MEMORY PRODUCTS

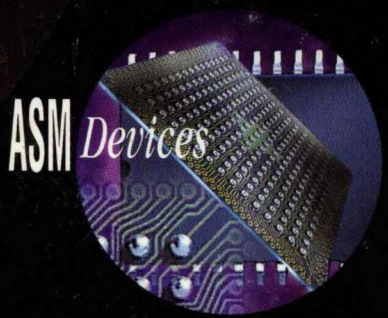
APPLICATION-SPECIFIC MEMORY DATA BOOK

FALL 1995

NEC

DATA BOOK

Fall 1995




Application-

Specific

Memory



NEC

# Application Specific Memory

## 1995 Data Book

Document No. M11002EU1V0DB00

Printed in the United States of America.

# Index


Selection Guide ..... 1

Dual Port Graphics Buffer ..... 7

Line Buffer ..... 189

Field Buffer ..... 227

Rambus™ DRAM ..... 253

Synchronous GRAM ..... 357

## NOTES FOR CMOS DEVICES

### ① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note: Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

### ② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note: No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS device behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to $V_{DD}$ or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

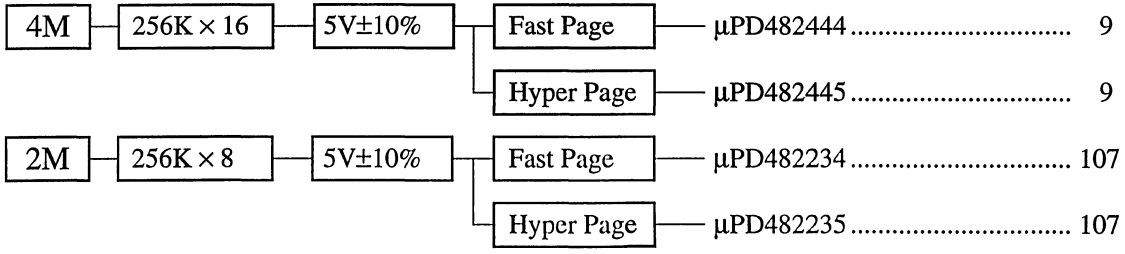
### ③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note: Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

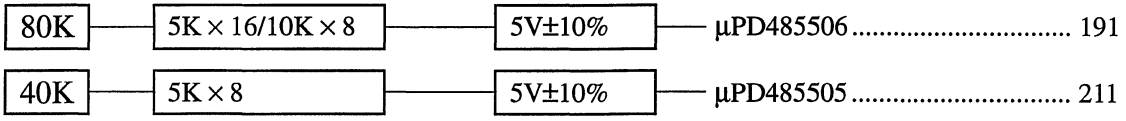
# Contents


Selection Guide ..... 1

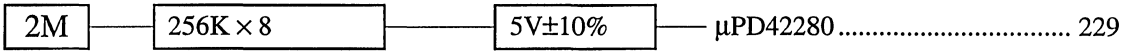
Dual Port Graphics Buffer ..... 7



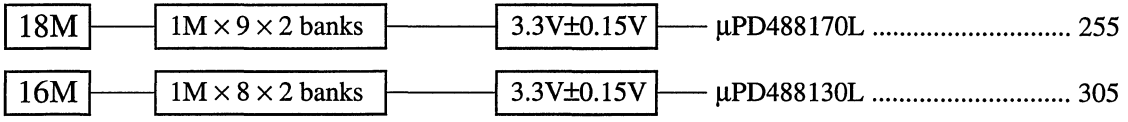

Line Buffer ..... 189



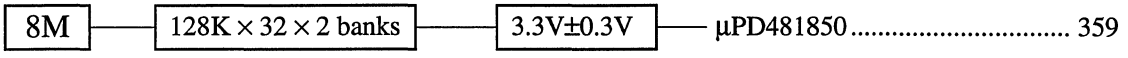

Field Buffer ..... 227




Rambus DRAM ..... 253



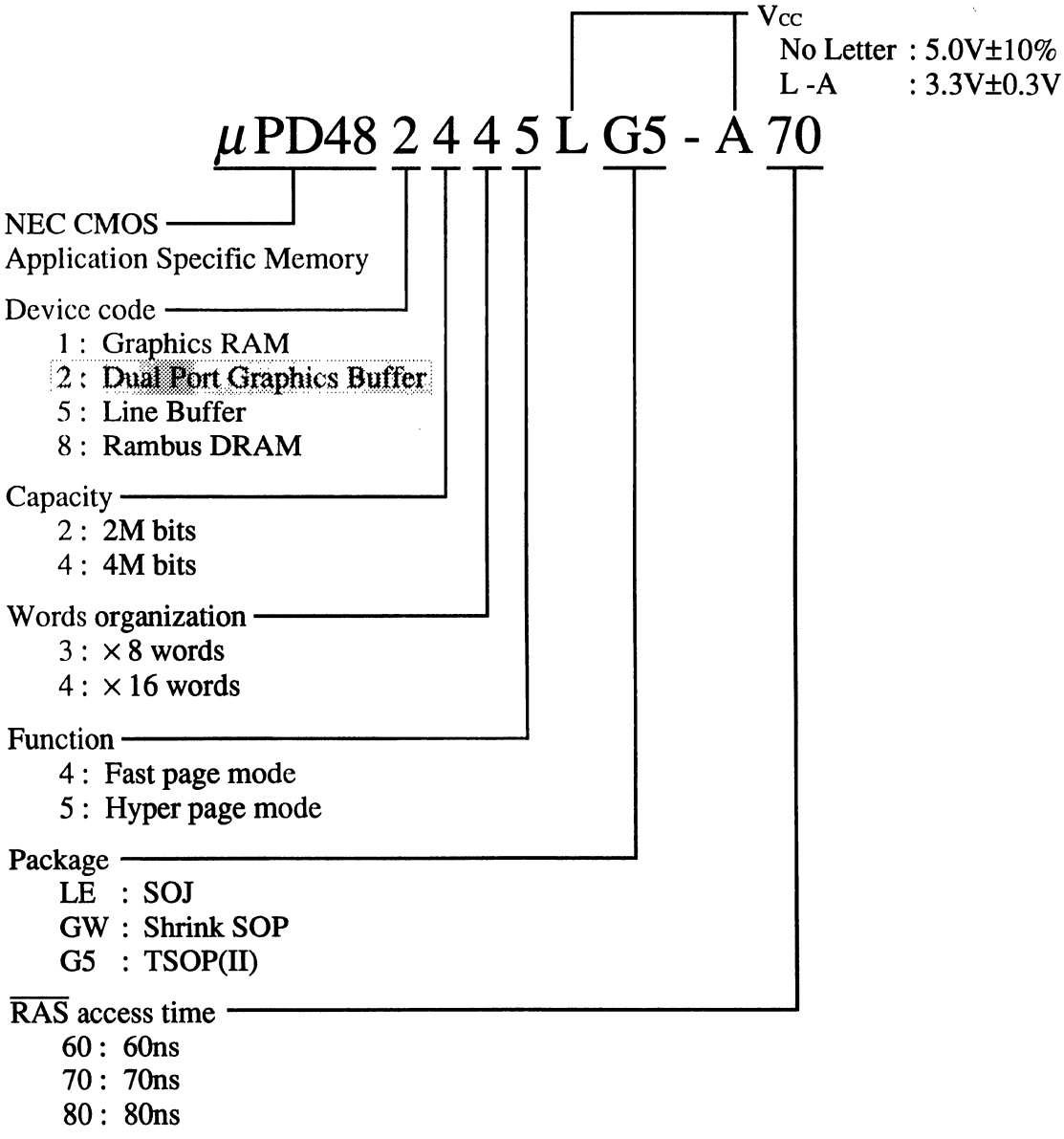

Synchronous GRAM ..... 357



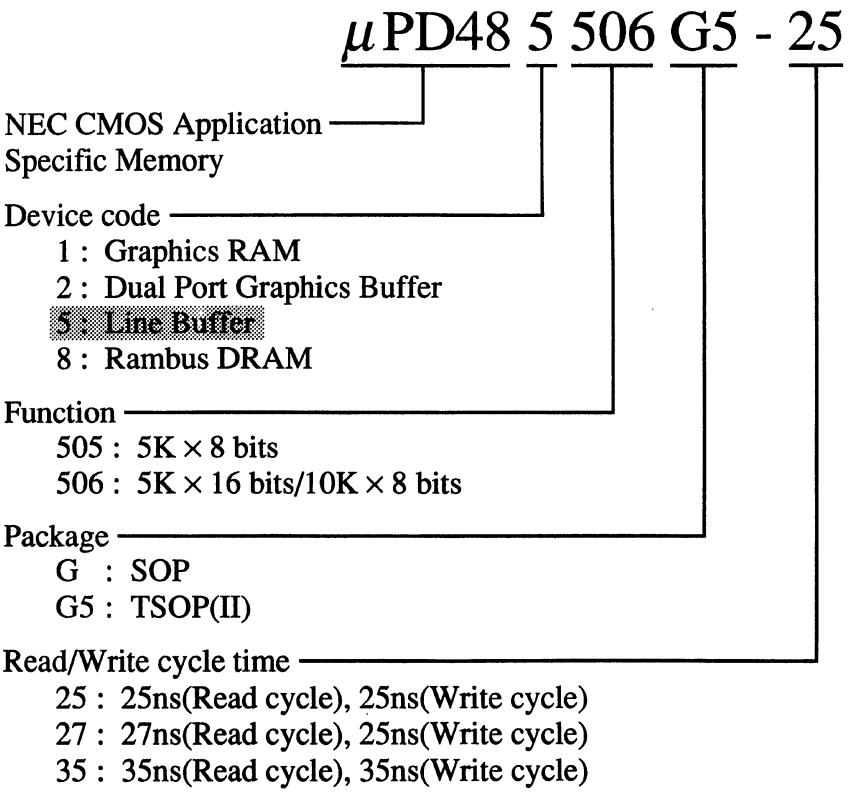
# Selection Guide

# Part Number

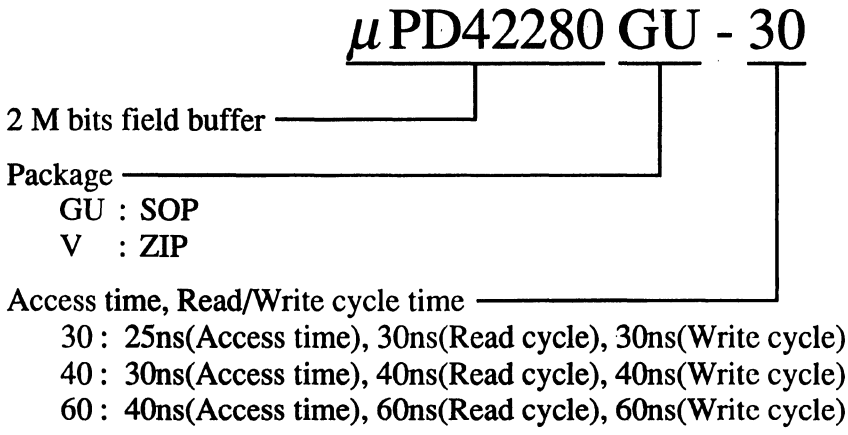
## Dual Port Graphics Buffer



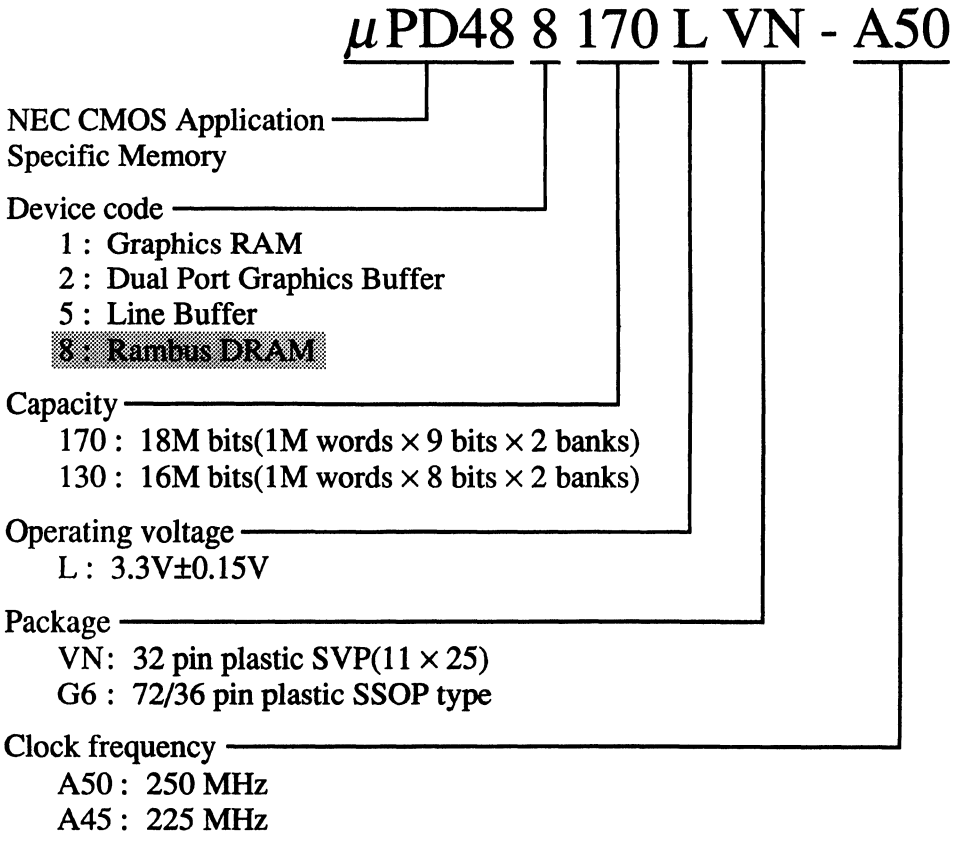
# Line Buffer


μPD48 5 506 G5 - 25
NEC CMOS Application Specific Memory
Device code
1 : Graphics RAM
2 : Dual Port Graphics Buffer
5 : Line Buffer
8 : Rambus DRAM
Function
505 : 5K × 8 bits
506 : 5K × 16 bits/10K × 8 bits
Package
G : SOP
G5 : TSOP(II)
Read/Write cycle time
25 : 25ns(Read cycle), 25ns(Write cycle)
27 : 27ns(Read cycle), 25ns(Write cycle)
35 : 35ns(Read cycle), 35ns(Write cycle)

# Field Buffer



# Rambus DRAM


μPD48 8 170 L VN - A50
NEC CMOS Application Specific Memory
Device code
1 : Graphics RAM
2 : Dual Port Graphics Buffer
5 : Line Buffer
8 : Rambus DRAM
Capacity
170 : 18M bits(1M words × 9 bits × 2 banks)
130 : 16M bits(1M words × 8 bits × 2 banks)
Operating voltage
L : 3.3V±0.15V
Package
VN: 32 pin plastic SVP(11 × 25)
G6 : 72/36 pin plastic SSOP type
Clock frequency
A50 : 250 MHz
A45 : 225 MHz

# Dual Port Graphics Buffer

NEC

MOS INTEGRATED CIRCUIT

# μPD482444, 482445

## 4M-Bit Dual Port Graphics Buffer

## 256K WORDS BY 16 BITS

### Description

The μPD482444 and μPD482445 have a random access port and a serial access port. The random access port has a 4M-bit (262,144 words × 16 bits) memory cell array structure. The serial access port can perform clock operations of up to 50 MHz from the 8K-bit data register (512 words × 16 bits).

