento de endereços com capacidade de permitir a transferência de 33 bits de cada vez. Sabe-se que o barramento de dados permite a transferência de 4 palavras em cada acesso e que cada célula da memória RAM armazena 1/8 de cada palavra. Considerando que a memória RAM pode armazenar um máximo de 64 Gbits, pergunta-se:**

a) Qual é a quantidade máxima de células que podem ser armazenadas na memória RAM?

Tamanho total da memória (em bits): $64 \text{ Gbits} = 64 \times 10^9 \text{ bits}$.

Tamanho de cada célula (em bits): Cada célula armazena 1/8 de uma palavra, ou seja, o tamanho de uma palavra é 8 vezes o tamanho de uma célula.

Se memória total = 64×10^9 , e uma célula armazena $\frac{1}{8}$ de uma palavra, a quantidade máxima de $Q_{\text{células}}$ é simplesmente o número de bits que podem ser armazenados (64 bilhões de bits), já que estamos falando de bits individuais:

$$Q_{\text{células}} = 64 \times 10^9 \text{ bits}$$

Então, **a quantidade máxima de células é 64 bilhões de células.**

b) Qual é o tamanho do REM e do barramento de dados existentes neste computador?

Com um barramento de 33 bits, o número máximo de endereços que podem ser representados é:

$$2^{33} = 8.589.934.592 \text{ endereços.}$$

No entanto, como a memória só tem 64 bilhões de células (ou $6,4 \times 10^{10}$ bits), o número de bits no REM corresponde ao que pode ser endereçado pelo barramento de 33 bits.

Sabemos que o barramento de dados transfere 4 palavras em cada acesso. Se considerarmos que cada palavra tem 8 bits, isso significa que o barramento de dados tem que ser capaz de transferir 32 bits por vez ($4 \text{ palavras} \times 8 \text{ bits por palavra} = 32 \text{ bits}$).

c) Qual é o tamanho de cada célula e da palavra desta máquina?

Tamanho da célula: 1 bit, tamanho da palavra: 8 bits.

10. **Um computador possui um RDM com 16 bits de tamanho e um REM com capacidade para armazenar números com 2º 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:

a) Qual é o tamanho do barramento de endereços?

Um barramento de endereços com 20 bits pode endereçar 2^{20} posições de memória, pois o número de endereços possíveis é dado por 2^n , onde n é o número de bits no barramento de endereços.

$$2^{20} = 1.048.576 \text{ células.}$$

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

Como o RDM é o registrador responsável por conter os dados transferidos do barramento de dados, ele precisa ser capaz de armazenar os dados de uma operação de leitura. Se o RDM tem 16 bits e cada célula tem 8 bits, então em uma única operação de leitura, o RDM consegue ler:

$$\frac{16 \text{ bits (RDM)}}{8 \text{ bits por célula}} = 2 \text{ células.}$$

c) Quantos bits tem a MP?

Para calcular o número total de bits da memória principal (MP), basta multiplicar a quantidade total de células pelo número de bits armazenados em cada célula:

$$MP = 2^{20} \text{ células} \times 2^3 \text{ bits por célula} = 2^{23} \text{ bits.}$$

11. Um microcomputador possui uma capacidade máxima de MP com 32 K células, cada uma capaz de armazenar uma palavra de 8 bits. Pergunta-se:

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

Sabemos que a memória principal (MP) possui **32 K células**. O número "K" representa **1024** ou 2^{10} , logo o número total de células é:

$$32 \times 1024 = 2^5 \times 2^{10} = 2^{15} = 32.768 \text{ células.}$$

O **maior endereço** de uma memória é sempre igual ao número total de células menos 1 (porque o endereçamento começa do zero). Portanto, o maior endereço, em decimal, é:

$$32.768 - 1 = 32.767.$$

Portanto, **o maior endereço desta memória, em decimal, é 32.767.**

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

O barramento de endereços precisa ser capaz de endereçar todas as **32.768 células** de memória. O tamanho do barramento de endereços em bits é determinado pelo número de endereços que ele pode gerar, que é dado por 2^n , onde n é o número de bits no barramento.

Sabemos que:

$$2^{15} = 32.768.$$

Portanto, o **barramento de endereços tem 15 bits**

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

RDM (Registrador de Dados de Memória): O RDM armazena uma **palavra** de dados. Como cada célula da memória armazena uma palavra de **8 bits**, o RDM deve ser capaz de armazenar **8 bits**.

d) Qual é o total máximo de bits que pode existir nesta memória?

