

**Tema:**

**“MAPAS DE MEMORIA:  
LÓGICA DE SELECCIÓN,  
GESTIÓN Y ORDENACIÓN DE  
LA MEMORIA”**

# MAPA DE MEMORIA

## Mapa de memoria

- ◆ Memoria que es capaz de direccionar un microprocesador.
- ◆ Distribución de la misma, es decir que direcciones ocupan los diferentes dispositivos destinados a funciones determinadas.
  - ◆ También se puede incluir en el mapa los periféricos con los que trabajará el microprocesador

### La especificación del mapa de memoria se puede realizar como:

**Funcional:** ubicación (direcciones) de los elementos (hardware o software) del sistema digital, atendiendo a la función de los mismos. Así se describirán la ubicación de: sectores de programa, posición de datos generales y tablas, registros de interfaz, etc.

**Físico:** correspondencia entre las direcciones del mapa y el dispositivo físico en el que se plasman. De acuerdo a él se realizará la conexión entre los diferentes dispositivos, teniendo en cuenta la estructura del bus de direcciones y del bus de datos, la forma de selección de dispositivos, etc.

# MAPA DE MEMORIA

| Mapa de Memoria  |   |                  |
|--|---|------------------|
| Funcional  | Físico  | Direcciones      |
| Programa y tablas de datos fijos.                      | Pastilla ROM de 8Kbytes.<br>Circuito Integrado tipo.... | 0x0000<br>0x1FFF |
| <i>Zona vacía (Ningún chip ocupa estas posiciones)</i> | ---   | 0x2000<br>0x3FFF |
| Dispositivos de Entrada/Salida                         | Registros varios:                                       | 0x4000<br>0x403F |
| <i>Zona vacía</i>                                      |   | 0x4400<br>0x8FFF |
| Teclado hexadecimal y registros auxiliares.            | modelo ....   | 0x9000<br>0x9003 |
| <i>Zona vacía</i>                                      | ---   | 0x9004<br>0xBFFF |
| Variables y tablas temporales(4Kbytes)                 | Pastilla RAM de 8Kbytes                                 | 0xC000<br>0xCFFF |
| Zona de datos transferencia serie (4Kbytes).           | C.I. tipo....   | 0xD000<br>0xDFFF |

DIRECCIÓN  
BASE

# MAPA DE MEMORIA: Lógica de selección

## Tareas fundamentales:

- ◆ **Diseño del mapa funcional**, donde se decide en que direcciones del mapa se van a colocar las diferentes partes y variables del sistema.
  
- ◆ **Diseño del mapa lógico** donde se seleccionan los dispositivos más adecuados.
  
- ◆ **Diseño de la lógica de selección** para acceder a la celda o posición de memoria deseada y no a otras. Activar los elementos que intervienen en la operación.
  - A partir de una dirección que aparece en el bus, activa la línea de selección del chip al que corresponde dicha dirección.
  
  - Puede diseñarse empleando puertas lógicas, decodificadores, etc, o empleando dispositivos programables como PLD, PAL, etc.

# MAPA DE MEMORIA: Lógica de selección

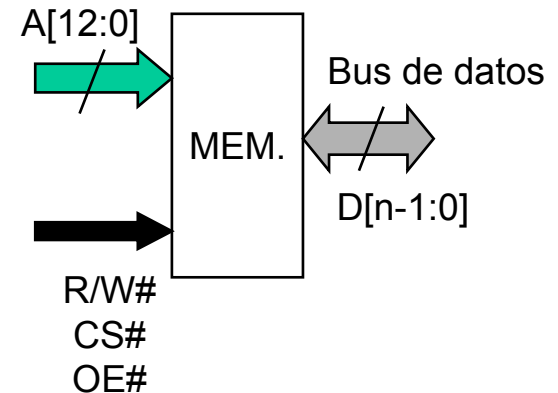
## ◆ Proceso

Tener en cuenta la **dirección base**

Reconocer la función de los diferentes bits de direcciones.

P.e. para el elemento (pastilla RAM de 8 Kbytes) las líneas A[12..0] se necesitan para seleccionar la posición deseada dentro del chip, y por tanto pueden presentar cualquier código, lo que se representa como "XX...X".

