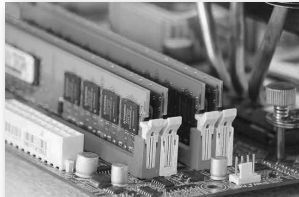




## VJ1214 - Consolas y dispositivos

### Bloque 1: Arquitectura de computadores

#### Construcción de memorias



## Índice de contenidos

1. Lógica combinacional vs lógica secuencial
2. El reloj
3. Flip-Flop
4. Registro de 1 bit
5. Registro de N bits
6. Memoria RAM
7. Memorias RAM a partir de otras memorias RAM.

## Libro de referencia



- Capítulo 3



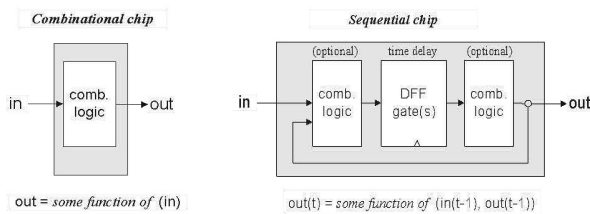
- Capítulos 7, 8, 9 y 10

### 1. Lógica combinacional vs lógica secuencial

### 1. Lógica combinacional vs lógica secuencial

- Lógica combinacional
  - El resultado depende únicamente de una **combinación** de las entradas.
  - No permite mantener el estado.
- Lógica secuencial
  - El resultado depende de una combinación de las entradas y del **estado** anterior.
  - Permite mantener el estado durante el tiempo.

## 1. Lógica combinacional vs lógica secuencial



## 2. El reloj

- Todos los circuitos secuenciales son alimentados con el **reloj** del sistema

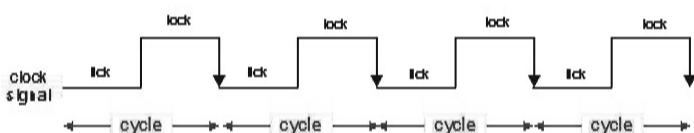


## 2. El reloj

## 3. Flip-Flop

## 2. El reloj

- El acto de recordar algo es inherentemente dependiente del **tiempo**.
- El paso del tiempo en un computador es posible gracias al reloj del sistema.
- Proporciona pulsos continuos



## 3. Flip Flop

- El elemento básico de la lógica secuencial es el Flip-Flop
- DFF: Data flip-flop
- **Todas** las memorias y circuitos secuenciales necesarios se construyen a partir de DFF.

### 3. Flip Flop

- $Out(t) = in(t-1)$



### 3. Flip Flop

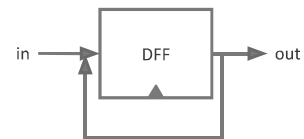
- En este curso **no nos preocupamos** de cómo están **implementados** los DFF internamente.
- Partiremos de DFF ya implementados.

### 4. Registro de 1 bit

- Un registro es un dispositivo de memoria que puede guardar (o recordar) un valor a lo largo del tiempo.
- $out(t) = out(t-1)$

### 4. Registro de 1 bit

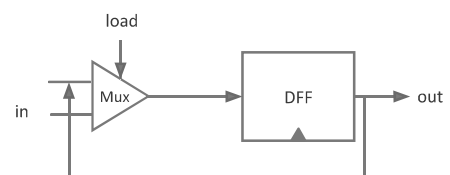
- Diseño **incorrecto**



- Guarda siempre el mismo elemento
- No permite introducir un **nuevo** elemento para ser recordado.

### 4. Registro de 1 bit

- Diseño **correcto**

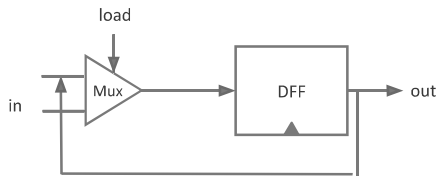


- Si  $load(t-1)$   
     $out(t) = in(t-1)$   
Si no  
     $out(t) = out(t-1)$

### 4. Registro de 1 bit

## 4. Registro de 1 bit

- Diseño **correcto**

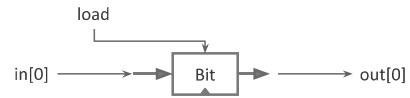


Si  $\text{load}(t-1)$   
 $\text{out}(t) = \text{in}(t-1)$   
 Si no  
 $\text{out}(t) = \text{out}(t-1)$

- Load = 0, si queremos mantener el valor actual.
- Load = 1, si queremos que el registro guarde un nuevo valor.