Para determinar o total máximo de bits que a memória pode armazenar, basta multiplicar o número total de células pela quantidade de bits que cada célula pode armazenar.

Sabemos que a memória tem **32.768 células**, e cada célula armazena **8 bits**. Logo, o total máximo de bits é:

$$32.768 \text{ células} \times 8 \text{ bits por célula} = 2^{15} \times 2^3 = 2^{17} = 262.144 \text{ bits.}$$

Como Kbits = 2^{10} , logo, 2^7 Kbits = 256 Kbits

12. **Considere uma célula de uma MP cujo endereço é, em hexadecimal, 2C81 e que tem armazenado em seu conteúdo um valor igual a, em hexadecimal, F5A. Sabe-se que, neste sistema, as células têm o mesmo tamanho das palavras e que em cada acesso é lido o valor de uma célula. Pergunta-se:**

a) Qual deve ser o tamanho do REM e do RDM neste sistema?

- REM (Registrador de Endereçamento de Memória):

O REM precisa armazenar o endereço da célula de memória. O endereço fornecido é **2C81** (em hexadecimal). Em formato binário, esse valor é:

$$2C81_{\text{hex}} = 0010110010000001_{\text{bin}} = 16 \text{ bits.}$$

- RDM (Registrador de Dados de Memória):

O valor armazenado na célula é **F5A** (hexadecimal), que em binário é:

$$F5A_{\text{hex}} = 111101011010_{\text{bin}} = 12 \text{ bits.}$$

b) Qual deve ser a máxima quantidade de bits que podem ser implementados nessa memória?

O número máximo de células endereçáveis é:

$$2^{16} = 65.536 \text{ células.}$$

Como cada célula armazena **12 bits** (tamanho do RDM), a capacidade máxima de armazenamento da memória é:

$$65.536 \text{ células} \times 12 \text{ bits por célula} = 786.432 \text{ bits.}$$

13. **Uma memória ROM pode ser também considerada uma memória do tipo Leitura/Escrita? Por que?**

Não, a memória ROM (Read-Only Memory) não pode ser considerada uma memória do tipo Leitura/Escrita. Isso ocorre porque o próprio nome da memória ROM indica que ela é apenas de leitura ("read-only"), ou seja, os dados armazenados nela são gravados uma única vez (normalmente durante o processo de fabricação ou, em alguns casos, com processos especiais de gravação) e não podem ser modificados ou apagados pelo usuário ou pelo sistema durante o uso normal.

14. **Qual é a vantagem do uso de muitos registradores em um processador?**

(Acesso rápido aos dados, Redução de operações de leitura/escrita na memória, Execução eficiente de operações aritméticas e lógicas)

Aumenta a eficiência do processador, acelerando o acesso aos dados, reduzindo a dependência da memória principal e melhorando o desempenho geral do sistema, especialmente em operações complexas e paralelas.

15. **Por que não é possível a MP ser totalmente volátil?**

No entanto, o sistema precisa manter certos dados persistentes, como o sistema operacional, configurações essenciais e outros programas importantes. Se a memória fosse completamente volátil, todas essas informações seriam perdidas a cada desligamento, tornando o sistema inutilizável após a reinicialização

16. **Sempre que o processador realiza um acesso à MP para efetuar uma operação de leitura ou de escrita, ele manipula dois valores distintos, mas que estão associados ao acesso. Quais são esses valores?**

Os dois valores manipulados pelo processador ao acessar a MP são:

1. **Endereço de memória:** O endereço onde o dado será lido ou escrito.
2. **Dados:** O valor que será armazenado (escrito) ou recuperado (lido) da memória no endereço especificado.

Esses dois valores são gerenciados pelos registradores **REM** (Registrador de Endereçamento de Memória) e **RDM** (Registrador de Dados de Memória), respectivamente.

17. **Quantos bits são requeridos para se endereçar células em uma memória de 128 G endereços?**

$$128\text{ G} = 128 \times 2^{30} \text{ endereços.}$$

Precisamos encontrar o valor de n em 2^n que seja capaz de endereçar essa quantidade. Logo:

$$2^n = 128 \times 2^{30} = 2^7 \times 2^{30} = 2^{37}.$$

Portanto, **são necessários 37 bits** para endereçar uma memória de 128 G endereços.

