

2. ¿Cuántos bits se necesitan para direccionar 1.048.576 palabras?

Para ello si usamos una calculadora común utilizamos la siguiente formula  
 $\log(\text{numero de palabras/ubicaciones}) / \log(2) = \text{cantidad de bits que necesitaremos}$

Resultado por chatgpt:

**Para ello si usamos una calculadora común utilizamos la siguiente formula**  
 **$\log(\text{numero de palabras/ubicaciones}) / \log(2) = \text{cantidad de bits que necesitaremos}$**   
**Paso a paso:**

1. Cuando hablamos de **direccionar memoria**, nos referimos a **cuántas direcciones diferentes** necesitamos para acceder a cada posición.
2. **1.048.576 palabras** significa que hay **1.048.576 ubicaciones distintas** en memoria.
3. Para saber cuántos **bits** necesito para representar esa cantidad de ubicaciones, uso la fórmula:

**Cantidad de bits =  $\log_2(\text{número de ubicaciones})$**

Porque con  $n$  bits puedo representar hasta  $2^n$  ubicaciones.

4. Buscamos el exponente tal que:

$$2^x = 1.048.576$$

$$x = \log_2(1.048.576)$$

$$2^{20} = 1.048.576$$

✓ **Resultado:** Se necesitan **20 bits** para direccionar 1.048.576 palabras

Fórmulas para calcular  $\log_2(x)$ :

**Opción 1: usando logaritmo decimal (base 10):**

$$\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)} \quad \log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$$

**Opción 2: usando logaritmo natural (ln):**

$$\log_2(x) = \frac{\ln(x)}{\ln(2)} \quad \log_2(x) = \frac{\ln(x)}{\ln(2)}$$

---

Para ahcer en calculadora:

Querés calcular:

$$\log_2(1.048.576) \quad \log_2(1.048.576)$$

**Paso 1: En la calculadora, hacé:**

- $\log(1048576) \approx 6.0206$
- $\log(2) \approx 0.3010$

**Paso 2:**

$$6.0206 \cdot 0.3010 \approx 20 \cdot \frac{6.0206}{0.3010} \approx 200.30106.0206 \approx 20$$

✓ Resultado:

$$\log_2(1.048.576) = 20 \cdot \log_2(1.048.576) = 20 \log_2(1.048.576) = 20$$

3)

3. ¿Cuántos bits para direccionar 8 GiB de memoria?

**Paso a paso:**

1. **1 GiB (gibibyte)** =  $2^{30}$  bytes  
→ Entonces, **8 GiB** =  $8 \times 2^{30} = 2^{33}$  bytes
2. Para direccionar cada **byte individualmente**, necesito una dirección única para cada uno.
3. Entonces aplico la misma lógica que en el punto 2:

¿Cuántos bits necesito para direccionar  $2^{33}$  bytes?

→ Respuesta: **33 bits**

✓ Resultado: Se necesitan **33 bits** para direccionar 8 GiB.

4. Análisis con memoria de 1GB, palabra de 64 bits, etc.

**Primero: convirtamos todo bien**

- **Memoria total:** 1 GB =  $2^{30}$  bytes
- **Palabra:** 64 bits = 8 bytes  
→ Total de **palabras:**  $2^{30} / 8 = 2^{27}$  palabras
- **Registro de instrucción:** 32 bits, donde el **campo de direcciones** es de **16 bits**

a. ¿Cuánta memoria se direcciona con direccionamiento directo?

- En direccionamiento directo, el campo de la instrucción tiene una **dirección directa de memoria**.
- Si ese campo tiene **16 bits**, entonces puede representar hasta  $2^{16} = 65.536$  direcciones.

- Como cada dirección es una **palabra** (de 8 bytes), entonces:

$$65.536 \text{ palabras} \times 8 \text{ bytes} = \mathbf{524.288 \text{ bytes} = 512 \text{ KiB}}$$

✓ **Resultado:** Se puede direccionar **512 KiB de memoria**.

---

#### b. ¿Cuánta memoria con direccionamiento indirecto?

- La dirección que aparece en la instrucción **no es la definitiva**, sino que **en la memoria** hay otra dirección.
- Esto **no limita** el tamaño de memoria directamente.

→ En teoría, se puede acceder a **toda la memoria disponible**, porque podés encadenar direcciones.

✓ **Resultado:** Se puede direccionar **1 GB completo**.

---

#### c. ¿Cuánta memoria con direccionamiento inmediato?

- En este caso, el campo de la instrucción contiene el **dato directamente**, no es una dirección de memoria.
- Si el campo inmediato tiene 16 bits, puede almacenar un valor entre 0 y  $2^{16} - 1 = \mathbf{65.535}$

✓ **Resultado:** No se direcciona memoria, pero se puede representar valores **hasta 65.535**

---

#### d. Direccionamiento por base y desplazamiento, con base de 24 bits

- Dirección = **registro base (24 bits) + desplazamiento (16 bits)**
- La suma total cubre un rango de:
  - Base: hasta  $2^{24} = 16.777.216$
  - Desplazamiento: hasta  $2^{16} = 65.536$
  - Total máximo accesible:  $\mathbf{16.777.216 + 65.536 = 16.842.752 \text{ palabras}}$

$$\rightarrow 16.842.752 \times 8 \text{ bytes} = \mathbf{134.742.016 \text{ bytes} \approx 128.5 \text{ MiB}}$$

✓ **Resultado:** Se puede direccionar **hasta ~128.5 MiB**

---

#### e. ¿Qué tamaño debe tener el contador de programa (PC)?

- El **contador de programa** apunta a la próxima instrucción.
- Si quiero que pueda **recorrer TODA la memoria como código**, debo saber cuántas instrucciones hay.
- La memoria tiene  $2^{30}$  bytes.
- Cada instrucción ocupa 4 bytes (porque el registro de instrucción es de 32 bits)

→ Total de instrucciones posibles =  $2^{30} / 2^2 = 2^{28}$  instrucciones

→ Necesito **28 bits** para contar hasta  $2^{28}$

✓ **Resultado:** El PC debe tener **28 bits**

a tener en cuenta:

Unidad	Valor Real en Bytes	Notación Binaria
1 KB	1.000 bytes	✗ No se usa así en RAM
1 KiB	1.024 bytes = $2^{10}$	✓
1 MB	1.000.000 bytes	✗
1 MiB	1.048.576 bytes = $2^{20}$	✓
1 GB	1.000.000.000 bytes	✗
1 GiB	1.073.741.824 bytes = $2^{30}$	✓

📄 GB (Gigabyte) – Sistema decimal:

- Se usa en marketing y sistemas de almacenamiento (como discos duros, SSD, pendrives).
- **1 GB = 1.000.000.000 bytes =  $10^9$  bytes**
- Es una unidad basada en **múltiplos de 10**

📄 GiB (Gibibyte) – Sistema binario:

- Se usa en informática "real", sobre todo en **RAM, arquitectura de computadoras y sistemas operativos**.
- **1 GiB = 1.073.741.824 bytes =  $2^{30}$  bytes**
- Es una unidad basada en **potencias de 2**