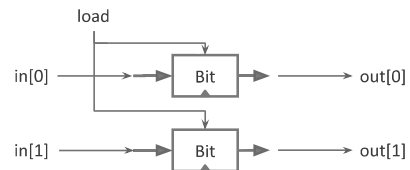
## 5. Registro de N bit

- Ejemplo con N=4



## 5. Registro de N bit

- Ejemplo con N=4



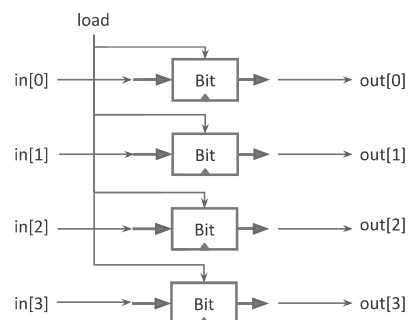
## 5. Registro de N bits

## 5. Registro de N bit

- Usaremos N registros de 1 bit, todos conectados a la misma entrada *clock* y al misma entrada *load*.
- Si  $\text{load}=1$ , guardamos el nuevo valor de N bits
- Si  $\text{load}=0$ , mantenemos el valor existente

## 5. Registro de N bit

- Ejemplo con N=4



## 6. Memoria RAM

## 6. Memoria RAM

- Ejemplo RAM de **8** registros de 16 bits
- Dirección es de **3 bits**

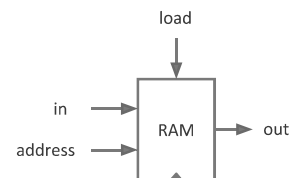
Dirección	Valor almacenado
000	0000 1111 0001 1111
001	1111 0000 1111 0000
010	1111 0000 1010 0000
011	0000 0000 1111 0001
100	1111 0000 1010 0000
101	0000 1111 0001 1111
110	1111 0000 1111 0000
111	0000 0000 1111 0001

## 6. Memoria RAM

- Una memoria RAM es una colección de N registros donde se puede especificar qué registro se desea leer o escribir.
- Entradas:
  - in: valor a almacenar
  - load: 1 si se desea escribir, 0 si se desea leer
  - address: número de registro a leer/escribir
- Salidas:
  - out: valor leído.

## 6. Memoria RAM

- Lectura de un registro
  - Load = 0
  - out devuelve el registro indicado en address
  - in no se usa



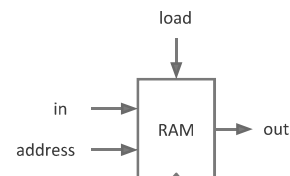
## 6. Memoria RAM

- Ejemplo RAM de 4 registros de 16 bits
- Dirección es de **2 bits**
  - 00 primer registro
  - 01 segundo registro
  - 10 tercer registro
  - 11 cuarto registro

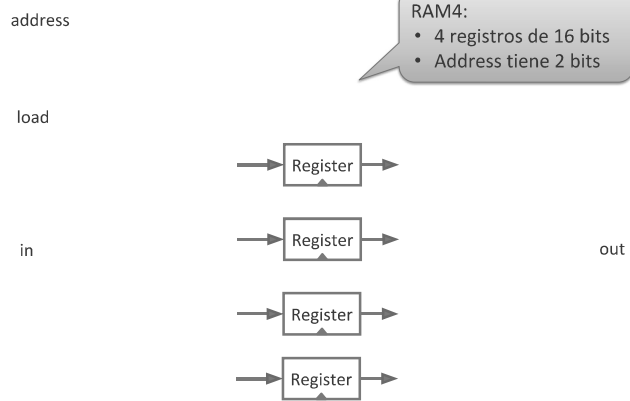
Dirección	Valor almacenado
00	0000 1111 0001 1111
01	1111 0000 1111 0000
10	1111 0000 1010 0000
11	0000 0000 1111 0001