To simplify the graphics system design, the split data transfer function and binary boundary jump function have been adopted so that the number of split data registers can be programmed with the software during serial read/write operations.

The μPD482445 is provided with the hyper page mode, an improved version of the fast page mode of the μPD482444. The random access port can input and output data by $\overline{CAS}$ clock operations of up to 33 MHz. The power supply voltage is 5 V ± 10 %.

### Features

Dual port structure (Random access port, Serial access port)

- Random access port (262,144-word × 16-bit structure)

**μPD482444**

| | μPD482444-60 | μPD482444-70 |
|---|---|---|
| $\overline{RAS}$ access time | 60 ns(MAX.) | 70 ns(MAX.) |
| Fast page mode cycle time | 35 ns(MIN.) | 40 ns(MIN.) |

**μPD482445**

| | μPD482445-60 | μPD482445-70 |
|---|---|---|
| $\overline{RAS}$ access time | 60 ns(MAX.) | 70 ns(MAX.) |
| Hyper page mode cycle time | 30 ns(MIN.) | 35 ns(MIN.) |

- Flash write function[Note]
- Block write function (8 columns)[Note]
- Mask write (Write-per-bit function)
- 512 refresh cycles /8 ms
- $\overline{CAS}$ before $\overline{RAS}$ refresh, $\overline{RAS}$ only refresh, Hidden refresh

**Note** Write-per-bit can be specified.

● Serial access port (512 words × 16 bits organization)

- Serial read/write cycle time

| μPD482444-60<br>μPD482445-60 | μPD482444-70<br>μPD482445-70 |
|---|---|
| 20 ns(MIN.) | 22 ns(MIN.) |

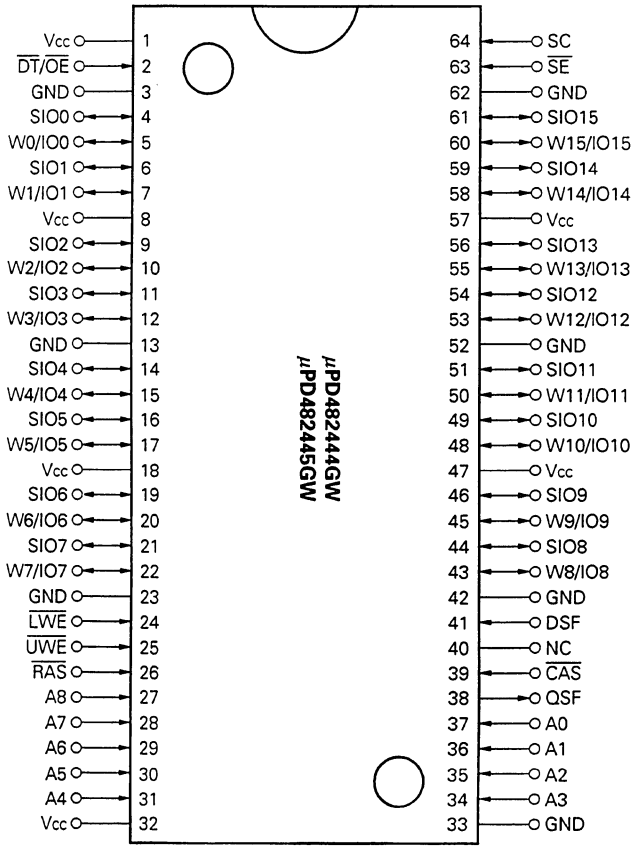
- Serial data read/write
- Split buffer data transfer
- Binary boundary jump function

## Ordering Information

| Part Number | $\overline{RAS}$ Access Time ns (MAX.) | Package | Power Supply Voltage | Page Mode |
|---|---|---|---|---|
| μPD482444GW-60 | 60 | 64-pin plastic shrink SOP (525 mil) | 5 V ± 10 % | Fast page mode |
| μPD482444GW-70 | 70 | | | |
| | | | | |
| μPD482445GW-60 | 60 | 64-pin plastic shrink SOP (525 mil) | 5 V ± 10 % | Hyper page mode |
| μPD482445GW-70 | 70 | | | |
| | | | | |

## Pin Configurations (Marking Side)

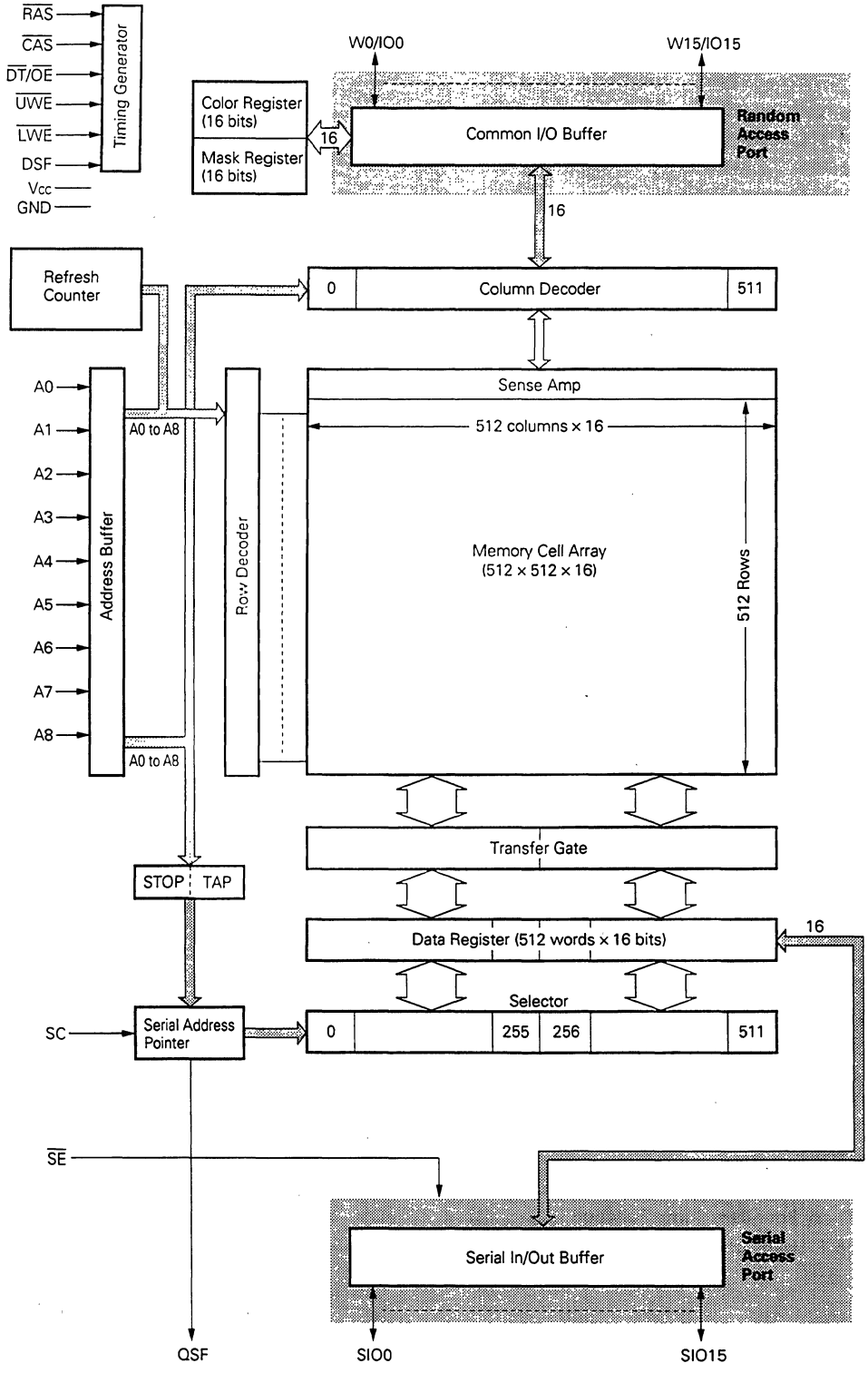
**64-Pin Plastic Shrink SOP (525 mil)**




| | | |
|---|---|---|
| A0 to A8 | : | Address inputs |
| W0 to W15/IO0 to IO15 | : | Mask data selects/Data inputs and outputs |
| SIO0 to SIO15 | : | Serial data inputs and outputs |
| $\overline{RAS}$ | : | Row address strobe |
| $\overline{CAS}$ | : | Column address strobe |
| $\overline{DT/OE}$ | : | Data transfer/Output enable |
| $\overline{UWE}$, $\overline{LWE}$ | : | Write-per-bit/Write enable |
| $\overline{SE}$ | : | Serial data input/Output enable |
| SC | : | Serial clock |
| QSF | : | Special function output |
| DSF | : | Special function enable |
| Vcc | : | Power supply voltage |
| GND | : | Ground |
| NC[Note] | : | No connection |

**Note** Some signals can be applied because this pin is not connected to the inside of the chip.

## Block Diagram



## 1. Pin Functions

This product is equipped with the $\overline{RAS}$, $\overline{CAS}$, $\overline{UWE}$, $\overline{LWE}$, $\overline{DT/OE}$, A0 to A8, DSF, SC, $\overline{SE}$ inputs, QSF output, and W0 to W15/IO0 to IO15, SIO0 to SIO15 input/output pins.

(1/3)

| Pin Name | Input/ Output | Function |
|---|---|---|
| $\overline{RAS}$ (Row address strobe) | Input | This signal latches the row addresses (A0 to A8), selects the corresponding word line, and activates the sense amplifier. It also refreshes the memory cell array of the one line (8,192 bits) selected from the row addresses (A0 to A8).<br><br>It also serves as the signal which selects the following operations.<br>• Write-per-bit • Flash write<br>• $\overline{CAS}$ before $\overline{RAS}$ refresh • Split data transfer |
| $\overline{CAS}$ (Column address strobe) | | This signal latches the column addresses (A0 to A8), selects the digit line connecting the sense amplifier, and activates the output circuit which outputs data to the random access port.<br><br>It also serves as the signal which selects the following operations.<br>• Read/write • Block write<br>• Color register set • Mask register set |
| A0 to A8 (Address inputs) | | These are the address input pins, TAP register input pins, and STOP register input pins.<br><br>**Address input**<br>This is a 9-bit address bus. It inputs a total of 18 bits of the address signal, starting from the upper 9 bits (row address) and then followed by the lower 9 bits (column bits) (address multiplex method). Using these, one word memory cells (16 bits) are selected from the 262,144 words × 16 bits memory cell array.<br>During use, specify the row address, activate the $\overline{RAS}$ signal, latch the row address, switch to the column address, and activate the $\overline{CAS}$ signal. After activating the $\overline{RAS}$ and $\overline{CAS}$ signals, each address signal is taken into the device. For this reason, the address input setup time ($t_{ASR}$, $t_{ASC}$) and hold time ($t_{RAH}$, $t_{CAH}$) are specified for activating the $\overline{RAS}$ and $\overline{CAS}$ signals.<br><br>**TAP Register Input**<br>In the data transfer cycle, this TAP register input pin functions as the address input pin which selects the memory cell for transferring (9 bits are latched at the falling edge of $\overline{RAS}$) and the TAP register data input pin which specifies the start addresses of the serial read/write operation after data transfer (9 bits are latched at the falling edge of the $\overline{CAS}$).<br><br>**STOP Register Input**<br>This pin functions as the STOP register input pin when the STOP register is set (STOP register data (9 bits) are latched at the falling edge of the $\overline{RAS}$.) |

(2/3)

| Pin Name | Input/ Output | Function |
|---|---|---|
| $\overline{DT}/\overline{OE}$ (Data transfer/ output enable) | Input | These are the data transfer control signal and read operation control signal respectively. They have different functions in the data transfer cycle and read cycle.<br><br>**Data transfer control signal (In data transfer cycle)**<br>The data transfer cycle is initiated when a low level is input to this pin at the falling edge of $\overline{RAS}$.<br><br>**Read operations control signal (In read cycle)**<br>Read operation is performed when this signal, and the $\overline{RAS}$ and $\overline{CAS}$ signals are activated. The input/output pin is high impedance when this signal is not activated. When the $\overline{UWE}$ and $\overline{LWE}$ signals are activated while the $\overline{DT}/\overline{OE}$ signals are activated, the $\overline{DT}/\overline{OE}$ signals are invalid in the memory and read operations cannot be performed. |
| $\overline{UWE}$, $\overline{LWE}$ (Write enable) | | These are the write operation control signal and mask write cycle (write-per-bit function) mask data input control signal, respectively.<br>$\overline{UWE}$ controls the upper bytes (W8 to W15/IO8 to IO15) and $\overline{LWE}$ controls the lower bytes (W0 to W7/IO0 to IO7) of the input/output pins.<br>When this signal, $\overline{RAS}$ and $\overline{CAS}$ signals are activated, write operations or mask write can be performed. These mode are determined by the level of $\overline{UWE}$ and $\overline{LWE}$ at the falling edge of $\overline{RAS}$.<br>• High level.....8 or 16-bit write cycle<br>• Low level.....Mask write cycle (Write-per-bit) |
| DSF (Special function enable) | | This signal controls the selection of functions.<br>The selection of functions is determined by the level of this signal at the falling edge of the $\overline{RAS}$ and $\overline{CAS}$. The functions will change as follows when this signal is high level.<br>• The data transfer cycle changes to a split data transfer cycle.<br>• The read/write cycle of each $\overline{RAS}$ clock changes to the flash write cycle.<br>• The write cycle of each $\overline{CAS}$ clock changes to the block write cycle. |
| W0 to W15/IO0 to IO15 (Mask data selects/ Data inputs, outputs) | Input/ Output | These are normally 16-bit data bus and are used for inputting and outputting data. (IO0 to IO15).<br>Function as the mask data input pins (W0 to W15) in the mask write cycle (write-per-bit function).<br>Write operations can be performed only for W0 to W15 that are input with a high level at the falling edge of $\overline{RAS}$ (new mask data).<br>Functions as the column selection data input pin in the block write cycle. |

(3/3)

| Pin Name | Input/ Output | Function |
|---|---|---|
| SC (Serial clock) | Input | This pin inputs the clock which controls the serial access port operation.<br><br>**Serial Read**<br>The data of the data register which is synchronized with the rising edge of the SC are output from the SIO0 to SIO15 pins and kept until the next SC rising edge.<br><br>**Serial Write**<br>The data from the SIO0 to SIO15 pins are latched at the rising edge of the SC and written in the data register. |
| $\overline{SE}$ (Serial data input/ output enable) | | This is a control pin for the serial access port input/output buffer.<br>It controls data output during serial reading and controls data input during serial writing.<br>By inputting the serial clock, the serial pointer will operate even if $\overline{SE}$ has not been activated (high level input). |
| SIO0 to SIO15 (Serial data inputs/ outputs) | Input/ Output | These are the serial data input and output pins of the serial access port. |
| QSF (Special function output) | Output | This is a position discrimination pin of the serial pointer (upper side or lower side).<br>Which side is being serial accessed (upper side or lower side) can be discriminated according to the output of this pin.<br>• High level ..... Upper side (Addresses 256 to 511)<br>• Low level ..... Lower side (Addresses 0 to 255) |

## 2. Random Access Port Operations

The operation mode is determined by the $\overline{CAS}$, $\overline{DT/OE}$, $\overline{UWE}$, $\overline{LWE}$, and DSF level at the falling edge of $\overline{RAS}$ and DSF level at the falling edge of $\overline{CAS}$.