18. **E quantos bits seriam requeridos se a memória tivesse 32 K endereços?**

Para uma memória de **32 K** endereços:

$$32\text{ K} = 32 \times 2^{10} \text{ endereços.}$$

Precisamos encontrar o valor de n em 2^n que seja capaz de endereçar essa quantidade. Logo:

$$2^n = 32 \times 2^{10} = 2^5 \times 2^{10} = 2^{15}.$$

Portanto, **são necessários 15 bits** para endereçar uma memória de 32 K endereços.

19. **Você considera válida a afirmação “vale aumentar a capacidade da memória principal para que o acesso aos meios magnéticos (discos rígidos e disquetes) seja mais rápido”?**

Não, a afirmação não é válida. Aumentar a capacidade da memória principal (RAM) melhora o desempenho geral do sistema ao permitir que mais dados e programas sejam mantidos na memória de acesso rápido, mas isso **não acelera diretamente o acesso aos meios magnéticos**, como discos rígidos ou disquetes. Esses dispositivos têm velocidades de leitura e gravação limitadas, independentes da quantidade de RAM instalada. O que pode melhorar o desempenho do sistema é o uso de cache ou técnicas como a memória virtual, que reduz a frequência de acesso a discos

20. **Descreva as funções básicas de uma UCP, indicando os seus principais componentes.**

A **Unidade Central de Processamento (UCP)** tem as seguintes funções básicas:

1. **Controle:** Coordena as operações de todas as partes do computador, determinando quais instruções devem ser executadas e em que ordem.
2. **Execução:** Realiza operações aritméticas e lógicas sobre os dados.
3. **Interpretação de Instruções:** A UCP busca as instruções da memória, as decodifica e as executa.

Principais componentes:

- **Unidade de Controle (UC):** Controla o fluxo de dados e as instruções.
 - **Unidade Aritmética e Lógica (UAL):** Realiza operações matemáticas e lógicas.
 - **Registradores:** Armazenam dados temporários.
 - **Clock:** Sincroniza as operações do processador.
-

21. Quais são as funções da Unidade Aritmética e Lógica – UAL?

A **Unidade Aritmética e Lógica (UAL)** realiza duas funções principais:

1. **Operações Aritméticas:** Soma, subtração, multiplicação, divisão, entre outras operações matemáticas.
2. **Operações Lógicas:** Comparações e operações lógicas como AND, OR, XOR, NOT.

Essas operações são usadas para processar dados conforme as instruções fornecidas ao processador

22. O que é e para que serve o Acc?

O **Acumulador (Acc)** é um registrador especial utilizado para armazenar o resultado das operações aritméticas e lógicas realizadas pela UAL. Ele é fundamental em muitas arquiteturas de processadores, pois facilita o acesso rápido aos resultados intermediários durante o processamento de instruções.

23. Qual é o componente de um processador que determina o período de duração de cada uma de suas atividades e controla o sincronismo entre elas?

O componente responsável por controlar o sincronismo e determinar o período de duração de cada atividade no processador é o **clock**. Ele gera pulsos em intervalos regulares, definindo o tempo que o processador tem para executar cada instrução ou operação

24. Quais são as funções da Unidade de Controle de um processador?

As funções da **Unidade de Controle (UC)** são:

1. **Buscar instruções:** A UC busca as instruções da memória principal.
2. **Decodificar instruções:** A UC interpreta o código de operação da instrução.
3. **Controlar o fluxo de dados:** A UC coordena a transferência de dados entre os componentes internos e externos (registradores, memória, UAL, etc.).
4. **Executar instruções:** A UC ativa os circuitos necessários para a execução de cada instrução.

25. Considere um computador cuja MP é organizada com N células de 1 byte cada uma. As instruções interpretadas pela UCP possuem três tamanhos diferentes: as do tipo A possuem 16 bits; as do tipo B têm 32 bits e as do tipo C possuem 48 bits. Considerando que o código de operação de cada uma tem um tamanho fixo e igual a 8 bits e que os programas executados nesse processador são constituídos de uma mistura dos três tipos de instruções, imagine um processo prático para incremento automático do CI (ou PC) após a execução de cada instrução de um programa.