## 6. Memoria RAM

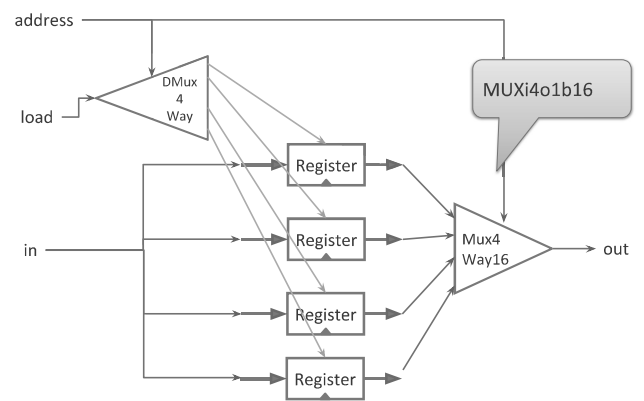
- Escritura en un registro
  - Load = 1
  - out no se usa
  - El contenido de in se almacena en el registro indicado por address



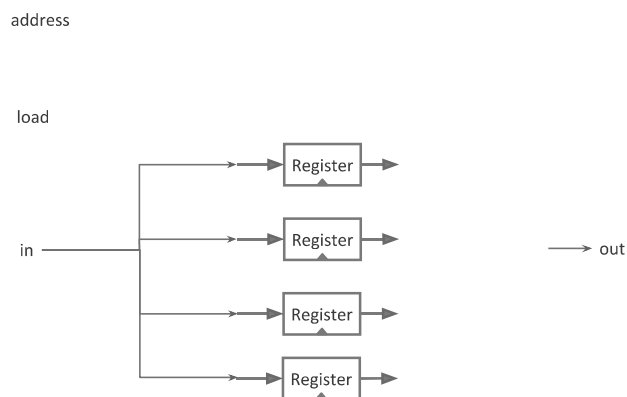
## 6. Memoria RAM



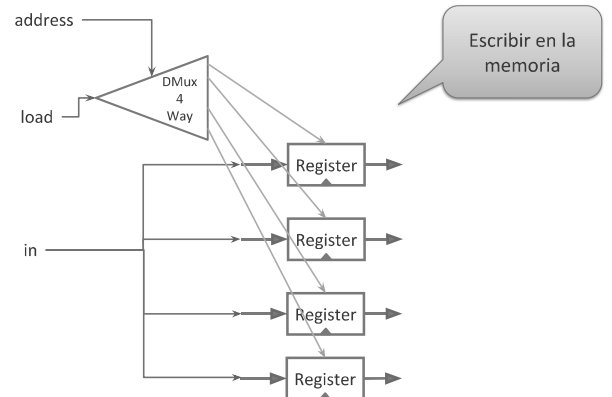
## 6. Memoria RAM



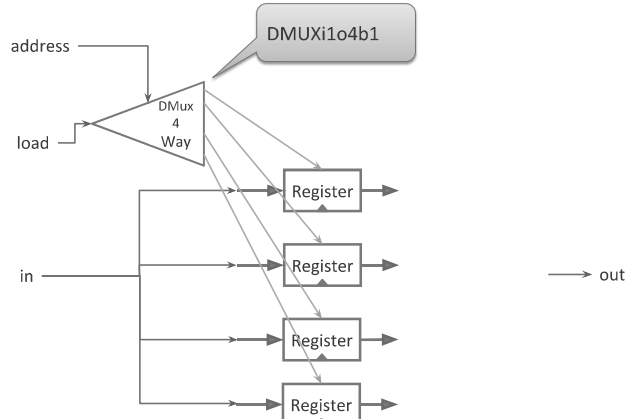
## 6. Memoria RAM



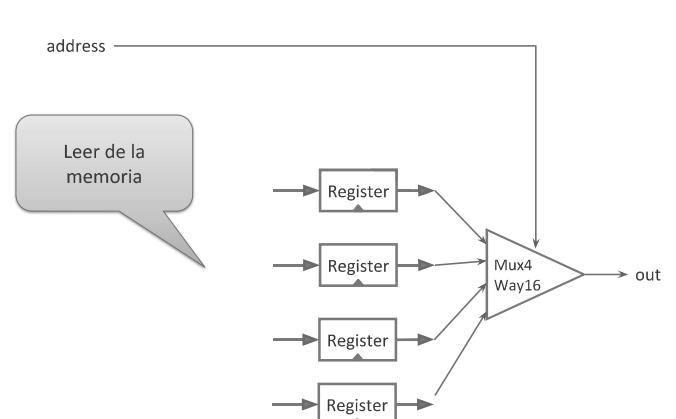
## 6. Memoria RAM



## 6. Memoria RAM



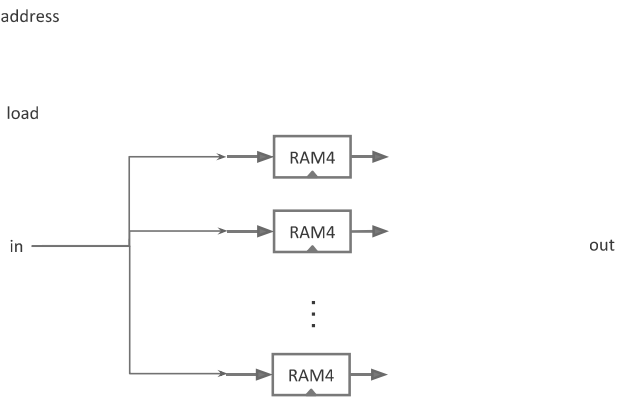
## 6. Memoria RAM



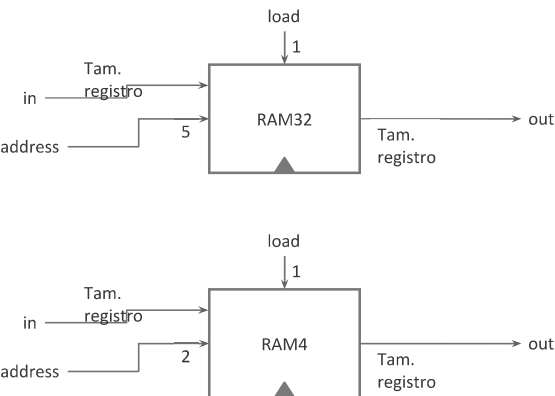
## 7. Memoria RAM a partir de otras

- Memoria RAM capaz de almacenar hasta 32 registros (RAM32), a partir de memorias RAM4:
  - Serán necesarias 8 memorias RAM4 para construir una RAM32 ( $32 = 8 \cdot 4$ ).
  - La entrada *address* tendrá 5 bits ( $2^5 = 32$ ).
    - Los tres bits más significativos [4, 3, 2] nos servirán para seleccionar la RAM4 que corresponda.
    - Los dos bits menos significativos [1, 0] nos servirán para seleccionar el registro que corresponda dentro de la RAM4 seleccionada.