**Table 2-1. Operation Mode**

| $\overline{RAS}$ Falling Edge | | | | | $\overline{CAS}$ Falling Edge | Operation Mode | |
|---|---|---|---|---|---|---|---|
| $\overline{CAS}$ | $\overline{DT/OE}$ | $\overline{UWE}$ | $\overline{LWE}$ | DSF | DSF | | |
| H | H | H | H | L | L | Read/Write Cycle | Read/Write cycle |
| H | H | H | H | L | H | | Block write cycle |
| H | H | L | L | L | L | | Mask write cycle[Note 1] |
| H | H | L | H | L | L | | Upper byte mask write cycle[Note 1] |
| H | H | H | L | L | L | | Lower byte mask write cycle[Note 1] |
| H | H | L | L | L | H | | Block mask write cycle[Note 1] |
| H | H | L | H | L | H | | Upper byte block mask write cycle[Note 1] |
| H | H | H | L | L | H | | Lower byte block mask write cycle[Note 1] |
| H | H | H | H | H | H | | Color register set cycle |
| H | H | H | H | H | L | | Write mask register set cycle |
| H | H | L | L | H | × | | Flash write cycle[Note 1] |
| H | H | L | H | H | × | | Upper byte flash write cycle[Note 1] |
| H | H | H | L | H | × | | Lower byte flash write cycle[Note 1] |
| H | L | H | H | L | × | Data Transfer Cycle | Single read data transfer cycle |
| H | L | H | H | H | × | | Split read data transfer cycle |
| H | L | L | L | L | × | | Single write data transfer cycle[Note 1] |
| H | L | L | L | H | × | | Split write data transfer cycle[Note 1] |
| L | × | × | × | L | × | Refresh Cycle | $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (Option reset)[Note 1, 2] |
| L | × | H | H | H | × | | $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (No reset) |
| L | × | L | L | H | × | | $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (STOP register set)[Note 2] |
| H | H | × | × | × | × | | $\overline{RAS}$ only refresh cycle |

**Notes 1.** Observe the following conditions when using the new mask data or old mask data in these cycles.

**(1) Old mask data**

Can be used after setting the mask data using the write mask register set cycle.

**(2) New mask data**

Can be used after selecting the new mask register using the write mask register set cycle (new mask selection) and after the optional reset cycle.

**2.** The STOP register is set to "FFH (11111111)" by the optional reset cycle.

**Remark** H : High level, L : Low level, × : High level or low level

### 2.1 Random Read Cycle

This product has a common 16-bit input/output pin. To output data, specify the address using the $\overline{RAS}$ and $\overline{CAS}$ clocks and then set $\overline{DT}/\overline{OE}$ to low level.

The data output will be kept until one of the following conditions is set.

(1) Set $\overline{RAS}$ and $\overline{CAS}$ to high level
(2) Set $\overline{DT}/\overline{OE}$ to high level
(3) Set $\overline{UWE}$ and $\overline{LWE}$ to low level ($\overline{UWE}$ controls the upper bytes, $\overline{LWE}$ controls the lower bytes)

The read cycle and data transfer cycle are differentiated according to the level of $\overline{DT}/\overline{OE}$ at the falling edge of the $\overline{RAS}$ clock. If $\overline{DT}/\overline{OE}$ is set to low level at the falling edge of the $\overline{RAS}$ clock, data transfer cycle operations will be initiated. Therefore, to set the read cycle, input a high level above $t_{DHH}$ (MIN.) to $\overline{DT}/\overline{OE}$ from the falling edge of the $\overline{RAS}$ clock, and then input a low level.

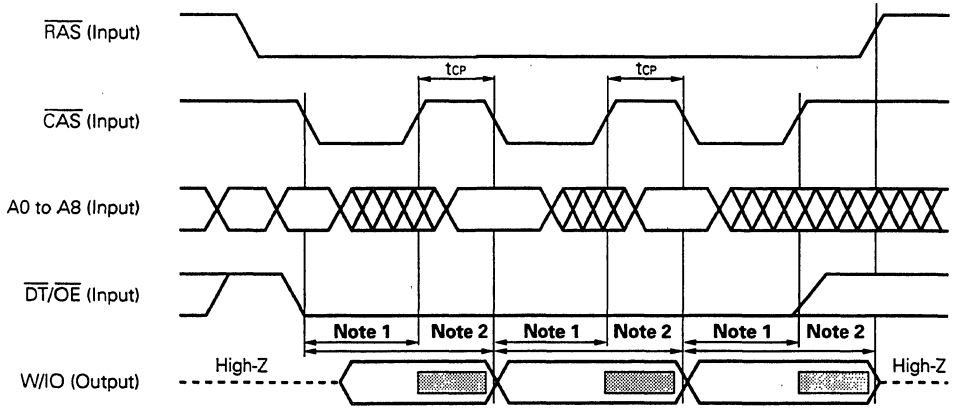
**Caution Set the DSF to low level at the falling edge of $\overline{RAS}$. If set to high level, the memory cell data cannot be output.**

#### 2.1.1 Extended Read Data Output (μPD482445)

The μPD482445 adopts the hyper page mode cycle which is a faster resd/write cycle than the fast page mode of the μPD482444 (Hyper page mode cycle time: 30 ns (MIN.)).

With this cycle, the read data output can be kept until the next $\overline{CAS}$ cycle, and because the output is extended, the minimum cycle can easily be used. For example, by fixing $\overline{DT}/\overline{OE}$ at low level after dropping $\overline{RAS}$ and executing the hyper page read cycle, each time the column address is latched at the falling edge of $\overline{CAS}$, the data output will be updated and kept until the next falling edge of $\overline{CAS}$. As a result, the output will be extended only during $\overline{CAS}$ precharge time ($t_{CP}$) as compared to the normal fast page mode.

**Figure 2-1. Extended Data Output of Hyper Page Mode**




**Notes 1.** Time during which the output data is kept in the fast page read cycle.

**2.** Time during which the output data is kept in the hyper page read cycle (▒ part: Extended data output).

### 2.2 Random Write Cycle (Early Write, Late Write, Read Modify Write)

There are three types of random write cycles-the early write, late write, and read modify write. To use these cycles, activate the $\overline{RAS}$ and $\overline{CAS}$ clocks and set $\overline{UWE}$ and $\overline{LWE}$ to low level. In addition, as this product has two write enables, data input can be controlled for every 8 bits (upper byte and lower byte). $\overline{UWE}$ controls the upper bytes (W8 to W15/IO8 to IO15) while $\overline{LWE}$ controls the lower bytes (W0 to W7/IO0 to IO7). Byte write cycle can therefore be performed by controlling $\overline{UWE}$ and $\overline{LWE}$.

The random write cycle, regardless of the word/byte write cycle, latches the word data (16 bits) input to the data bus. By inputting a low level to $\overline{UWE}$ (or $\overline{LWE}$) during the byte write cycle, the latched word (16 bits) data will be written only in the upper byte (or lower byte) and the data of the unselected lower byte (or upper byte) will be ignored. In the same write cycle, by inputting a low level to $\overline{LWE}$ (or $\overline{UWE}$) later, the ignored lower byte (or upper byte) data can be written. By controlling the $\overline{UWE}$ and $\overline{LWE}$ pins, the word data (16 bits) in the same cycle can be written in one byte (8 bits).

The $\overline{UWE}$ and $\overline{LWE}$ also control the mask data for the write-per-bit function (mask write cycle). Therefore, when performing the normal write cycle which does not use the write-per-bit function, set these pins to high level at the falling edge of the $\overline{RAS}$ clock.

#### 2.2.1 Early Write Cycle

The early write cycle controls data writing according to the $\overline{CAS}$ clock.

To execute this cycle, set $\overline{UWE}$ and $\overline{LWE}$ to low level earlier than the $\overline{CAS}$ clock. The write data is taken into the device at the falling edge of the $\overline{CAS}$ clock.

#### 2.2.2 Late Write Cycle

The late write cycle controls data writing according to the $\overline{WE}$ clock.

To execute this cycle, set $\overline{UWE}$ and $\overline{LWE}$ to low level later than the $\overline{CAS}$ clock. The write data is taken into the device at the falling edge of $\overline{UWE}$ and $\overline{LWE}$. To set the output to high impedance at this time, keep $\overline{DT}/\overline{OE}$ at high level until $\overline{UWE}$ and $\overline{LWE}$ are input.

#### 2.2.3 Read Modify Write Cycle

The read modify write cycle performs data reading and writing in one $\overline{RAS}$ and $\overline{CAS}$ cycle.

To execute this cycle, delay $\overline{UWE}$ and $\overline{LWE}$ from the late write cycle by $t_{RWD}$ (MIN.), $t_{CWD}$ (MIN.), and $t_{AWD}$ (MIN.). Follow the $t_{OEZ}$ and $t_{OED}$ specifications so that the output data and input data do not clash in the data bus. The data after modification can be input after more than $t_{OED}$ (MIN.) from the rising edge of $\overline{DT}/\overline{OE}$.

### 2.3 Fast Page Mode Cycle (μPD482444)

The μPD482444 adopts the fast page mode. This mode accesses memory cells in the same row array in about 1/3 of the time taken by the normal random read/write cycle. This fast page mode cycle is executed by repeating the $\overline{CAS}$ clock cycle more than two times while the $\overline{RAS}$ clock is being activated. In this mode read, write and read modify write cycles are available for each of the consecutive $\overline{CAS}$ cycles within the same $\overline{RAS}$ cycle.

### 2.4 Hyper Page Mode Cycle (μPD482445, 482445L)

The μPD482445 adopts a hyper page mode cycle which is a faster read/write cycle than the fast page mode of the μPD482444 (Hyper page mode cycle time: 30 ns (MIN.)).

In this cycle, because the read data output is kept until the following $\overline{CAS}$ cycle and as a result, the output is extended, the minimum cycle can easily be used. The output is extended compared to the normal fast page mode of μPD482444. Refer to **2.1.1 Extended Read Data Output.**

### 2.5 Flash Write Cycle

This cycle writes the color register data in a 8,192-bit or 4,096-bit memory cell in one cycle. The memory cell range for one flash write cycle is 512 columns on the same row address (512-column × 16 · IO = 8,192 bits or 512-column × 8 · IO = 4,096 bits).

#### 2.5.1 Execution of Flash Write Cycle

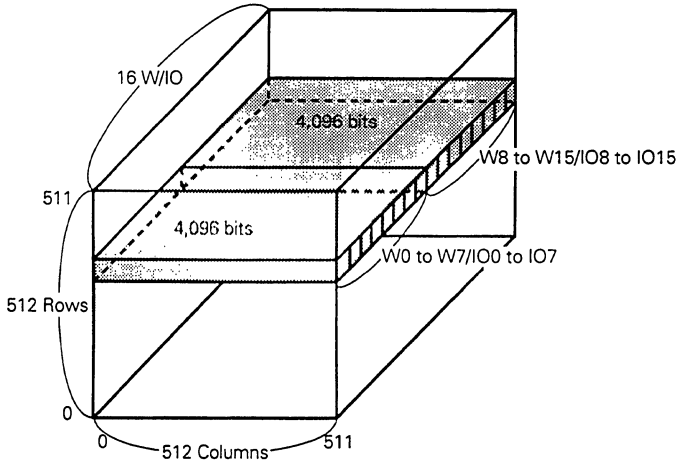
**(1) Execution of flash write for word (512-column × 16 · IO = 8,192 bits)**

To execute the flash write cycle, set both $\overline{UWE}$ and $\overline{LWE}$ to low level at the falling edge of $\overline{RAS}$. By using the write-per-bit function (new mask data/old mask data), only the required W/IO can be selected and written.

**(2) Execution of flash write for byte (512-column × 8 · IO = 4,096 bits)**

To execute the flash write cycle, set either $\overline{UWE}$ or $\overline{LWE}$ to low level at the falling edge of $\overline{RAS}$. By using the write-per-bit function (new mask data/old mask data), only the required W/IO can be selected and written.

**Figure 2-2. Memory Cell Range That Can be Written with Flash Write Cycle**




**Remark** ▒ is the memory cell range that can be written in one flash write cycle.

### 2.6 Block Write Cycle

This cycle writes the color register data in 128-bit or 64-bit memory cell in one cycle. The memory cell range in which data can be written in one block write cycle is eight continuous columns on one row address (8-column × 16 · IO = 128 bits or 8-column × 8 · IO = 64 bits).

Any column of the eight columns can be selected and writing prohibited. Determine whether to write or prohibit writing according to the data selected for column.

#### 2.6.1 Free Column Selection

Determine which column to select according to the W/IO pin to which the data selected for the column is to be input.

The eight columns (1st to 8th) correspond to W0 to W15/IO0 to IO15 to which the data selected for column will be input (The following table shows the 1st to 8th columns specified by A0, A1, and A2 and the corresponding W/IO pins to which the data selected will be input.).

#### 2.6.2 Column Select Data

Input column select data for every eight columns at the upper 64 bits and lower 64 bits (a total of 16 columns). The data will be written if the column select data is "1". Writing will be prohibited if the column select data is "0".

### 2.6.3 Execution of Block Write Cycle

At the falling edge of the slowest signal ($\overline{CAS}$, $\overline{UWE}$, or $\overline{LWE}$), input the "1" column select data or "0" column select data to W0 to W15/IO0 to IO15 corresponding to columns 1st to 8th.

By using the write-per-bit (new mask data/old mask data) function, only the required W/IO can be selected and written.