El resto de las líneas A[15..13], para que la dirección presente en el bus se corresponda a este chip, deben presentar el código "1 1 0", *C000 h a DFFF h*



**Si la dirección base no termina en tantos "...000" como líneas de direcciones tiene el chip. !!!!!**

# MAPA DE MEMORIA **Lógica de selección**

| <i>Lineas de Direcciones</i> |            |            |            |            |            |           |           |                  |                            |            |            |            |            |  |
|------------------------------|------------|------------|------------|------------|------------|-----------|-----------|------------------|----------------------------|------------|------------|------------|------------|--|
| <i>Byte Alto</i>             |            |            |            |            |            |           |           | <i>Byte Bajo</i> | <i>Lineas de Selección</i> |            |            |            |            |  |
| <i>A15</i>                   | <i>A14</i> | <i>A13</i> | <i>A12</i> | <i>A11</i> | <i>A10</i> | <i>A9</i> | <i>A8</i> | <i>A7-A0</i>     | <i>CS1</i>                 | <i>CS2</i> | <i>CS3</i> | <i>ES1</i> | <i>ES2</i> | <i>Dispositivo seleccionado</i>        |
| <i>0</i>                     | <i>0</i>   | <i>0</i>   | <i>X</i>   | <i>X</i>   | <i>X</i>   | <i>X</i>  | <i>X</i>  | <i>XX</i>        | <i>0</i>                   | <i>1</i>   | <i>1</i>   | <i>1</i>   | <i>1</i>   | <i>Selecciona el C.I. nº 1</i>         |
| <i>1</i>                     | <i>1</i>   | <i>0</i>   | <i>0</i>   | <i>X</i>   | <i>X</i>   | <i>X</i>  | <i>X</i>  | <i>XX</i>        | <i>1</i>                   | <i>0</i>   | <i>1</i>   | <i>1</i>   | <i>1</i>   | <i>Selecciona el C.I. nº 2</i>         |
| <i>1</i>                     | <i>1</i>   | <i>0</i>   | <i>1</i>   | <i>X</i>   | <i>X</i>   | <i>X</i>  | <i>X</i>  | <i>XX</i>        |                            |            |            |            |            |  |
| <i>0</i>                     | <i>1</i>   | <i>0</i>   | <i>0</i>   | <i>0</i>   | <i>0</i>   | <i>X</i>  | <i>X</i>  | <i>XX</i>        | <i>1</i>                   | <i>1</i>   | <i>1</i>   | <i>0</i>   | <i>1</i>   | <i>Activa Entrada/Salida 1</i>         |
| <i>1</i>                     | <i>0</i>   | <i>1</i>   | <i>0</i>   | <i>0</i>   | <i>0</i>   | <i>0</i>  | <i>0</i>  | <i>0000 00XX</i> | <i>1</i>                   | <i>1</i>   | <i>1</i>   | <i>1</i>   | <i>0</i>   | <i>Activa Entrada/Salida (teclado)</i> |