Para incrementar automaticamente o **Contador de Instruções (CI)** ou **Program Counter (PC)**, o sistema pode verificar o tamanho da instrução após sua execução. Como o código de operação (opcode) é sempre de 8 bits, o PC pode ser incrementado de acordo com o tamanho da instrução:

- **Instruções de 16 bits:** Incrementa o PC em 2 bytes.
- **Instruções de 32 bits:** Incrementa o PC em 4 bytes.
- **Instruções de 48 bits:** Incrementa o PC em 6 bytes.

Essa estratégia garante que o CI sempre aponte para o próximo endereço de instrução correto.

26. Considerando as instruções a seguir, indique a quantidade de ciclos de memória despendidos para realizar o ciclo de instrução completo. Explícite a quantidade de operações de leitura e de escrita, quando foi o caso:

ADD Op	$R0 \leftarrow R0 + (Op)$ (I)
SUB Op	$(Op) \leftarrow R0 - (Op)$ (II)
ADD Op1, Op2	$(Op1) \leftarrow (Op1) + (Op2)$ (III)
INCR	$R0 \leftarrow R0 + 1$ (IV)
LDA Op	$R0 \leftarrow (Op)$ (V)

- **ADD Op:** $R0 \leftarrow R0 + (Op)$
 - **Operações de memória:** 1 leitura (do operando Op da memória).
 - **Total de ciclos de memória: 1 ciclo** de leitura.
- **SUB Op:** $(Op) \leftarrow R0 - (Op)$
 - **Operações de memória:** 1 leitura (do operando Op) e 1 escrita (para gravar o resultado).
 - **Total de ciclos de memória: 2 ciclos** (1 leitura + 1 escrita).
- **ADD Op1, Op2:** $(Op1) \leftarrow (Op1) + (Op2)$
 - **Operações de memória:** 2 leituras (para Op1 e Op2) e 1 escrita (para gravar o resultado em Op1).
 - **Total de ciclos de memória: 3 ciclos** (2 leituras + 1 escrita).
- **INCR:** $R0 \leftarrow R0 + 1$

- **Operações de memória:** Nenhuma, pois a operação é feita diretamente no registrador R0.
 - **Total de ciclos de memória: 0 ciclos** (operação apenas no registrador).
 - **LDA Op:** $R0 \leftarrow (Op)$
 - **Operações de memória:** 1 leitura (para o operando Op da memória).
 - **Total de ciclos de memória: 1 ciclo** de leitura.
-

27. Considere um computador com 64 K células de memória, instruções de um operando, tendo possibilidade de ter um conjunto de 256 instruções de máquina. Considerando que cada instrução tem o tamanho de uma célula, que é o mesmo tamanho da palavra do sistema, qual é o tamanho, em bits, dos registradores CI (ou PC) e RDM? Qual é o tamanho total em bits dessa memória?

- **Tamanho do CI (ou PC): 16 bits.**

O Contador de Instruções (CI ou PC) precisa endereçar todas as células da memória. Como o computador possui **64 K células**, precisamos de $\log_2(64 \times 1024) = 2^6 \times 2^{10} = 16$ bits para o CI.

- **Tamanho do RDM: 8 bits.**

O RDM (Registrador de Dados de Memória) armazena uma palavra de dados. Cada célula de memória armazena uma palavra. Se o sistema é baseado em **256 instruções de máquina**, isso implica que o tamanho da palavra é de **8 bits** (pois $\log_2(256) = 8$).

- **Tamanho total da memória: 524.288 bits (ou 512 Kbits).**

A memória possui **64 K células**, e cada célula armazena **8 bits** (uma palavra). Logo, o tamanho total da memória é:

$$64 K \times 8 \text{ bits} = 64 \times 1024 \times 8 = 524.288 \text{ bits}$$

28. Um computador tem um REM de 16 bits e um barramento de dados de 20 bits. Possui instruções de um operando, todas do tamanho de uma célula e do mesmo tamanho da palavra. Ele foi adquirido com apenas uma placa de MP de 4K. Pergunta-se:

a) Qual é o tamanho, em bits, do RDM e do PC?

- **Tamanho do RDM: 20 bits.**

O barramento de dados tem **20 bits**, então o RDM precisa armazenar **20 bits** para receber ou enviar dados da memória através do barramento.

- **Tamanho do PC: 16 bits.**

O PC deve ser capaz de endereçar a memória. Como o REM tem **16 bits**, o PC também precisa ser capaz de conter esses 16 bits para endereçar a memória.

b) Seria possível aumentar a capacidade de armazenamento dessa memória? Até quanto e por quê?