**Table 2-2. I/O Pins Input with Column Select Data Corresponding to Columns 1st to 8th**

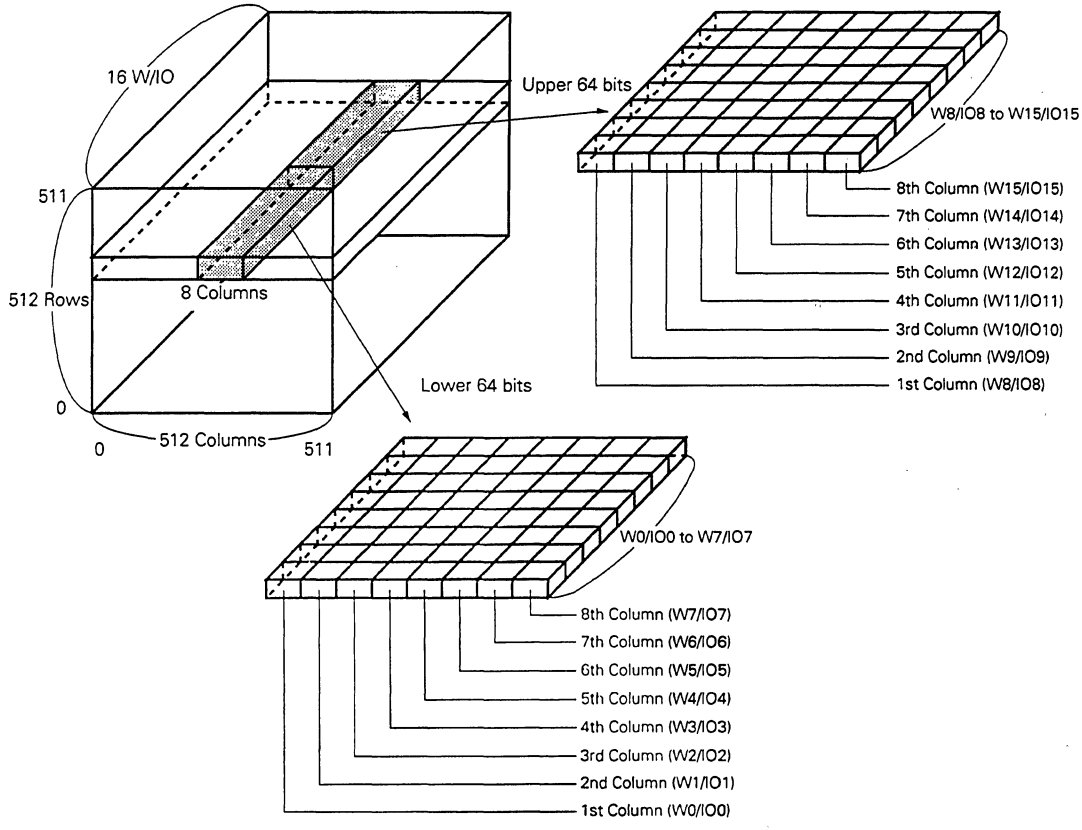
**Column Select Data of Lower Byte (IO0 to IO7)**

| Selected 8 Columns | Column Address and Corresponding W/IO Pin | | | | Column Select Data | Writing |
|---|---|---|---|---|---|---|
| | A2 | A1 | A0 | IO | | |
| 1st column | 0 | 0 | 0 | IO0 | 1 | Yes |
| | | | | | 0 | No |
| 2nd column | 0 | 0 | 1 | IO1 | 1 | Yes |
| | | | | | 0 | No |
| 3rd column | 0 | 1 | 0 | IO2 | 1 | Yes |
| | | | | | 0 | No |
| 4th column | 0 | 1 | 1 | IO3 | 1 | Yes |
| | | | | | 0 | No |
| 5th column | 1 | 0 | 0 | IO4 | 1 | Yes |
| | | | | | 0 | No |
| 6th column | 1 | 0 | 1 | IO5 | 1 | Yes |
| | | | | | 0 | No |
| 7th column | 1 | 1 | 0 | IO6 | 1 | Yes |
| | | | | | 0 | No |
| 8th column | 1 | 1 | 1 | IO7 | 1 | Yes |
| | | | | | 0 | No |

**Column Select Data of Upper Byte (IO8 to IO15)**

| Selected 8 Columns | Column Address and Corresponding W/IO Pin | | | | Column Select Data | Writing |
|---|---|---|---|---|---|---|
| | A2 | A1 | A0 | IO | | |
| 1st column | 0 | 0 | 0 | IO8 | 1 | Yes |
| | | | | | 0 | No |
| 2nd column | 0 | 0 | 1 | IO9 | 1 | Yes |
| | | | | | 0 | No |
| 3rd column | 0 | 1 | 0 | IO10 | 1 | Yes |
| | | | | | 0 | No |
| 4th column | 0 | 1 | 1 | IO11 | 1 | Yes |
| | | | | | 0 | No |
| 5th column | 1 | 0 | 0 | IO12 | 1 | Yes |
| | | | | | 0 | No |
| 6th column | 1 | 0 | 1 | IO13 | 1 | Yes |
| | | | | | 0 | No |
| 7th column | 1 | 1 | 0 | IO14 | 1 | Yes |
| | | | | | 0 | No |
| 8th column | 1 | 1 | 1 | IO15 | 1 | Yes |
| | | | | | 0 | No |

**Figure 2-3. Memory Cell Range That Can be Written in Block Write Cycle**




**Remarks 1.** ☐ is the memory cell range that can be written in one block write cycle.

**2.** ( ) is the W/IO pin input with the column select data.

### 2.7 Register Set Cycle (Color Register, Write Mask Register)

This cycle writes data in the color register and write mask register. To execute the register set cycle, set $\overline{CAS}$, $\overline{DT/OE}$, $\overline{UWE}$, $\overline{LWE}$ and DSF to high level at the falling edge of $\overline{RAS}$. Determine which register to select according to the DSF level at the falling edge of $\overline{CAS}$.

The register set cycle also serves as the $\overline{RAS}$ only refresh cycle.

**Table 2-3. Register Selection**

| DSF level at $\overline{CAS}$ falling edge | Selected register |
|---|---|
| High level | Color register |
| Low level | Write mask register |

**Caution After selecting the write mask register and writing the mask data, the write-per-bit function in the mask write cycle will be set for the old mask register. Refer to 2.8.1 Write-Per-Bit Function.**

### 2.8 Mask Write Cycle

Cycles that use the write-per-bit function during the random write cycle, flash write cycle, block write cycle, write data transfer cycle, are called mask write cycles. In the fast page/hyper page mode write cycle, the mask data cannot be changed during the $\overline{CAS}$ cycle.

#### 2.8.1 Write-Per-Bit Function

The write-per-bit function writes data using the mask data only in the required IO-pin. It writes when the mask data is "1" and prohibits writing when the data is "0".

**Table 2-4. Mask Data Selection**

| W Pin | Mask Data | Writing |
|---|---|---|
| W0 to W15 | 1 | Yes |
| | 0 | No |

#### 2.8.2 Selecting Mask Data

There are two ways of selecting mask data. One is the new mask data method and the other is the old mask data method.

With the new mask data method, new mask data is set in the cycle writing. With the old mask data, mask data set in the write mask register is used.

**(1) New Mask Data Method**

The new mask data method consists of the write mask register set cycle (new mask selection) method and $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (optional reset cycle) method.

**(a) Method Using Write Mask Register Set Cycle (New Mask Selection)**

To switch to the mode using new mask data, keep the $\overline{DT}/\overline{OE}$, $\overline{UWE}$, $\overline{LWE}$ DSF to high level and set the $\overline{CAS}$ and DSF to high level at the falling edge of $\overline{RAS}$, the DSF to low level at the falling edge of $\overline{CAS}$, and start up the next $\overline{RAS}$ and $\overline{CAS}$ after the $t_{CAS}$ and $t_{RAS}$.

As a result, the write-per-bit function can be used using the new mask data from the next mask write cycle.

**(b) Method Using $\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (Optional Reset Cycle)**

To switch to the mode using new mask data, set the DSF to low level at the falling edge of $\overline{CAS}$ in the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle.

As a result, the write-per-bit function can be used using the old mask data from the next mask write cycle.

**(2) Old Mask Data Method**

To switch to the mode using old mask data, set the DSF to low level at the falling edge of $\overline{CAS}$ in the write mask register set cycle, and write the mask data in the write mask register.

As a result, the write-per-bit function can be used using the old mask data from the next mask write cycle.

### 2.8.3 Execution of Mask Write Cycle

To execute the write-per-bit function, select the new mask data method or old mask data method, and set $\overline{UWE}$ and $\overline{LWE}$ to low level at the falling edge of $\overline{RAS}$ of each write cycle ($\overline{UWE}$ controls the upper byte (W8 to W15/IO8 to IO15) and $\overline{LWE}$ controls the lower byte (W0 to W7/IO0 to IO7).). At this time, input the mask data to the W pin in the write cycle using the new mask data. In the write cycle using the old mask data, as the mask data set to the write mask register will be used, there is no need to input the mask data to the W pin.

This function is valid only at the falling edge of $\overline{RAS}$. In the fast page/hyper page mode write cycle, the mask data determined in the first $\overline{RAS}$ cycle for moving onto the next fast page/hyper page mode will be valid while the fast page/hyper page mode write cycle continues.

### 2.9 Refresh Cycle

The refresh cycle of this product consists of the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle and refresh cycle using external address inputs ($\overline{RAS}$ only refresh and read/write refresh). The refresh period is the same as the 2M-bit dual port graphics buffer (× 8), 512 cycles/8 ms.

### 2.9.1 Refresh Cycle Using External Address Input ($\overline{RAS}$ Only Refresh and Read/Write Refresh)

By specifying the row address using the 9 bits between A0 to A8 at the falling edge of $\overline{RAS}$, setting $\overline{CAS}$ to high level, and keeping $\overline{CAS}$ at high level while $\overline{RAS}$ is low level, the memory cells on the specified row address (512 × 16 bits) can be refreshed. At this time, refresh is executed, W0 to W15/IO0 to IO15 pins are kept at high impedance, and information such as memory contents, register data, function settings, etc. are all also kept.

At the falling edge of $\overline{RAS}$, all cycles whose $\overline{CAS}$ are high level input the external address. Therefore, in addition to the read/write cycle operations, etc. refresh operations similar to the $\overline{RAS}$ only refresh operations will be performed. For this reason, in systems in which addresses in the memory are always increased or decreased, it may not be necessary to perform refresh again.

When several devices exist on one bus, data will clash in the bus during the above read/write operations unless each device is equipped with a buffer. Consequently, as it is necessary to set the I/O line to high impedance beforehand during refresh, normally the $\overline{RAS}$ only refresh operation is used.

### 2.9.2 $\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (Including Hidden Refresh)

When $\overline{CAS}$ is set to low level at the falling edge of $\overline{RAS}$, the refresh address is supplied from the internal refresh address counter. The internal refresh address counter is increased automatically each time this refresh cycle is executed.

During this refresh cycle, functions of random access port and serial access port are selected as follows according to the DSF, $\overline{UWE}$, and $\overline{LWE}$ levels at the falling edge of $\overline{RAS}$.

**(1) When DSF is low level: Optional reset**

All STOP register data become "1" and the mask write cycle switches to the new mask data method.

**(2) When DSF is high level and $\overline{UWE}$, $\overline{LWE}$ are low level: STOP register set**

The STOP register data is input from the A0 to A8 pins at the falling edge of $\overline{RAS}$.

**(3) When DSF, $\overline{UWE}$, and $\overline{LWE}$ are high level: No reset**

Only refresh operations are performed and the function selection state is kept.

In all cases, the W/IO pin is kept at high impedance. When $\overline{CAS}$ and $\overline{DT}/\overline{OE}$ are kept low level while the mode is changed to the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle following the read cycle, and $\overline{RAS}$ is activated, the hidden refresh cycle will be initiated. In this cycle, the W/IO pin does not become high impedance and the data read in the former read cycle will be kept as it is.

Because internal memory operations are equivalent to $\overline{CAS}$ before $\overline{RAS}$ refresh, no external addresses are required.

Like $\overline{CAS}$ before $\overline{RAS}$ refresh, in the hidden cycle, functions will be selected according to the level of DSF, $\overline{UWE}$, and $\overline{LWE}$ at the falling edge of $\overline{RAS}$. Operations are guaranteed when DSF is low level and when DSF, $\overline{UWE}$, and $\overline{LWE}$ are high level.

### 3. Serial Access Port Operations

There are two types of data transfer cycles-data transfer from the random access port to the serial access port (read data transfer) and data transfer from the serial access port to the random access port (write data transfer). There are also two types of data transfer methods-single data transfer and split data transfer.

To set the data transfer cycle, input high level to $\overline{CAS}$ and input low level to $\overline{DT}/\overline{OE}$ at the falling edge of $\overline{RAS}$.

The data transfer type differs according to the input levels of $\overline{UWE}$, $\overline{LWE}$, and DSF at the falling edge of $\overline{RAS}$.

**Table 3-1. Serial Access Port Operation Mode**

| At $\overline{RAS}$ Falling Edge | | | | Data Transfer Type | Transfer Direction | |
|---|---|---|---|---|---|---|
| $\overline{CAS}$ | $\overline{DT}/\overline{OE}$ | $\overline{UWE}$, $\overline{LWE}$ | DSF | | Transfer Source | Transfer Destination |
| H | L | H | L | Single read data transfer | Random access port | Serial access port |
| H | L | H | H | Split read data transfer | | |
| H | L | L | L | Single mask write data transfer[Note] | Serial access port | Random access port |
| H | L | L | H | Split mask write data transfer[Note] | | |

**Note** Write-per-bit function can be specified.

**Remark** H: High level, L: Low level

### 3.1 Single Data Transfer Method

With this method, 512 words × 16 bits (whole memory range of serial access port) data is transferred at one time. This method can be used in both write data transfer and read data transfer.

#### 3.1.1 Single Read Data Transfer Cycle

This cycle transfers the 8K-bit (512 words × 16 bits) data of the random access port to the serial access port in one cycle.

**(a) Setting of Single Read Data Transfer Cycle**

To set the data transfer cycle, input a high level to $\overline{CAS}$, $\overline{UWE}$, and $\overline{LWE}$ and low level to $\overline{DT}/\overline{OE}$ and DSF at the falling edge of $\overline{RAS}$.

Using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$, the memory cells (512 words × 16 bits) of the transfer source of the random access port can be selected. The address data input to A0 to A8 at the falling edge of $\overline{CAS}$ will be latched as the TAP register data. Refer to **3.4 TAP Register**.

**(b) Execution of Single Read Data Transfer Cycle**

To execute the data transfer cycle, set the single read data transfer cycle and then input a high level to $\overline{DT}/\overline{OE}$ and $\overline{RAS}$.

When SC is active (edge control), data transfer will be executed at the rising edge of $\overline{DT}/\overline{OE}$. When SC is inactive (self control), it will be executed at the rising edge of $\overline{RAS}$. At the same time, the serial address pointer jumps to the start column (TAP) of the next serial read cycle, and the TAP register will be set the empty state.

After the transfer is completed, the new serial access port data is output after $t_{SCA}$ following the rise of the SC clock that occurs after $t_{SDH}$ if the SC is active, and after $t_{SDHR}$ if SC is inactive.

**Caution When the single read data transfer cycle is executed while the serial access port is performing serial write operations, the serial access port will start serial read operations at the rising edge of $\overline{RAS}$. Refer to 4. Electrical Characteristics Read Data Transfer Cycle (Serial Write → Serial Read Switching) Timings.**

### 3.1.2 Single Mask Write Data Transfer Cycle

This cycle transfers 8K-bit (512 word × 16 bits) data of the serial access port to the random access port in one cycle. Because $\overline{UWE}$ and $\overline{LWE}$ are low level at the falling edge of $\overline{RAS}$, the write-per-bit function always functions in this transfer cycle. Refer to **2.8 Mask Write Cycle.**

**(a) Setting of Single Mask Write Data Transfer Cycle**

To set this cycle, latch the data to be transferred to the serial access port, and then input a high level to $\overline{CAS}$ and low level to $\overline{DT/OE}$, $\overline{UWE}$, $\overline{LWE}$, and DSF at the falling edge of $\overline{RAS}$. Because the write-per-bit function functions in this transfer operation, for the new mask data method, the mask data must be supplied to W0 to W15 at the falling edge of $\overline{RAS}$, and for the old mask data method, there is no need to control the mask data.

The memory cells (512 words × 16 bits) of the transfer destination of the random access port are selected using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$. The address data input to A0 to A8 at the falling edge of $\overline{CAS}$ is input as the TAP register data. Refer to **3.4 TAP Register.**

**(b) Execution of Single Mask Write Data Transfer Cycle**

To execute this cycle, set the single write data transfer cycle and then input high level to $\overline{RAS}$. Data will be transferred at the rising edge of $\overline{RAS}$. At the same time, the serial address pointer jumps to the start column (TAP) of the next serial write cycle, and the TAP register will be set the empty state.

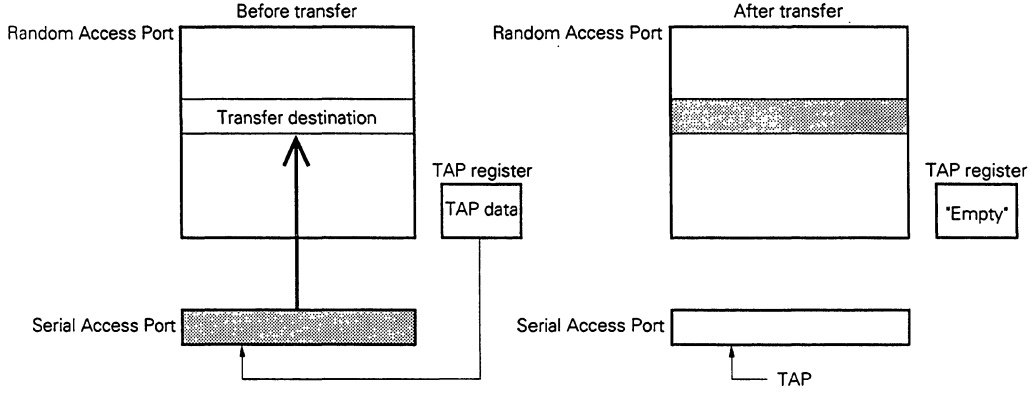
After the transfer is completed, the new serial access port data is latched at the rising edge of the SC clock that occurs after $t_{SDHR}$.