# MAPA DE MEMORIA **Lógica de selección**

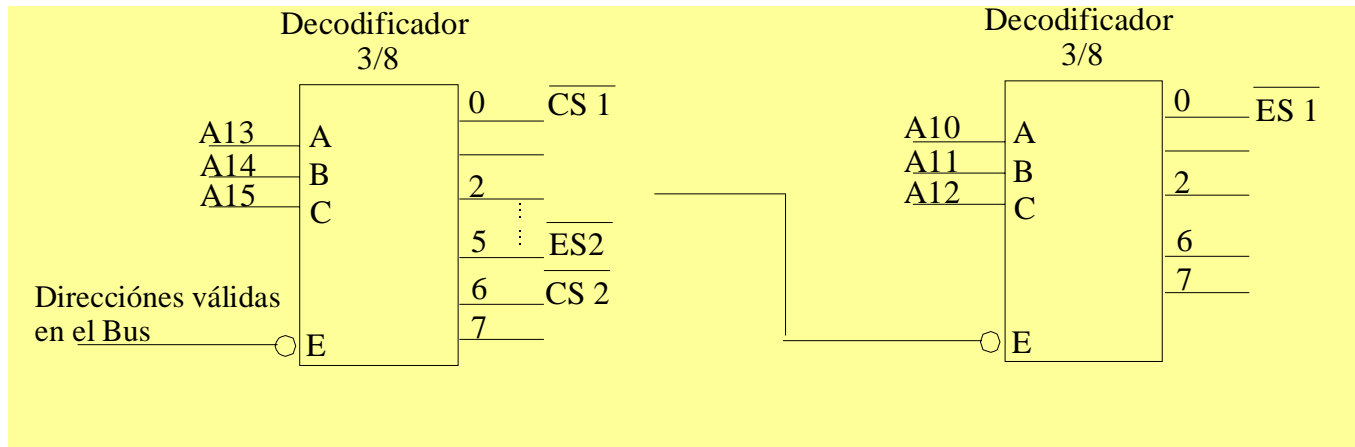
## **Decodificación completa:**

Se emplea este término cuando el acceso a una posición concreta de memoria se puede realizar sólo para una única combinación de los bits del bus de direcciones; esto es: ***{una posición física = una dirección lógica}***.

## **Decodificación incompleta:**

Se corresponde a aquella situación en la que el acceso a una posición concreta de memoria se puede producir con varias combinaciones diferentes de los bits del bus de direcciones: ***{una posición física = varias direcciones lógicas}***.

# MAPA DE MEMORIA Lógica de selección

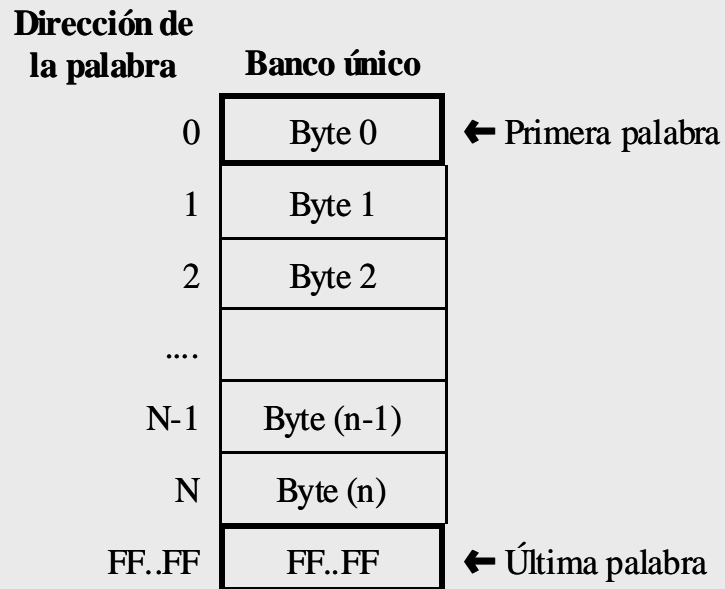




# GESTIÓN DE LA MEMORIA

◆ En función del **número de bits del bus de datos externo** del  $\mu$ P, la estructura de la memoria puede ser de diversos modos.

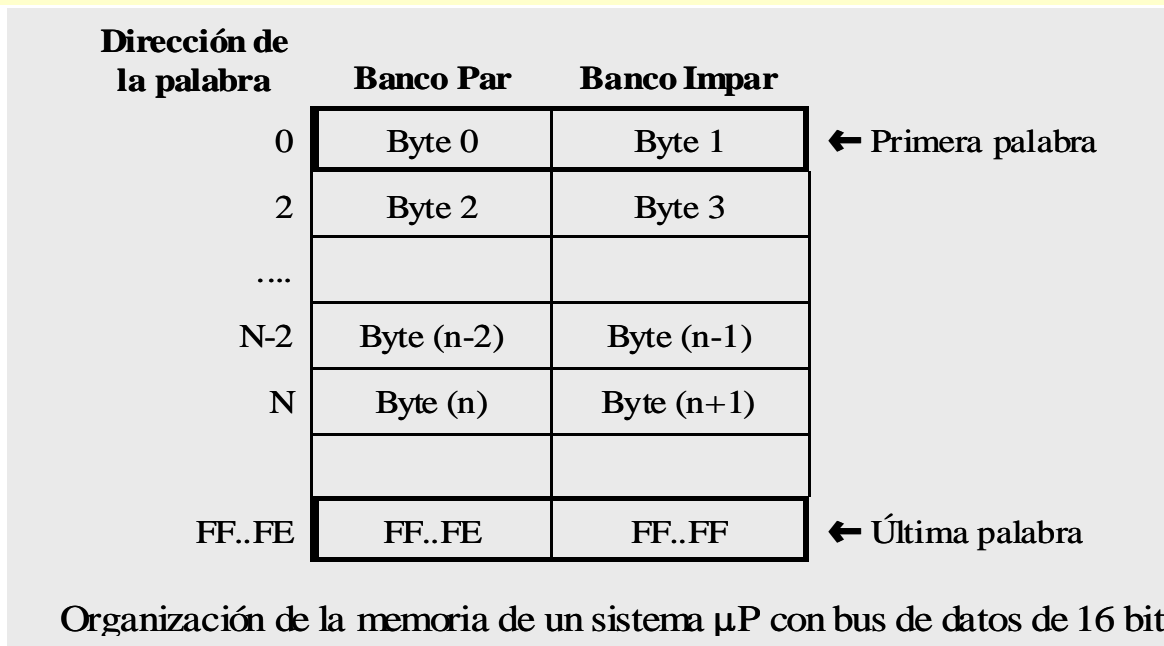
- bus de datos de 8 bits, único banco (grupo) de bytes..