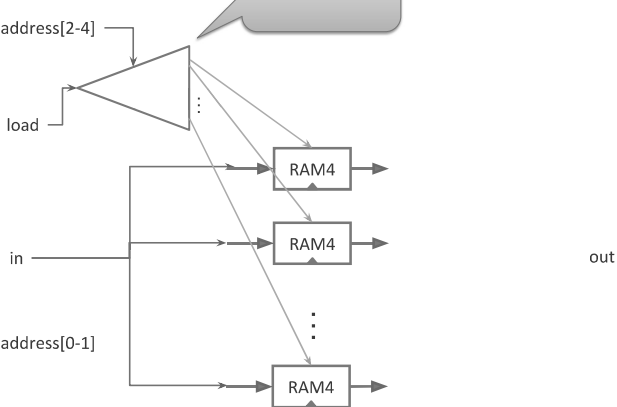
## 7. Memoria RAM a partir de otras



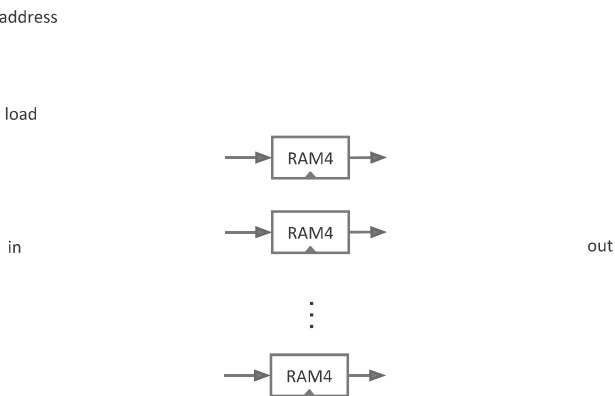
## 7. Memoria RAM a partir de otras



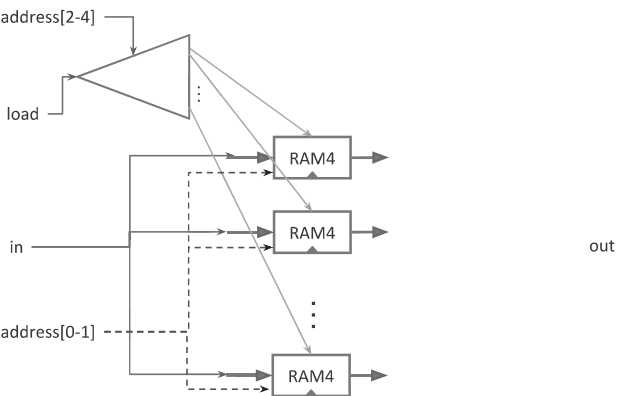
## 7. Memoria RAM a partir de otras



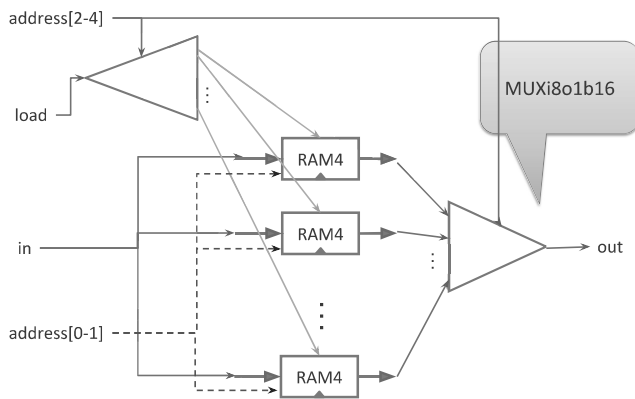
## 7. Memoria RAM a partir de otras



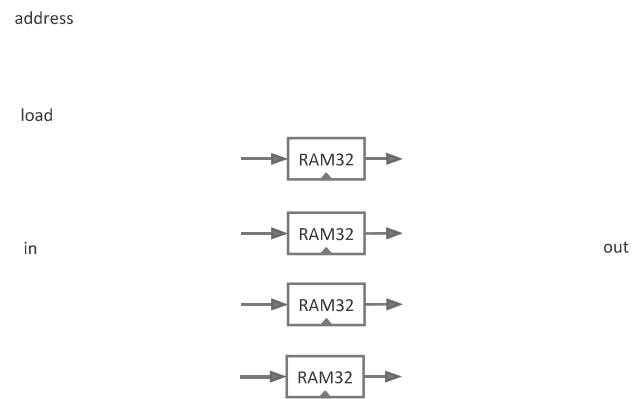
## 7. Memoria RAM a partir de otras



## 7. Memoria RAM a partir de otras



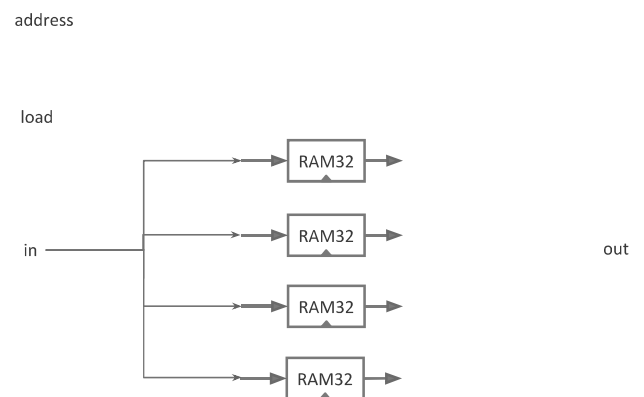
## 7. Memoria RAM a partir de otras



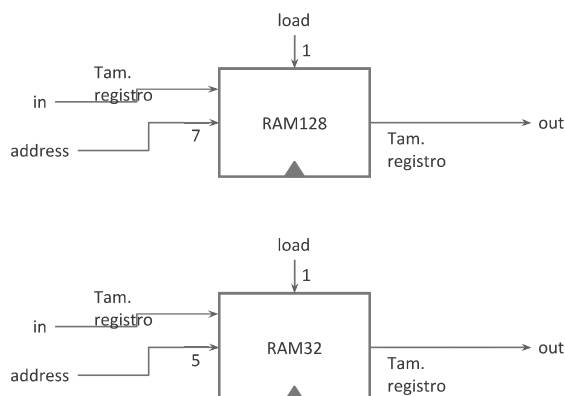
## 7. Memoria RAM a partir de otras

- Memoria RAM capaz de almacenar hasta 128 registros (RAM128), a partir de memorias RAM32:
  - Serán necesarias 4 memorias RAM32 para construir una RAM128 ( $128 = 4 \cdot 32$ ).
  - La entrada *address* tendrá 7 bits ( $2^7 = 128$ ).
    - Los 2 bits más significativos [6, 4] nos servirán para seleccionar la RAM32 que corresponda.
    - Los bits [4, 3, 2] nos servirán para seleccionar la RAM4 que corresponda dentro de la RAM32 seleccionada.
    - Los dos bits menos significativos [1, 0] nos servirán para seleccionar el registro que corresponda dentro de la RAM4 seleccionada.