**Caution 1. When the single mask write data transfer cycle is executed while the serial access port is performing serial read operations, the serial access port will start serial write operations at the rising edge of $\overline{RAS}$. Refer to 4. Electrical Characteristics Write Data Transfer Cycle (Serial Read → Serial Write Switching) Timings.**

**2. Always make $\overline{CAS}$ low level in the write data transfer cycle and latch TAP. If write data transfer is performed without setting TAP, serial access port operations cannot be ensured until either one of the following points. If the SC clock is input during this time, the serial register value also cannot be guaranteed.**

- **Until the falling edge of $\overline{CAS}$ during the write data transfer cycle**
- **Until the read data transfer cycle is executed again**

**Figure 3-1. Single Write Data Transfer and TAP Operation**



### 3.2 Split Data Transfer Method

With this method, the 512 words × 16 bits (whole memory range of serial access port) data is divided into the lower column (0 to 255) and upper column (256 to 511), each consisting of 256 words × 16 bits.

Because the columns are divided into upper and lower columns with this method, data transfer can be performed on lower column (or upper column) while performing read/write operations in the upper column (or lower column). For this reason, transfer timing design is easy. This transfer method can be used in both write data transfer and read data transfer.

#### 3.2.1 Split Read Data Transfer Cycle

This cycle divides the 8K-bit (512 words × 16 bits) data of the random access port into the lower and upper columns and transfers them to the serial access port.

In this cycle, the serial read/write can be performed in the columns to which data is not transfer.

**(a) Setting of Split Read Data Transfer Cycle**

To set this cycle, input a high level to $\overline{CAS}$, $\overline{UWE}$, $\overline{LWE}$ and DSF, and low level to $\overline{DT}/\overline{OE}$ at the falling edge of $\overline{RAS}$.

The memory cells (512 words × 16 bits) of the transfer source of the random access port are selected using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$. And the address data input to A0 to A7 at the falling edge of $\overline{CAS}$ is latched as the TAP register data of serial access port. There is no need to control address data input to A8. Refer to **3.4 TAP Register**.

**(b) Execution of Split Read Data Transfer Cycle**

To execute this cycle, set the split read data transfer cycle and then input the high level to $\overline{RAS}$. Data will be transferred at the rising edge of $\overline{RAS}$. Data is transferred from the random access port to the serial access port automatically at the column side where serial access port is inactive. To confirm the transferred column side, check the output state of the QSF pin. Refer to **3.3.3 QSF Pin Output**.

When the serial address pointer comes to the jump source address specified by the STOP register, the serial address pointer jumps to the start column (TAP) of the serial read/write cycle at the inactive column side, and the TAP register will be set the empty state.

### 3.2.2 Split Mask Write Data Transfer Cycle

This cycle divides the 8K-bit (512 words × 16 bits) data of the serial access port into the lower and upper columns and transfers them to the random access port.

In this cycle, serial read/write can be performed for columns to which data is not transferred.

Because $\overline{UWE}$ and $\overline{LWE}$ are low level at the falling edge of $\overline{RAS}$, the write-per-bit function always functions in this transfer cycle. Refer to **2.8 Mask Write Cycle.**

**(a) Setting of Split Mask Write Data Transfer Cycle**

To set this data transfer cycle, input a high level to $\overline{CAS}$ and DSF and low level to $\overline{DT/OE}$, $\overline{UWE}$, and $\overline{LWE}$ at the falling edge of $\overline{RAS}$. Because the write-per-bit function functions in this transfer operation, for the new mask data method, the mask data must be supplied to W0 to W15 at the falling edge of $\overline{RAS}$, and for the old mask data method, there is no need to control the mask data.

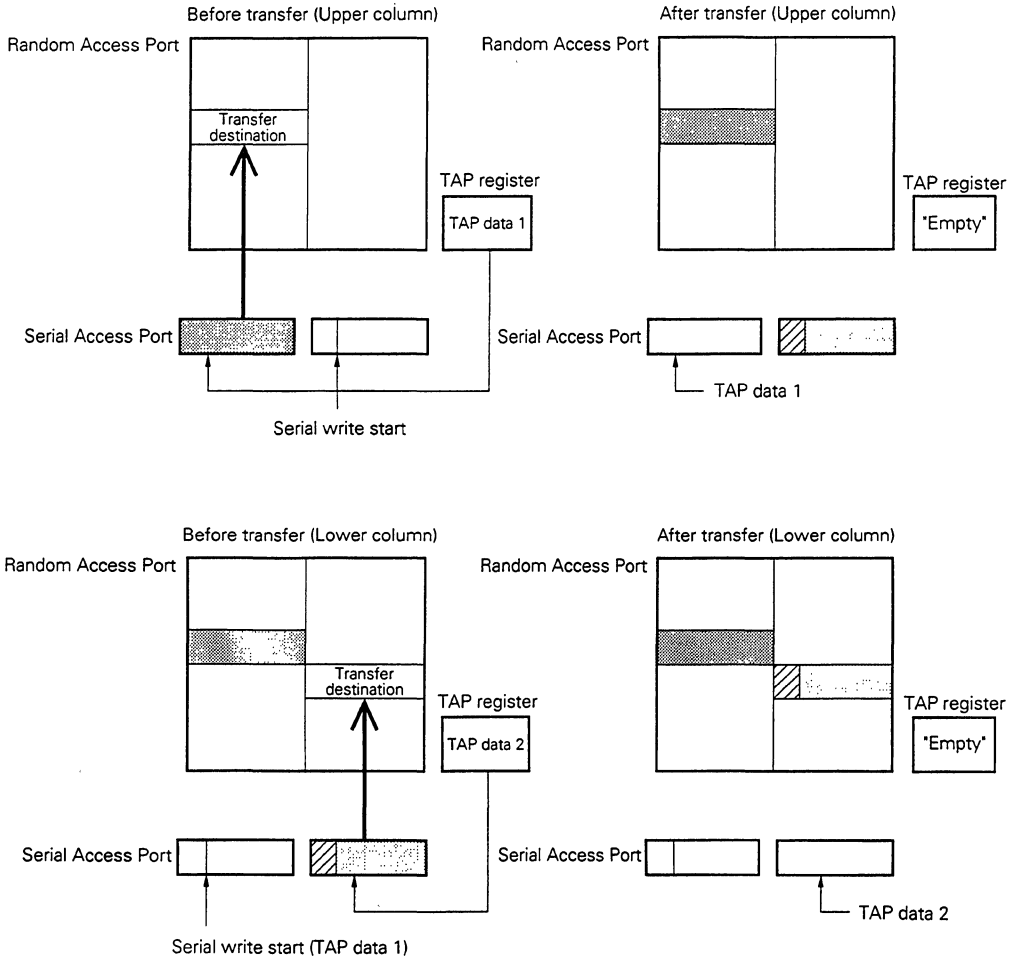
The memory cells (512 words × 16 bits) of the transfer destination of the random access port are selected using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$. The address data input to A0 to A7 at the falling edge of $\overline{CAS}$ is input as the TAP register data. There is no need to control address data input to A8. Refer to **3.4 TAP Register.**

**(b) Execution of Split Mask Write Data Transfer Cycle**

To execute this cycle, set the split write data transfer cycle and then input high level to $\overline{RAS}$. Data will be transferred at the rising edge of $\overline{RAS}$. Data is transferred from the serial access port to the random access port automatically at the column side where the serial access port is inactive. To confirm the transferred column side, check the output state of the QSF pin. Refer to **3.3.3 QSF Pin Output.**

When the serial address pointer comes to the jump source address specified by the STOP register, the serial address pointer jumps to the start column (TAP) of the serial read/write cycle at the inactive column side, and the TAP register will be set the empty state.

**Figure 3-2. Split Mask Write Data Transfer and TAP Operations**




### 3.3 Serial Read/Write

The serial access port (512K × 16 bits) is independent from the random access port and can perform read and write operations. The serial access port performing single data transfer and split data transfer can not perform read and write operations independently.

**Caution When the power is turned on, the serial access port sets into the input (write) mode and the SIO pin is the high impedance state.**

### 3.3.1 Serial Read Cycle

To set the serial read cycle, perform the single read data transfer cycle (The mode will not change in the split read data transfer cycle.).

Execute the single read data transfer cycle and latch the data and TAP data. By inputting a clock signal to the SC pin and inputting a low level to the $\overline{SE}$ pin, data will be output from the serial address pointer specified by TAP register. The data synchronizes with the rising edge of the SC clock and is output from the SIO0 to SIO15 pin, and the data is kept until the next rising edge of the SC clock.

**(a) Reading-Jump**

The $\overline{SE}$ pin controls the SIO pin output buffer independently from the SC clock. By setting the $\overline{SE}$ pin to high level even while inputting the SC clock, SIO0 to SIO15 pins become high impedance. But the operations of serial address pointer will be continued while the SC clock is being input even though reading has been prohibited from $\overline{SE}$ pin. Reading-jump of the column can be performed using this function.

### 3.3.2 Serial Write Cycle

To set the serial write cycle, perform the single write data transfer cycle (The mode will not change in the split write data transfer cycle.). To prevent the transfer data from being written in the memory cell of the random access port, set all bits of the mask data to "0" and control the mask data.

Execute the single write data transfer cycle and set the serial write cycle. By inputting the clock signal to the SC pin and inputting a low level to the $\overline{SE}$ pin, data can be latched from the serial address pointer specified by TAP register. The data synchronizes with the rising edge of the SC clock and is input from SIO0 to SIO15 pins. Be sure to follow the specifications for the setup time ($t_{SES}$) and hold time ($t_{SEH}$) of $\overline{SE}$ pin for the SC clock.

**(a) Writing-Jumps (Intermittent Writing)**

The $\overline{SE}$ pin controls writing operations independently from the SC clock. By setting the $\overline{SE}$ pin to high level even while inputting the SC clock, writing will not be executed. But the operations of serial address pointer will be continued while the SC clock is being input even though writing has been prohibited from $\overline{SE}$ pin. These functions enable writing-jumps (intermittent writing) to be performed. The masked data is kept as the old data.

### 3.3.3 QSF Pin Output

QSF pin determines whether the serial address pointer is at the upper column side (addresses 256 to 511) or the lower column side (addresses 0 to 255) at the rising edge of the following SC clock during serial read or write. In other words, it outputs the uppermost bit (A8) of the column address of the serial address pointer.

The following table shows the QSF pin output state and the access pointer of following SC clocks.

| Access Address of Following SC clock | QSF Output |
|---|---|
| Addresses 0 to 255 | Low level |
| Addresses 256 to 511 | High level |

### 3.4 TAP (Top Access Point) Register

The TAP register is a data register which specifies the start address (first serial address point = TAP) of the serial read or serial write.

Set data to this register each time a transfer cycle is executed.

#### 3.4.1 Setting of TAP Register

The data input to A0 to A8 at the falling edge of $\overline{CAS}$ during the setting of a transfer cycle is set as the TAP register data. By executing the transfer cycle, the start address of the following serial read (or write) operations is specified by the data of the TAP register and the TAP register will be kept in the empty state until the TAP register is set again.

In the split data transfer cycle, because the inactive serial access port column addresses are specified by the data of the TAP register automatically, there is no need to control the A8 data.

**Caution When the TAP register is empty, the address following the 511 serial address point will be 0. In addition, because the serial address pointer will not jump to the column specified by the STOP register, the binary boundary jump function cannot be used. Refer to 3.6 Binary Boundary Jump Function.**

### 3.5 STOP Register

The STOP register is a data register which determines the column of the jump source when jumping to a different column side (lower column or upper column) in the split data transfer cycle. Five types of columns can be selected for starting jump (jumping is possible at 2, 4, 8, 16, and 32 points). The following table shows the correspondence between the column at the jump source and data of the STOP register.

Once set, the STOP register data is kept until it is set again.

#### 3.5.1 Setting of STOP Register

To set the STOP register, set $\overline{UWE}$ and $\overline{LWE}$ to low level at the falling edge of $\overline{RAS}$ in the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle. The data input to A0 to A7 will be input as the STOP register data.

**Table 3-2. STOP Register Data and Jump Source Column**

| STOP Register Data | | | | | Division | Bit Width | Jump Source Bit Column (Decimal Number) |
|---|---|---|---|---|---|---|---|
| A7 | A6 | A5 | A4 | A3 to A0 | | | |
| 1 | 1 | 1 | 1 | 1 | 1/2 | 256 | 255 |
| | | | | | | | 511 |
| 0 | 1 | 1 | 1 | 1 | 1/4 | 128 | 127, 255 |
| | | | | | | | 383, 511 |
| 0 | 0 | 1 | 1 | 1 | 1/8 | 64 | 63, 127, 191, 255 |
| | | | | | | | 319, 383, 447, 511 |
| 0 | 0 | 0 | 1 | 1 | 1/16 | 32 | 31, 63, 95, 127, 159, 191, 223, 255 |
| | | | | | | | 287, 319, 351, 383, 415, 447, 479, 511 |
| 0 | 0 | 0 | 0 | 1 | 1/32 | 16 | 15, 31, 47, 63, 79, 95, 111, 127, 143, 159, 175, 191, 207, 223, 239, 255 |
| | | | | | | | 271, 287, 303, 319, 335, 351, 367, 383, 399, 415, 431, 447, 463, 479, 495, 511 |

**Remark** A8: Don't care.

**Caution When the power is supplied, all STOP register data will be set to all "1".**

### 3.6 Binary Boundary Jump Function

This function causes the serial address pointer jump to the TAP specified by the TAP register when the pointer moves to a column specified by the STOP register (split data transfer).

This function cannot be used when the jump destination address is not set (TAP register is empty).

This function facilitates tile map application which divides the screen into tiles and manages data for each tile.

#### 3.6.1 Usage of Binary Boundary Jump Function

After setting the STOP register, execute the single read (or write) data transfer and initialize the serial access port. The initialization process will switch the serial access port read (or write) operations, set TAP, set the serial access port data, and set the TAP register to empty. By inputting the serial clock in this state, the serial access port will read (or write) operations from TAP in ascending order of address. Because the TAP register is in the empty state, the address at the jump source set by the STOP register will be ignored, and the serial address pointer will move on.

When the column to be jumped approaches, execute split data transfer and set new TAP data in the TAP register. The serial pointer will jump at the desired jump source address. Jump can be controlled freely by repeating these operations.

### 3.7 Special Operations

#### 3.7.1 Serial Address Set Operations

Because the serial address counter is undefined when the power up, the serial access port operations when the SC clock is input are not guaranteed. Execute single read (or write) transfer after turning on the power. The serial access port will be initialized, enabling serial access port operations to be performed.

#### 3.7.2 Lap Around Operations

If all the data of the register is read (write) during data transfer while the serial read (write) cycle is being executed, the serial pointer will repeat 0 to 511.