Organización en bytes de la memoria de un sistema con  $\mu$ P.

# GESTIÓN DE LA MEMORIA

- Bus de datos externo de **16 bits** (aunque internamente lo posea de 32), cuando va a leer o escribir a la memoria tiene la posibilidad de hacerlo de 16 en 16 bits. Entonces, **la memoria se organiza en dos bancos: el par y el impar.**
  - Se debe permitir acceder a datos de uno y dos bytes



# GESTIÓN DE LA MEMORIA

- Bus de datos externo de **32 bits**, la memoria se estructura en **4 bancos**.
  - se deben permitir los accesos a uno, dos y cuatro bytes.

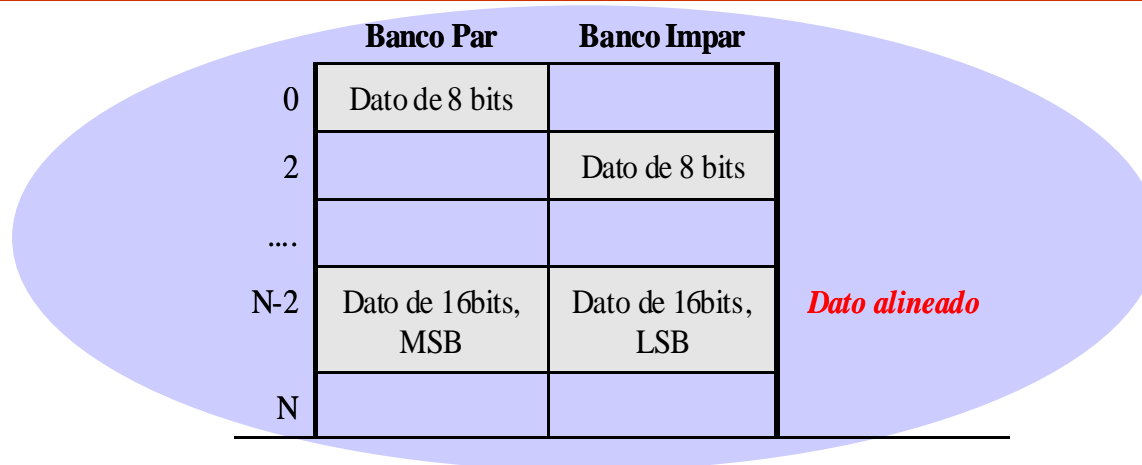
| Dirección de la doble palabra | Banco 0                    | Banco 1    | Banco 2          | Banco 3    |                 |
|-------------------------------|----------------------------|------------|------------------|------------|-----------------|
| 0                             | Byte 0                     | Byte 1     | Byte 2           | Byte 3     | ← Primera palab |
| 4                             | Palabra (Word) 4           |            | Palabra (Word) 6 |            |                 |
| 8                             | Doble Palabra (Longword) 8 |            |                  |            |                 |
| ...                           | ....                       | ....       | ....             | ....       |                 |
| N-4                           | Byte (n-4)                 | Byte (n-3) | Byte (n-2)       | Byte (n-1) |                 |
| N                             | Palabra (N)                |            | Palabra (N+2)    |            |                 |
| N+4                           | Doble Palabra (N+4)        |            |                  |            |                 |
| ...                           | ....                       | ....       | ....             | ....       |                 |
| FF..F8                        | FF..F8                     | FF..F9     | FF..FA           | FF..FB     |                 |
| FF..FC                        | FF..FC                     | FF..FD     | FF..FE           | FF..FF     | ← Última palab  |

# ORDENACIÓN DE LA MEMORIA

- Queda por determinar de los varios bytes que componen el dato cuál representa el de mayor peso, los de pesos intermedios y el de menor peso

**Big endian.** Este tipo de ordenación hace corresponder el **byte de menor peso** del dato con el almacenado en la celda de memoria de **dirección más alta**, dentro de las que contienen el dato.