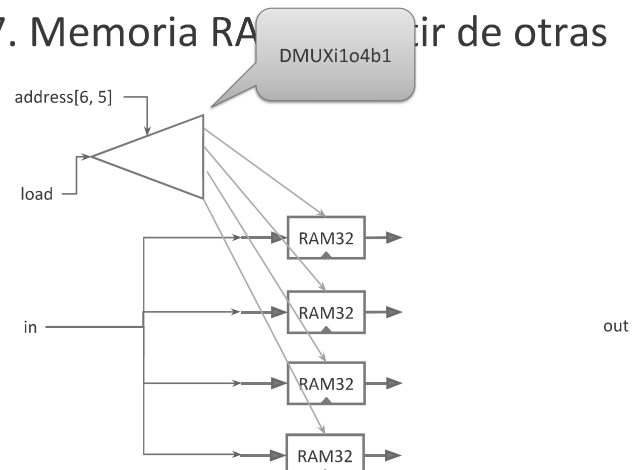
## 7. Memoria RAM a partir de otras



## 7. Memoria RAM a partir de otras

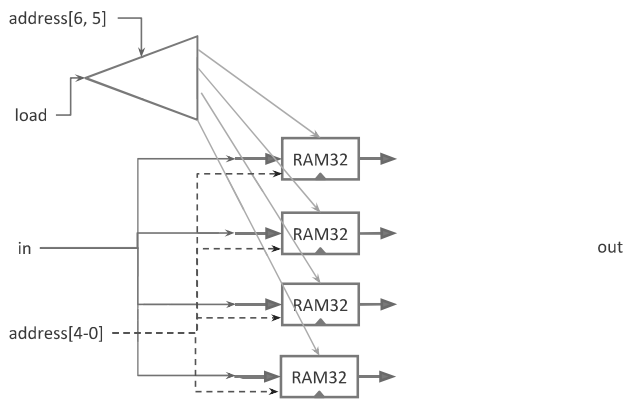


## 7. Memoria RAM





## 7. Memoria RAM a partir de otras



## Circuitos para proyecto 03

Bit

Register (16 bits)

RAM8 (8 registros de 16 bits)

RAM64 (8 memorias RAM8)

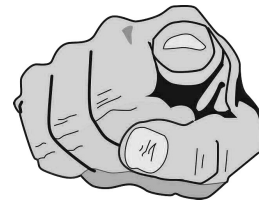
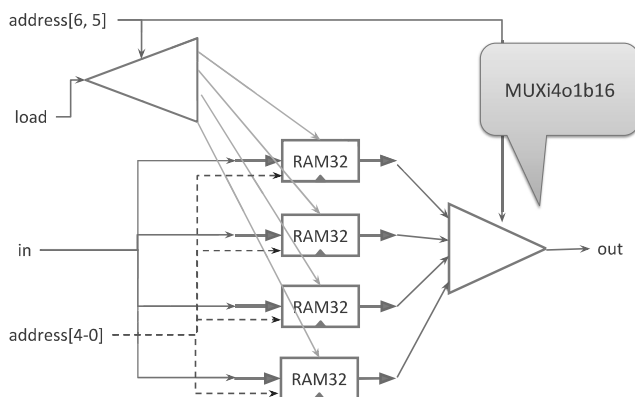
RAM512 (8 memorias RAM64)

RAM4K (8 memorias RAM512)

RAM16K (4 memorias RAM4K)

PC

## 7. Memoria RAM a partir de otras



## Circuitos para proyecto 03

## Créditos imágenes utilizadas

Bit

Register (16 bits)

RAM8

RAM64

RAM512

RAM4K

RAM16K

PC

- <https://pixabay.com/es/chips-de-memoria-memoria-ram-57269/>
- <https://pixabay.com/es/reloj-analogico-tiempo-ver-147257/>