#### 3.7.3 Cycle After Power On

Execute the dummy cycle eight times more than 100 μs after $V_{CC}$ reaches the specified voltage in the recommended operation conditions.

If $\overline{RAS}$, $\overline{CAS}$, $\overline{DT/OE}$, $\overline{UWE}$, $\overline{LWE}$ are kept at high level when the power is turned on, the following will be set automatically.

- Serial access port ..... Input mode, SIO: High impedance
- Color register ..... Undefined
- Mask register ..... All "1"
- TAP register ..... Undefined
- STOP register ..... All "1"

## 4. Electrical Characteristics

### 4.1 μPD482444, 482445 (Power Supply Voltage $V_{CC}$ = 5 V ± 10 %)

**Absolute Maximum Ratings**

| Parameter | Symbol | Rating | Unit |
|---|---|---|---|
| Pin voltage | $V_T$ | −1.0 to +7.0 | V |
| Supply voltage | $V_{CC}$ | −1.0 to +7.0 | V |
| Output current | $I_O$ | 50 | mA |
| Power dissipation | $P_D$ | 1.5 | W |
| Operating ambient temperature | $T_A$ | 0 to 70 | °C |
| Storage temperature | $T_{stg}$ | −55 to +125 | °C |

**Caution Exposing the device to stress above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits in the operational sections of this characteristics. Exposure to Absolute Maximum rating conditions for extended periods may affect device reliability.**

**Recommended Operating Conditions**

| Parameter | Symbol | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|
| Supply voltage | $V_{CC}$ | 4.5 | 5.0 | 5.5 | V |
| High level input voltage | $V_{IH}$ | 2.4 | | 5.5 | V |
| Low level input voltage | $V_{IL}$ | −1.0 | | +0.8 | V |
| Operating ambient temperature | $T_A$ | 0 | | 70 | °C |

**DC Characteristics 1 (Recommended operating conditions unless otherwise noted)**

| Parameter | Symbol | Test conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input leakage current | $I_{IL}$ | $V_{IN}$ = 0 V to 5.5 V, Other inputs are 0 V | −10 | | +10 | μA |
| Output leakage current | $I_{OL}$ | W/IO, SIO, QSF are inactive, $V_{OUT}$ = 0 V to 5.5 V | −10 | | +10 | μA |
| Random access port high level output voltage | $V_{OH}$ (R) | $I_{OH}$ (R) = −1.0mA | 2.4 | | | V |
| Random access port low level output voltage | $V_{OL}$ (R) | $I_{OL}$ (R) = 2.1mA | | | 0.4 | V |
| Serial access port high level output voltage | $V_{OH}$ (S) | $I_{OH}$ (S) = −1.0mA | 2.4 | | | V |
| Serial access port low level output voltage | $V_{OL}$ (S) | $I_{OL}$ (S) = 2.1mA | | | 0.4 | V |

**Capacitance ($T_A$ = 25 °C, f = 1MHz)**

| Parameter | Symbol | Test conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_{I1}$ | $\overline{RAS}$, $\overline{CAS}$, $\overline{UWE}$, $\overline{LWE}$, $\overline{DT/OE}$, DSF, $\overline{SE}$, SC | | | 8 | pF |
| | $C_{I2}$ | A0 to A8 | | | 5 | |
| Input/Output Capacitance | $C_{IO}$ | W/IO (0 to 15), SIO (0 to 15) | | | 7 | pF |
| Output Capacitance | $C_O$ | QSF | | | 7 | pF |

**DC Characteristics 2 (Recommended operating conditions unless otherwise noted)[Note 1]**

| Random Access Port | Serial Access Port | | Symbol | μPD482444-60 μPD482445-60 | | μPD482444-70 μPD482445-70 | | Unit | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| | Standby | Active | | MIN. | MAX. | MIN. | MAX. | | |
| Random Read/Write Cycle $\overline{RAS}$, $\overline{CAS}$ cycle, $t_{RC} = t_{RC}$ (MIN.), Io = 0mA | ○ | | $I_{CC1}$ | | 110 | | 95 | mA | |
| | | ○ | $I_{CC7}$ | | 130 | | 110 | | |
| Standby $\overline{RAS} = \overline{CAS} = V_{IH}$, $D_{OUT}$ = high impedance | ○ | | $I_{CC2}$ | | 10 | | 10 | mA | Note 2 |
| | | ○ | $I_{CC8}$ | | 50 | | 45 | mA | Note 2 |
| $\overline{RAS}$ only refresh cycle $\overline{RAS}$ cycle, $\overline{CAS} = V_{IH}$, $t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC3}$ | | 100 | | 85 | mA | Note 3 |
| | | ○ | $I_{CC9}$ | | 140 | | 120 | | |
| Fast/Hyper page mode cycle $\overline{RAS} = V_{IL}$, $\overline{CAS}$ cycle, $t_{PC} = t_{PC}$ (MIN.) or $t_{HPC} = t_{HPC}$ (MIN.) | ○ | | $I_{CC4}$ | | 120 | | 105 | mA | Note 4, 5 |
| | | ○ | $I_{CC10}$ | | 150 | | 130 | | |
| $\overline{CAS}$ before $\overline{RAS}$ refresh cycle $t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC5}$ | | 100 | | 95 | mA | |
| | | ○ | $I_{CC11}$ | | 130 | | 120 | | |
| Data transfer cycle $t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC6}$ | | 120 | | 105 | mA | |
| | | ○ | $I_{CC12}$ | | 150 | | 130 | | |
| Color/Mask write register set cycle $t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC13}$ | | 90 | | 80 | mA | |
| | | ○ | $I_{CC14}$ | | 120 | | 105 | | |
| Flash write cycle $t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC15}$ | | 90 | | 80 | mA | |
| | | ○ | $I_{CC16}$ | | 120 | | 105 | | |
| Block write cycle $t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC17}$ | | 110 | | 100 | mA | |
| | | ○ | $I_{CC18}$ | | 140 | | 125 | | |
| Fast/Hyper page mode block write cycle $t_{PC} = t_{PC}$ (MIN.) or $t_{HPC} = t_{HPC}$ (MIN.) | ○ | | $I_{CC19}$ | | 135 | | 120 | mA | |
| | | ○ | $I_{CC20}$ | | 155 | | 135 | | Note 4, 5 |

**Notes** 1. No load on W/IO, SIO, QSF. The current consumption actually used depends on the output load and operating frequency of each pin.

2. A change in row addresses must not occur more than once in $t_{RC} = t_{RC}$ (MIN.).
3. When the address input is set to $V_{IH}$ or $V_{IL}$ during the $t_{RAS}$ period.
4. Value when the address in $t_{PC}$ one cycle is changed once when μPD482444 $t_{PC} = t_{PC}$ (MIN.).
5. Value when the address in $t_{HPC}$ one cycle is changed once when μPD482445 $t_{HPC} = t_{HPC}$ (MIN.).

**AC Characteristics ($T_A$ = 0 to 70 °C, $V_{CC}$ = 5.0 V ± 10 %, GND = 0 V)[Notes 1, 2, 3, 4]**

**(Common for μPD482444, 482445)** (1/4)

| Parameter | Symbol | μPD482444-60 μPD482445-60 | | μPD482444-70 μPD482445-70 | | Unit | Condition |
|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | | |
| Access time from $\overline{RAS}$ | $t_{RAC}$ | | 60 | | 70 | ns | **Note 5** |
| Access time from $\overline{CAS}$ | $t_{CAC}$ | | 18 | | 18 | ns | **Note 5** |
| Access time from column address | $t_{AA}$ | | 30 | | 35 | ns | **Note 5** |
| Access time from $\overline{CAS}$ trailing edge | $t_{ACP}$ | | 30 | | 35 | ns | |
| Access time from $\overline{OE}$ | $t_{OEA}$ | | 18 | | 18 | ns | |
| Serial output access time from SC | $t_{SCA}$ | | 15 | | 17 | ns | |
| Serial output access time from $\overline{SE}$ | $t_{SEA}$ | | 15 | | 17 | ns | |
| Output disable time from $\overline{SE}$ high | $t_{SEZ}$ | 0 | 15 | 0 | 15 | ns | **Note 6** |
| Random read or write cycle time | $t_{RC}$ | 110 | | 130 | | ns | |
| Read modify-write cycle time | $t_{RWC}$ | 160 | | 180 | | ns | |
| Transition time (Rise/Fall) | $t_T$ | 3 | 35 | 3 | 35 | ns | |
| $\overline{RAS}$ precharge time | $t_{RP}$ | 40 | | 50 | | ns | |
| $\overline{RAS}$ pulse width (Non page mode) | $t_{RAS}$ | 60 | 10,000 | 70 | 10,000 | ns | |
| $\overline{RAS}$ pulse width (Fast page/Hyper page mode) | $t_{RASP}$ | 60 | 125,000 | 70 | 125,000 | ns | |
| $\overline{RAS}$ hold time | $t_{RSH}$ | 15 | | 18 | | ns | |
| $\overline{CAS}$ precharge time (Non page mode) | $t_{CPN}$ | 10 | | 10 | | ns | |
| $\overline{CAS}$ precharge time (Fast page/Hyper page mode) | $t_{CP}$ | 10 | | 10 | | ns | |
| $\overline{CAS}$ hold time | $t_{CSH}$ | 60 | | 70 | | ns | |
| $\overline{RAS}$ to $\overline{CAS}$ delay time | $t_{RCD}$ | 25 | 40 | 30 | 50 | ns | **Note 5** |
| $\overline{CAS}$ high to $\overline{RAS}$ low precharge time | $t_{CRP}$ | 10 | | 10 | | ns | |
| $\overline{RAS}$ high to $\overline{CAS}$ low precharge time | $t_{RPC}$ | 10 | | 10 | | ns | |
| Row address setup time | $t_{ASR}$ | 0 | | 0 | | ns | |
| Row address hold time | $t_{RAH}$ | 15 | | 15 | | ns | |
| Column address setup time | $t_{ASC}$ | 0 | | 0 | | ns | |
| Column address hold time | $t_{CAH}$ | 10 | | 10 | | ns | |
| $\overline{RAS}$ to column address delay time | $t_{RAD}$ | 15 | 30 | 15 | 35 | ns | **Note 5** |
| Column address to $\overline{RAS}$ lead time | $t_{RAL}$ | 30 | | 35 | | ns | |
| Read command setup time | $t_{RCS}$ | 0 | | 0 | | ns | |

**(Common for μPD482444, 482445)** (2/4)

| Parameter | Symbol | μPD482444-60 μPD482445-60 | | μPD482444-70 μPD482445-70 | | Unit | Condition |
|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | | |
| Read command hold time after $\overline{RAS}$ high | $t_{RRH}$ | 0 | | 0 | | ns | Note 7 |
| Read command hold time after $\overline{CAS}$ high | $t_{RCH}$ | 0 | | 0 | | ns | Note 7 |
| $\overline{OE}$ hold time after $\overline{RAS}$ high | $t_{ORH}$ | 10 | | 10 | | ns | |
| $\overline{OE}$ hold time after $\overline{CAS}$ high | $t_{OCH}$ | 10 | | 10 | | ns | |
| Write command setup time | $t_{WCS}$ | 0 | | 0 | | ns | Note 9 |
| Write command hold time | $t_{WCH}$ | 12 | | 12 | | ns | |
| Write command pulse width | $t_{WP}$ | 12 | | 12 | | ns | |
| Write command to $\overline{RAS}$ lead time | $t_{RWL}$ | 20 | | 20 | | ns | |
| Write command to $\overline{CAS}$ lead time | $t_{CWL}$ | 15 | | 15 | | ns | |
| Data in setup time | $t_{DS}$ | 0 | | 0 | | ns | Note 8 |
| Data in hold time | $t_{DH}$ | 15 | | 15 | | ns | Note 8 |
| Refresh period | $t_{REF}$ | | 8 | | 8 | ms | |
| $\overline{CAS}$ to $\overline{UWE}$, $\overline{LWE}$ delay time | $t_{CWD}$ | 40 | | 40 | | ns | Note 9 |
| $\overline{RAS}$ to $\overline{UWE}$, $\overline{LWE}$ delay time | $t_{RWD}$ | 85 | | 90 | | ns | Note 9 |
| Column address to $\overline{UWE}$, $\overline{LWE}$ delay time | $t_{AWD}$ | 55 | | 55 | | ns | Note 9 |
| $\overline{CAS}$ setup time (for $\overline{CAS}$ before $\overline{RAS}$ refresh cycle) | $t_{CSR}$ | 5 | | 5 | | ns | |
| $\overline{CAS}$ hold time (for $\overline{CAS}$ before $\overline{RAS}$ refresh cycle) | $t_{CHR}$ | 10 | | 10 | | ns | |
| Masked byte write setup time | $t_{MCS}$ | 0 | | 0 | | ns | |
| Masked byte write to $\overline{RAS}$ hold time | $t_{MRH}$ | 0 | | 0 | | ns | |
| Masked byte write to $\overline{CAS}$ hold time | $t_{MCH}$ | 0 | | 0 | | ns | |
| $\overline{DT}$ low setup time | $t_{DLS}$ | 0 | | 0 | | ns | |
| $\overline{DT}$ low hold time after $\overline{RAS}$ low | $t_{RDH}$ | 55 | | 60 | | ns | Note 10 |
| $\overline{DT}$ low hold time after $\overline{RAS}$ low | $t_{RDHS}$ | 15 | | 15 | | ns | Note 10 |
| $\overline{DT}$ low hold time after $\overline{CAS}$ low | $t_{CDH}$ | 20 | | 20 | | ns | Note 10 |
| $\overline{DT}$ low hold time after address | $t_{ADD}$ | 25 | | 25 | | ns | Note 10 |
| SC high to $\overline{DT}$ high | $t_{SDD}$ | 0 | | 0 | | ns | Note 10 |
| SC high to $\overline{CAS}$ low | $t_{SSC}$ | 10 | | 10 | | ns | Note 10, 14, 15 |
| SC low hold time after $\overline{DT}$ high | $t_{SDH}$ | 60 | | 60 | | ns | Note 10 |