Sim, seria possível aumentar a capacidade de armazenamento. Atualmente, o sistema possui **4K células**, e o REM tem **16 bits**, o que permite endereçar até $2^{16} = 65.536$ endereços de memória (ou 64 K células). Portanto, a memória pode ser expandida até **64 K células** de memória.

c) Qual é a quantidade máxima de instruções de máquina que poderia existir nesse computador?

O número máximo de instruções de máquina depende do tamanho do campo de código de operação (opcode). Se o opcode ocupa **8 bits** (o valor típico), então o conjunto de instruções pode ter até $2^8 = 256$ instruções de máquina. Portanto, o computador pode ter **até 256 instruções de máquina**.

29. Um computador possui um conjunto de 128 instruções de um operando; supondo que sua memória tenha capacidade de armazenar 512 palavras e que cada instrução tenha o tamanho de uma palavra e da célula de memória. Pergunta-se:

a) Qual o tamanho em bits do REM, RDM, RI e CI?

- **REM (Registrador de Endereçamento de Memória):**

A memória tem **512 palavras**. O REM deve ser capaz de endereçar todas essas palavras. O número de bits necessários para isso é:

$$\log_2(512) = 9 \text{ bits}$$

- **RDM (Registrador de Dados de Memória):**

O RDM deve armazenar o conteúdo de uma palavra. Se o sistema tem **128 instruções**, o tamanho do RDM pode ser de **8 bits** (pelo menos), o suficiente para armazenar uma palavra ou o código da instrução.

- **RI (Registrador de Instruções):**

O RI precisa armazenar o código da instrução. Como o conjunto de instruções tem **128 instruções**, o RI precisa de $\log_2(128) = 7$ bits para armazenar o código de operação.

- **CI (Contador de Instruções):**

O CI precisa endereçar todas as palavras da memória, então, como no caso do REM, ele deve ter **9 bits**.

b) Qual a capacidade de memória em bytes?

Cada palavra armazena ****8 bits**** (ou 1 byte), e a memória tem ****512 palavras****. Portanto, a capacidade total de memória é:

$$512 \text{ palavras} \times 1 \text{ byte por palavra} = 512 \text{ bytes.}$$

c) Se se quisesse alterar o tamanho das instruções para 17 bits, mantendo inalterado o tamanho do REM, quantas novas instruções poderiam ser criadas?

Se o tamanho das instruções fosse alterado para **17 bits** e o tamanho do REM permanecesse o mesmo, o campo do opcode (código de operação) teria:

$$17 \text{ bits de instrução} - 9 \text{ bits para o endereço do operando} = 8 \text{ bits para o opcode.}$$

Com **8 bits disponíveis para o opcode**, seria possível ter até $2^8 = 256$ instruções. Portanto, **128 novas instruções poderiam ser criadas**, além das 128 já existentes, totalizando **256 instruções**.

30. Considere um computador que possua uma UCP com CI de 16 bits e RI de 38 bits. Suas instruções têm dois operandos do mesmo tamanho (16 bits), além, é claro, de um código de operação. Pergunta-se:

a) Qual o tamanho da instrução?

Cada instrução tem dois operandos de **16 bits** e um código de operação. O RI (Registrador de Instruções) tem **38 bits**, então o código de operação pode ser calculado como:

$$38 \text{ bits} - (2 \times 16 \text{ bits dos operandos}) = 6 \text{ bits para o opcode.}$$

Portanto, o tamanho total da instrução é:

$$16 \text{ bits (operando 1)} + 16 \text{ bits (operando 2)} + 6 \text{ bits (opcode)} = 38 \text{ bits.}$$

b) Qual o tamanho do campo do código da operação?

Conforme calculado acima, o campo do código de operação (opcode) ocupa **6 bits**.

c) Considerando que a configuração básica dessa máquina é de 16 Kbytes de memória, até que tamanho pode a memória ser expandida?

Para expandir a memória, precisamos analisar o tamanho do CI (Contador de Instruções), que tem **16 bits**. Um CI de 16 bits pode endereçar até $2^{16} = 65.536$ endereços, ou seja, $2^{16} = 65.536$ endereços. Se cada endereço corresponde a 1 byte, a memória pode ser expandida até **65.536 bytes**, ou seja, **64 Kbytes**.

- **Tamanho máximo da memória: 64 Kbytes.**