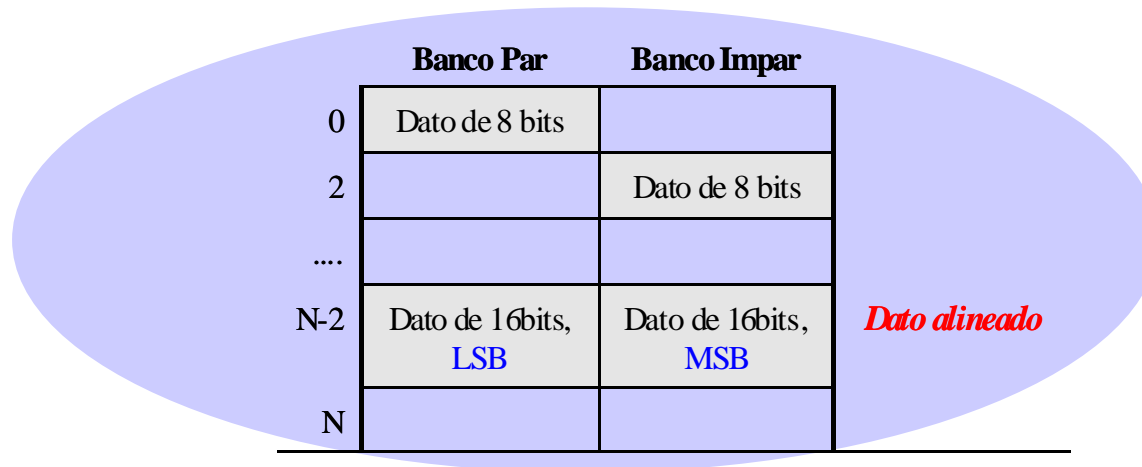
**En cualquier caso las celdas de memoria donde se encuentra almacenado el dato han de ser consecutivas.**



# ORDENACIÓN DE LA MEMORIA

**Little endian.** Este tipo de ordenación hace corresponder el **byte de menor peso** del dato con el almacenado en la celda de memoria de **dirección más baja**, dentro de las que contienen el dato.

**En cualquier caso las celdas de memoria donde se encuentra almacenado el dato han de ser consecutivas.**



# MAPA DE MEMORIA

|     | Banco 0                | Banco 1                | Banco 2                | Banco 3                |                       |
|-----|------------------------|------------------------|------------------------|------------------------|-----------------------|
| 0   | Dato de 8 bits         |                        |                        | Dato de 8 bits         |                       |
| 2   |                        | Dato de 8 bits         | Dato de 8 bits         |                        |                       |
| ... |                        |                        |                        |                        |                       |
| N-2 | Dato de 16bits,<br>LSB | Dato de 16bits,<br>MSB | Dato de 16bits,<br>LSB | Dato de 16bits,<br>MSB | Datos alineados       |
| N   |                        |                        |                        |                        |                       |
| N+2 |                        | Dato de 16bits,<br>LSB | Dato de 16bits,<br>MSB | Dato de 16bits,<br>LSB | Datos no<br>alineados |
|     | Dato de 16bits,<br>MSB |                        |                        |                        |                       |
|     |                        |                        |                        |                        |                       |
|     | Dato de 32bits,<br>LSB | Dato de 32bits         | Dato de 32bits         | Dato de 32bits,<br>MSB | Datos alineados       |
|     |                        | Dato de 32bits,<br>LSB | Dato de 32bits         | Dato de 32bits         | Datos no<br>alineados |
|     | Dato de 32bits,<br>MSB |                        | Dato de 32bits,<br>LSB | Dato de 32bits         |                       |
|     | Dato de 32bits         | Dato de 32bits,<br>MSB |                        | Dato de 32bits,<br>LSB |                       |
| N+1 | Dato de 32bits         | Dato de 32bits         | Dato de 32bits,<br>MSB |                        |                       |
| 6   |                        |                        |                        |                        |                       |