**(Common for μPD482444, 482445)** (3/4)

| Parameter | Symbol | μPD482444-60 μPD482445-60 | | μPD482444-70 μPD482445-70 | | Unit | Condition |
|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | | |
| SC low hold time after $\overline{DT}$ high | $t_{SDHR}$ | 60 | | 60 | | ns | **Note 10, 14** |
| $\overline{OE}$ high to data in setup delay time | $t_{OED}$ | 15 | | 15 | | ns | |
| $\overline{OE}$ high hold time after $\overline{UWE}$, $\overline{LWE}$ low | $t_{OEH}$ | 0 | | 0 | | ns | |
| Serial clock cycle time | $t_{SCC}$ | 20 | | 22 | | ns | |
| SC pulse width | $t_{SCH}$ | 5 | | 5 | | ns | |
| SC precharge time | $t_{SCL}$ | 5 | | 5 | | ns | |
| $\overline{SE}$ low to serial output setup delay time | $t_{SOO}$ | 3 | | 5 | | ns | |
| Serial output hold time after SC high | $t_{SOH}$ | 3 | | 5 | | ns | |
| $\overline{DT}$ high setup time | $t_{DHS}$ | 0 | | 0 | | ns | |
| $\overline{DT}$ high hold time | $t_{DHH}$ | 15 | | 15 | | ns | |
| $\overline{DT}$ high to $\overline{RAS}$ high delay time | $t_{DTR}$ | 0 | | 0 | | ns | **Note 10** |
| $\overline{DT}$ high pulse width | $t_{DTP}$ | 20 | | 20 | | ns | |
| $\overline{OE}$ to $\overline{RAS}$ inactive setup time | $t_{OES}$ | 0 | | 0 | | ns | |
| Write-per-bit setup time | $t_{WBS}$ | 0 | | 0 | | ns | |
| Write-per-bit hold time | $t_{WBH}$ | 15 | | 15 | | ns | |
| DSF setup time from $\overline{RAS}$ | $t_{FRS}$ | 0 | | 0 | | ns | |
| DSF hold time from $\overline{RAS}$ | $t_{FRH}$ | 15 | | 15 | | ns | |
| DSF setup time from $\overline{CAS}$ | $t_{FCS}$ | 0 | | 0 | | ns | |
| DSF hold time from $\overline{CAS}$ | $t_{FCH}$ | 12 | | 12 | | ns | |
| Write-per-bit selection setup time | $t_{WS}$ | 0 | | 0 | | ns | |
| Write-per-bit selection hold time | $t_{WH}$ | 15 | | 15 | | ns | |
| $\overline{SE}$ pulse width | $t_{SEE}$ | 5 | | 5 | | ns | |
| $\overline{SE}$ precharge time | $t_{SEP}$ | 5 | | 5 | | ns | |
| $\overline{SE}$ setup time | $t_{SES}$ | 0 | | 0 | | ns | |
| $\overline{SE}$ hold time from SC | $t_{SEH}$ | 10 | | 10 | | ns | |
| Serial data in setup time | $t_{SIS}$ | 0 | | 0 | | ns | |
| Serial data in hold time | $t_{SIH}$ | 10 | | 10 | | ns | |
| Serial input disable time from SC | $t_{SIZ}$ | 0 | | 0 | | ns | |
| Serial output disable time from $\overline{RAS}$ | $t_{SRZ}$ | 0 | | 0 | | ns | |

**(Common for μPD482444, 482445)** (4/4)

| Parameter | Symbol | μPD482444-60 μPD482445-60 | | μPD482444-70 μPD482445-70 | | Unit | Condition |
|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | | |
| Serial input enable time from $\overline{RAS}$ | $t_{SZH}$ | 40 | | 40 | | ns | |
| SC setup time from $\overline{RAS}$ | $t_{SRS}$ | 10 | | 10 | | ns | **Note 13, 14, 15** |
| SC hold time from $\overline{RAS}$ | $t_{SRH}$ | 10 | | 10 | | ns | **Note 13** |
| Propagation delay time from SC to QSF | $t_{PD}$ | 0 | 20 | 0 | 20 | ns | |
| Propagation delay time from $\overline{RAS}$ to QSF | $t_{RQD}$ | 0 | 80 | 0 | 95 | ns | |
| Propagation delay time from $\overline{CAS}$ to QSF | $t_{CQD}$ | 0 | 60 | 0 | 65 | ns | |
| Propagation delay time from $\overline{DT}/\overline{OE}$ to QSF | $t_{DQD}$ | 0 | 30 | 0 | 30 | ns | |
| Propagation delay time from $\overline{RAS}$ high to QSF | $t_{DQR}$ | 0 | 40 | 0 | 40 | ns | |

**(μPD482444 Only)**

| Parameter | Symbol | μPD482444-60 | | μPD482444-70 | | Unit | Condition |
|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | | |
| Output disable time from $\overline{CAS}$ high | $t_{OFF}$ | 0 | 15 | 0 | 15 | ns | **Note 6, 11** |
| Output disable time from $\overline{OE}$ high | $t_{OEZ}$ | 0 | 15 | 0 | 15 | ns | **Note 6, 11** |
| Output disable time from $\overline{LWE}$, $\overline{UWE}$ low | $t_{WEZ}$ | 0 | 15 | 0 | 15 | ns | **Note 6, 11** |
| Write command pulse width | $t_{WPZ}$ | 12 | | 12 | | ns | **Note 11** |
| Fast page mode cycle time | $t_{PC}$ | 35 | | 40 | | ns | |
| Fast page mode read modify write cycle time | $t_{PRWC}$ | 90 | | 95 | | ns | |
| $\overline{CAS}$ pulse width | $t_{CAS}$ | 15 | 100,000 | 15 | 100,000 | ns | |

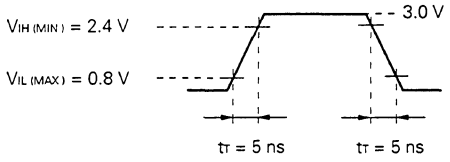
**(μPD482445 Only)**

| Parameter | Symbol | μPD482445-60 | | μPD482445-70 | | Unit | Condition |
|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | | |
| Output hold time from $\overline{CAS}$ | $t_{DHC}$ | 3 | | 5 | | ns | |
| Output disable time from $\overline{RAS}$ high | $t_{OFR}$ | 0 | 15 | 0 | 15 | ns | **Note 6, 12** |
| Output disable time from $\overline{CAS}$ high | $t_{OFC}$ | 0 | 15 | 0 | 15 | ns | **Note 6, 12** |
| Output disable time from $\overline{OE}$ high | $t_{OEZ}$ | 0 | 15 | 0 | 15 | ns | **Note 6, 12** |
| Output disable time from $\overline{LWE}$, $\overline{UWE}$ low | $t_{WEZ}$ | 0 | 15 | 0 | 15 | ns | **Note 6, 12** |
| Write command pulse width | $t_{WPZ}$ | 12 | | 12 | | ns | **Note 12** |
| Hyper page mode cycle time | $t_{HPC}$ | 30 | | 35 | | ns | |
| Hyper page mode read modify write cycle time | $t_{HPRWC}$ | 80 | | 90 | | ns | |
| $\overline{CAS}$ pulse width | $t_{HCAS}$ | 10 | 100,000 | 10 | 100,000 | ns | |

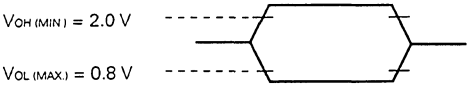
**Notes** 1. All applied voltages are referenced to GND.

2. After power up, wait more than 100 μs and then, execute eight $\overline{RAS}$ cycles as dummy cycles to initialize internal circuit.
3. Measure at $t_T$ = 5 ns
4. AC characteristic measuring conditions

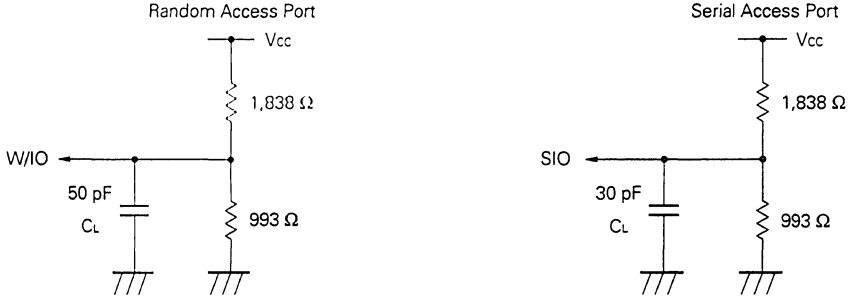
**(1) Input voltage, timing**




**(2) Output voltage determined**




**(3) Output load conditions**




5. For read cycle, access time is defined as follows:

| Input conditions | Access time | Access time from $\overline{RAS}$ |
|---|---|---|
| $t_{RAD} \leq t_{RAD}$ (MAX.) and $t_{RCD} \leq t_{RCD}$ (MAX.) | $t_{RAC}$ (MAX.) | $t_{RAC}$ (MAX.) |
| $t_{RAD} > t_{RAD}$ (MAX.) and $t_{RCD} \leq t_{RCD}$ (MAX.) | $t_{AA}$ (MAX.) | $t_{RAD} + t_{AA}$ (MAX.) |
| $t_{RCD} > t_{RCD}$ (MAX.) | $t_{CAC}$ (MAX.) | $t_{RCD} + t_{CAC}$ (MAX.) |

$t_{RAD}$ (MAX.) and $t_{RCD}$ (MAX.) are specified as reference points only; they are not restrictive operating parameters. They are used to determine which access time ($t_{RAC}$, $t_{AA}$, $t_{CAC}$) is to be used for finding out data will be available. Therefore, the input conditions $t_{RAD} \geq t_{RAD}$ (MAX.) and $t_{RCD} \geq t_{RCD}$ (MAX.) will not cause any operation problems.

6. $t_{SEZ}$, $t_{OEZ}$, $t_{WEZ}$, $t_{OFF}$, $t_{OFR}$, and $t_{OFC}$ define the time when the output achieves the condition of high impedance and is not referenced to $V_{OH}$ or $V_{OL}$.
7. Either $t_{RCH}$ (MIN.) or $t_{RRH}$ (MIN.) should be met in read cycles.
8. These parameters are referenced to the following points.
   (1) Early write cycle : The falling edge of $\overline{CAS}$
   (2) Late write cycle : The falling edge of $\overline{UWE}$, $\overline{LWE}$
   (3) Read modify write cycle : The falling edge of $\overline{UWE}$, $\overline{LWE}$
9. $t_{WCS} \geq t_{WCS}$ (MIN.) is the condition for early write cycle to be set. $D_{OUT}$ becomes high impedance during the cycle.
   $t_{RWD} \geq t_{RWD}$ (MIN.), $t_{CWD} \geq t_{CWD}$ (MIN.), $t_{AWD} \geq t_{AWD}$ (MIN.), are conditions for read modify write cycle to be set. The data of the selected address is output to $D_{OUT}$.
   If any of the above conditions are not met, pin W/IO will become undefined.

**Notes 10.** One of the following specifications will be valid depending on the type of read data transfer method used.

(1) $\overline{DT}/\overline{OE}$ edge control: Satisfy the following specifications.

- For $\overline{DT}/\overline{OE}$ edge inputs : $t_{RDH}$, $t_{CDH}$, $t_{ADD}$, $t_{DTR}$
- For SC inputs : $t_{SDD}$, $t_{SDH}$

(2) Self control: Satisfy the following specification.

- For $\overline{DT}/\overline{OE}$ edge inputs : $t_{RDHS}$
- For SC inputs : $t_{SSC}$, $t_{SDHR}$

**11.** Control pins $\overline{CAS}$, $\overline{OE}$, $\overline{UWE}$, $\overline{LWE}$ to set pin W/IO to high impedance. Because the timings at which $\overline{CAS}$ and $\overline{OE}$ are set to high level and $\overline{UWE}$ and $\overline{LWE}$ are set to low level affect the high impedance state, the specifications will change as follows.

(1) When $\overline{CAS}$ is set to high level at $\overline{OE}$ (low level) and $\overline{UWE}$ and $\overline{LWE}$ (high level) at the end of the read cycle: $t_{OFF}$ is valid

(2) When $\overline{UWE}$ and $\overline{LWE}$ are set to low level at $\overline{CAS}$ (low level) and $\overline{OE}$ (low level) at the end of the read cycle: $t_{WEZ}$ and $t_{WPZ}$ are valid

(3) When $\overline{OE}$ is set to high level at $\overline{CAS}$ (low level) and $\overline{UWE}$ and $\overline{LWE}$ (high level) at the end of the read cycle: $t_{OEZ}$ is valid

**12.** Control pins $\overline{RAS}$, $\overline{CAS}$, $\overline{OE}$, $\overline{UWE}$, $\overline{LWE}$ to set pin W/IO to high impedance. Because the timings at which $\overline{RAS}$, $\overline{CAS}$, and $\overline{OE}$ are set to high level and $\overline{UWE}$ and $\overline{LWE}$ are set to low level affect the high impedance state, the specifications will change as follows.

When controlling $\overline{RAS}$ and $\overline{CAS}$, the output cannot be made high impedance unless both pins are set to high. There is difference between $t_{OFC}$ and $t_{OFR}$, because $\overline{RAS}$ and $\overline{CAS}$ control is specified from the rising edge of the slower one.

(1) When $\overline{RAS}$ is set to high level after $\overline{CAS}$ is set to high level at $\overline{OE}$ (low level) and $\overline{UWE}$ and $\overline{LWE}$ (high level) at the end of the read cycle: $t_{OFR}$ is valid

(2) When $\overline{CAS}$ is set to high level after $\overline{RAS}$ is set to high level at $\overline{OE}$ (low level) and $\overline{UWE}$ and $\overline{LWE}$ (high level) at the end of the read cycle: $t_{OFC}$ is valid

(3) When $\overline{UWE}$ and $\overline{LWE}$ are set to low level at $\overline{RAS}$, $\overline{CAS}$ (low level) and $\overline{OE}$ (low level) at the end of the read cycle: $t_{WEZ}$ and $t_{WPZ}$ are valid

(4) When $\overline{OE}$ is set to high level at $\overline{RAS}$, $\overline{CAS}$ (low level) and $\overline{UWE}$ and $\overline{LWE}$ (high level) at the end of the read cycle: $t_{OEZ}$ is valid

**13.** The $t_{SRS}$ and $t_{SRH}$ in the hidden refresh cycle, $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (STOP register set cycle and optional reset cycle) are specified to guarantee the serial port operations until the transfer cycle is executed after the STOP register value is changed. When the STOP register value is not to be changed, or when the binary boundary jump function is not used (when the TAP register is empty), $t_{SRS}$ and $t_{SRH}$ will not be specified.

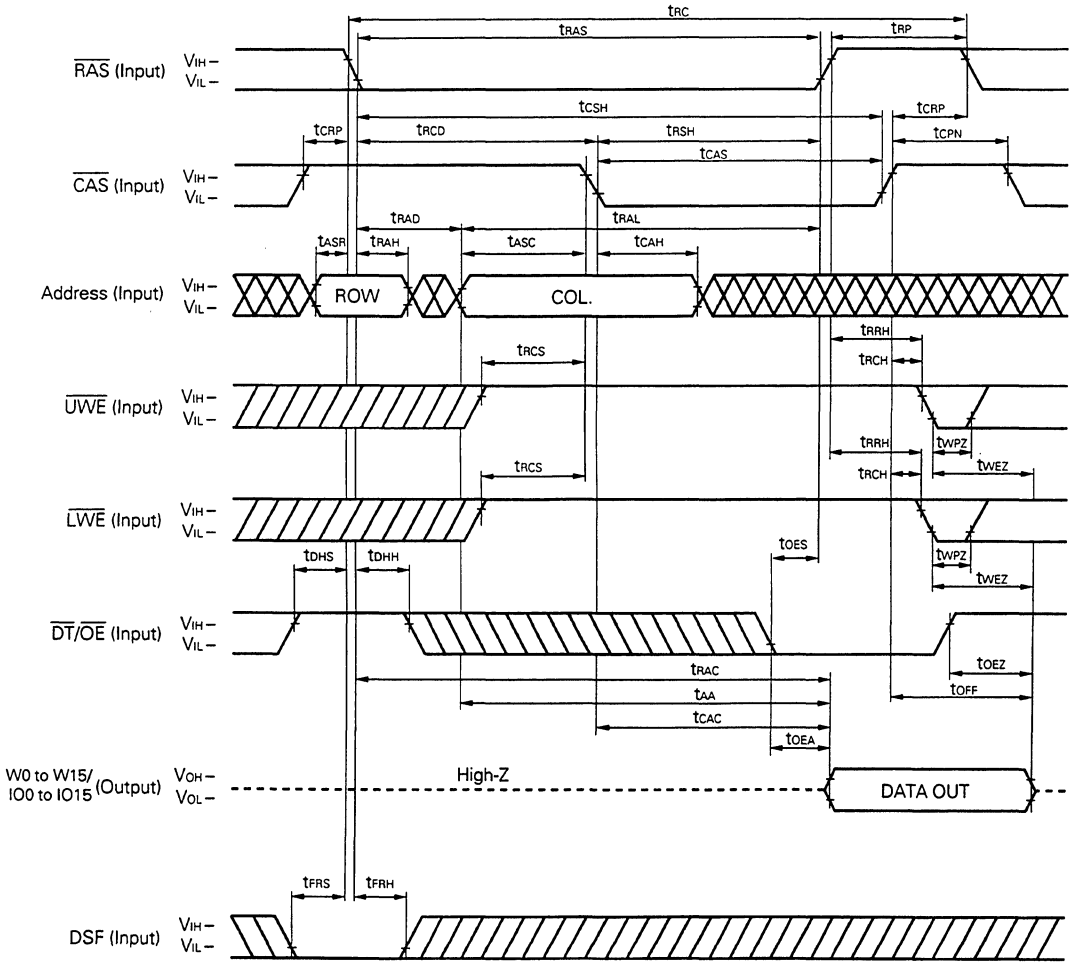
**14.** $t_{SSC}$ (split read data transfer cycle) and $t_{SRS}$ (split write data transfer cycle) are specified at the rising edge of SC which reads/writes the address of the jump source in the binary boundary jump function. $t_{SDHR}$ (split read data transfer cycle and split write data transfer cycle) is specified at the rising edge of SC which reads/writes the address of the jump destination in the binary boundary jump function. The rising edge of these SCs cannot be input in periods (1) and (2).