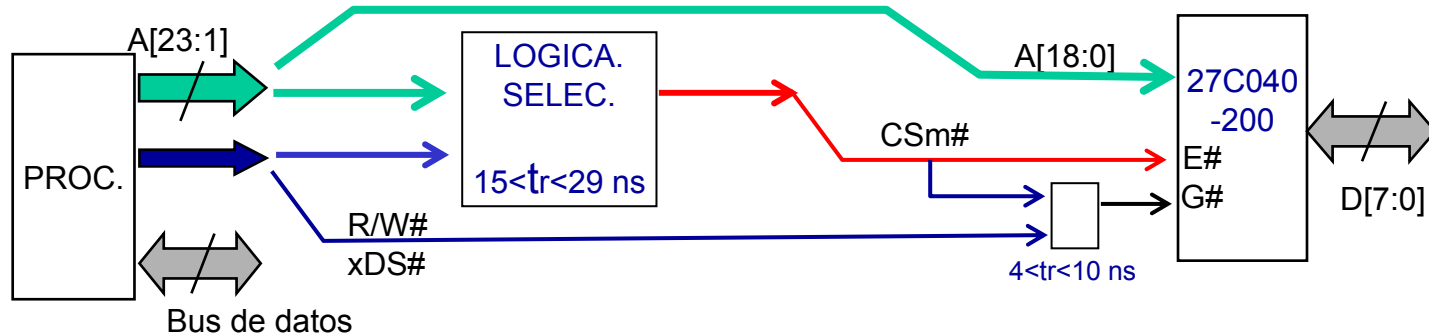
Figura 1.18. Ordenación de datos little endian en un sistema  $\mu$ P con bus de datos de 32 bits.

# Ejercicio

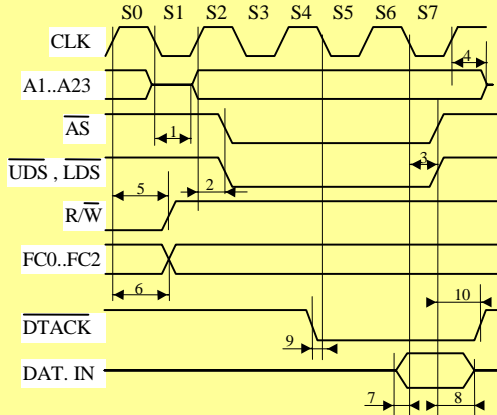
Dibujar las conexiones del bus de datos según la estructura de bancos necesaria.

Calcular los ciclos de espera a introducir.

Calcular los tiempos de acceso de una memoria para 0 ciclos de espera.



Ciclo de lectura



|    | tmín (ns) | tmáx (ns) |
|----|-----------|-----------|
| 1  |           | 25        |
| 2  | 3         | 25        |
| 3  | 3         | 25        |
| 4  | 0         |           |
| 5  | 0         | 25        |
| 6  | 0         | 25        |
| 7  | 5         |           |
| 8  | 0         |           |
| 9  | 5         |           |
| 10 | 0         | 95        |

| Parameter Symbols |           | Description  | Test Setup           | Am27C040 |      |      |      | Unit |    |
|-------------------|-----------|--|----------------------|----------|------|------|------|------|----|
| JEDEC             | Std.      |  |                      | -90      | -120 | -150 | -200 |      |    |
| $t_{AVQV}$        | $t_{ACC}$ | Address to Output Delay  | CE# = OE# = $V_{IL}$ | Max      | 90   | 120  | 150  | 200  | ns |
| $t_{ELQV}$        | $t_{CE}$  | Chip Enable to Output Delay  | OE# = $V_{IL}$       | Max      | 90   | 120  | 150  | 200  | ns |
| $t_{GLQV}$        | $t_{OE}$  | Output Enable to Output Delay  | CE# = $V_{IL}$       | Max      | 40   | 50   | 65   | 75   | ns |
| $t_{EHQZ}$        | $t_{GF}$  | Chip Enable High or Output Enable High, Whichever Occurs First, to Output High Z |                      | Max      | 30   | 30   | 30   | 40   | ns |
| $t_{GHQZ}$        | (Note 2)  |  |                      |          |      |      |      |      |    |
| $t_{AXQX}$        | $t_{OH}$  | Output Hold Time from Addresses, CE# or OE#, Whichever Occurs First              |                      | Min      | 0    | 0    | 0    | 0    | ns |