(1) Split read data transfer cycle: Period from the rising edge of the SC specifying $t_{SSC}$ to that of the SC specifying $t_{SDHR}$ (Refer to **Note 2 Split Read/Write Data Transfer Cycle Timing Chart.**)

(2) Split write data transfer cycle: Period from the rising edge of the SC specifying $t_{SRS}$ to that of the SC specifying $t_{SDHR}$ (Refer to **Note 2 Split Read/Write Data Transfer Cycle Timing Chart.**)

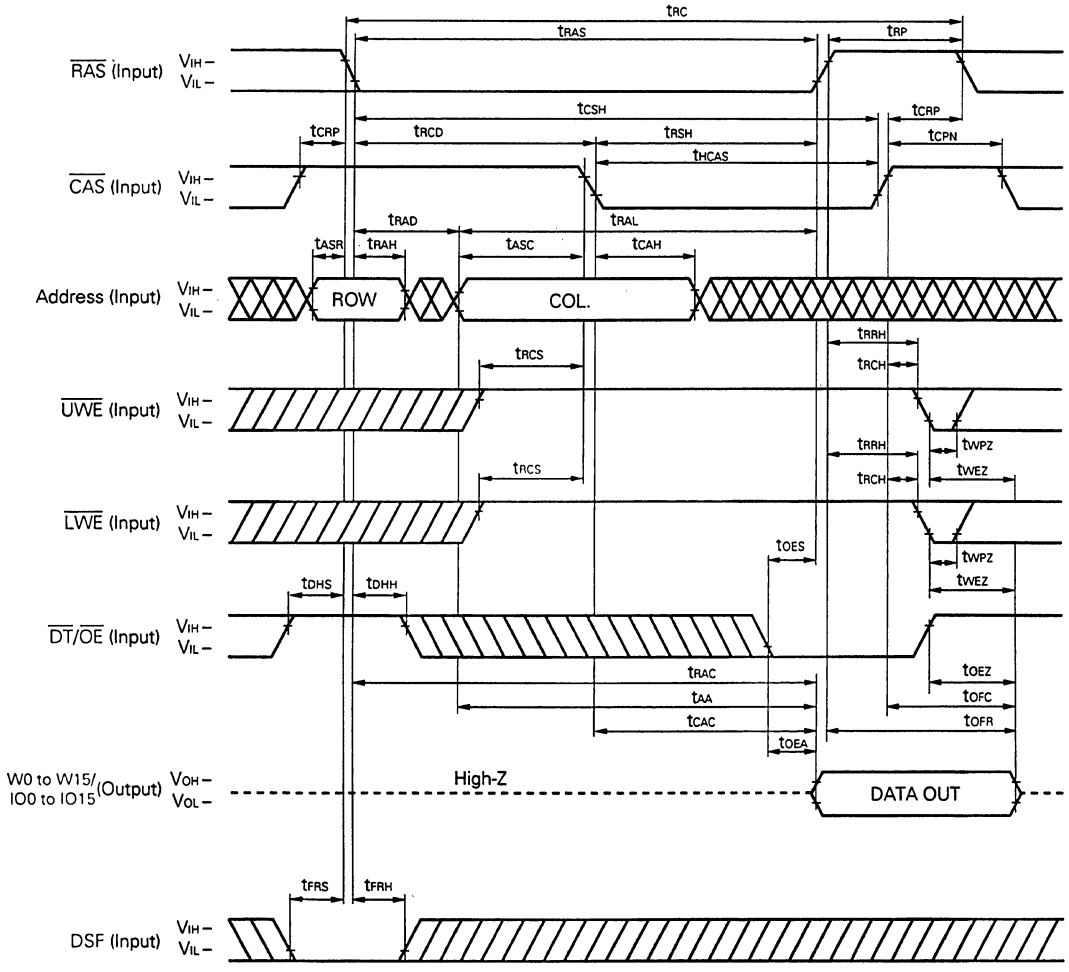
**15.** Limitations of split read/write data transfer cycle during serial write operations. When split read/write data transfer is performed while serial write is executed for the column specified by the STOP register, serial write operations cannot be guaranteed.

**Read Cycle (μPD482444)**



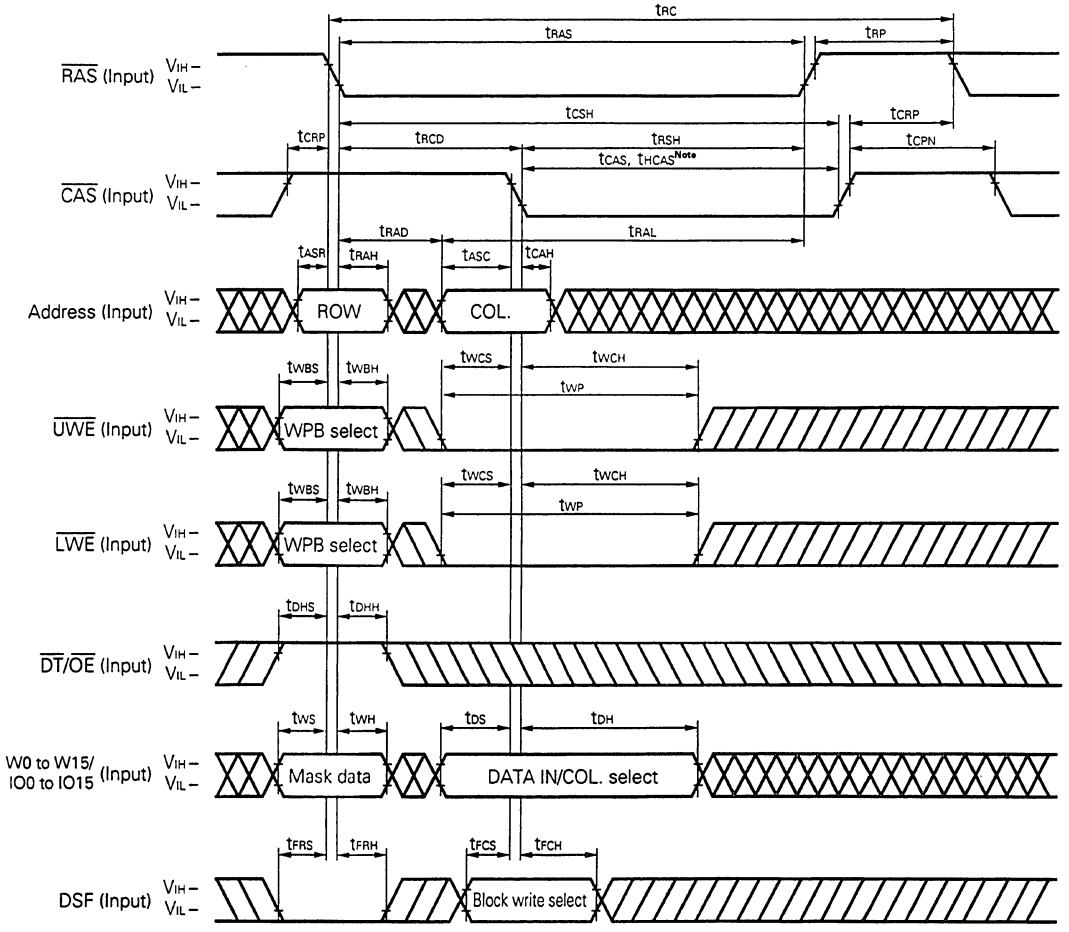

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Read Cycle (Extended data output: μPD482445)**




**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

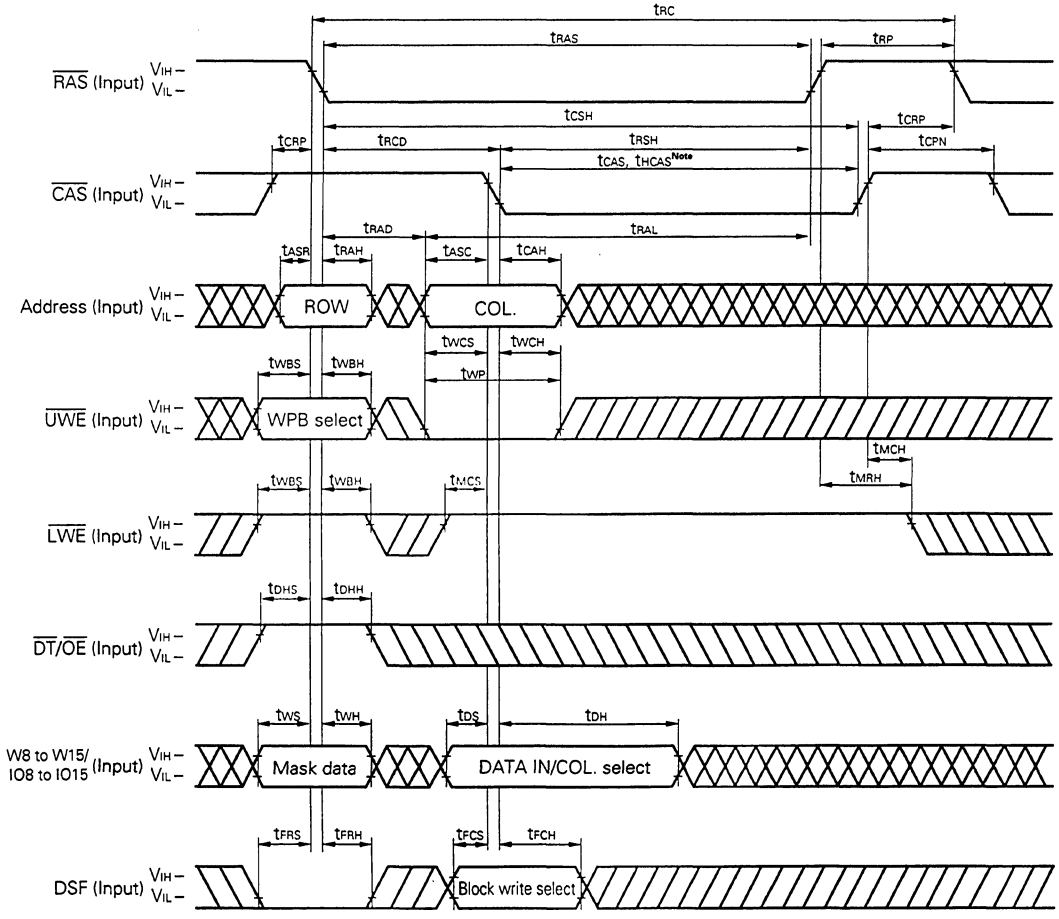
**Early Write Cycle/Early Block Write Cycle**




**Note** tCAS for the μPD482444
tHCAS for the μPD482445

**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

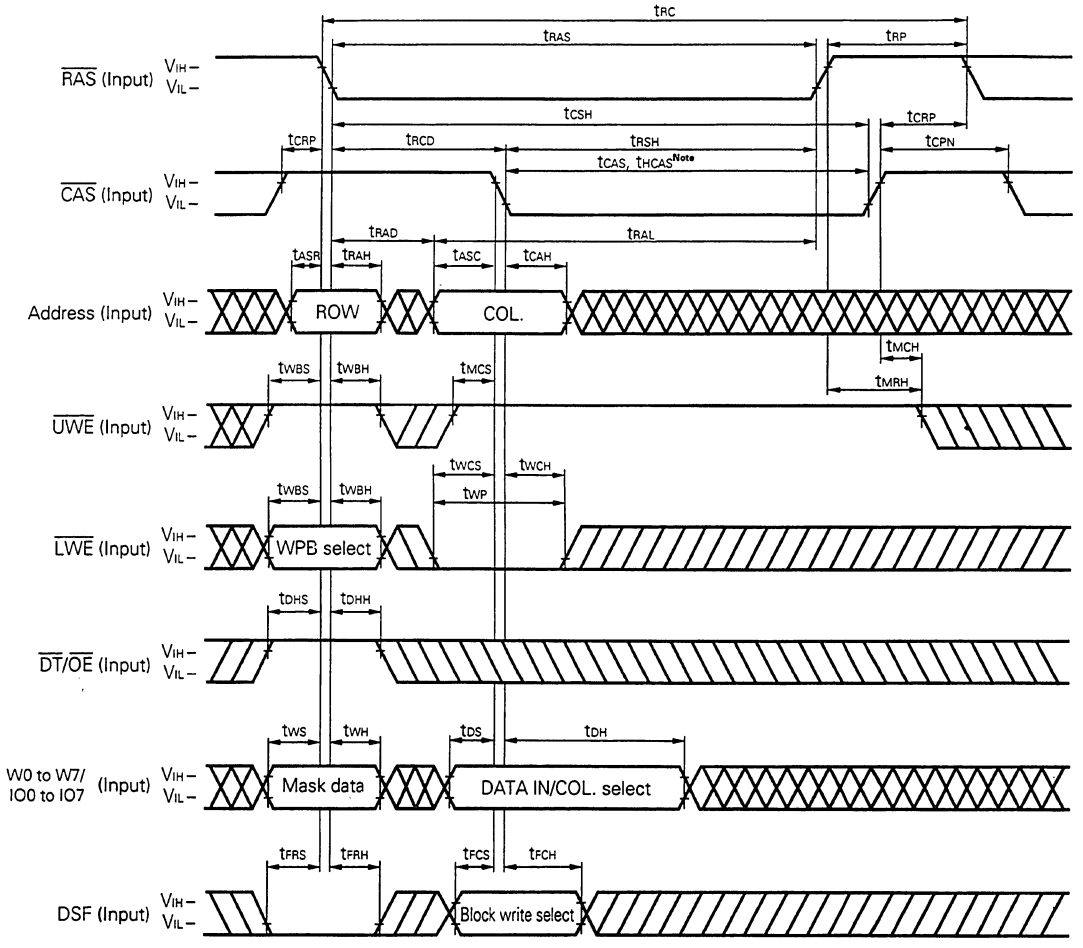
### Upper Byte Early Write Cycle/Upper Byte Early Block Write Cycle




**Note** $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

**Remarks** 1. W0 to W7/IO0 to IO7 : Don't care
2. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
3. WPB : Write-per-bit
4. When block write cycle is selected, input the column selection data to DATA IN.
5. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

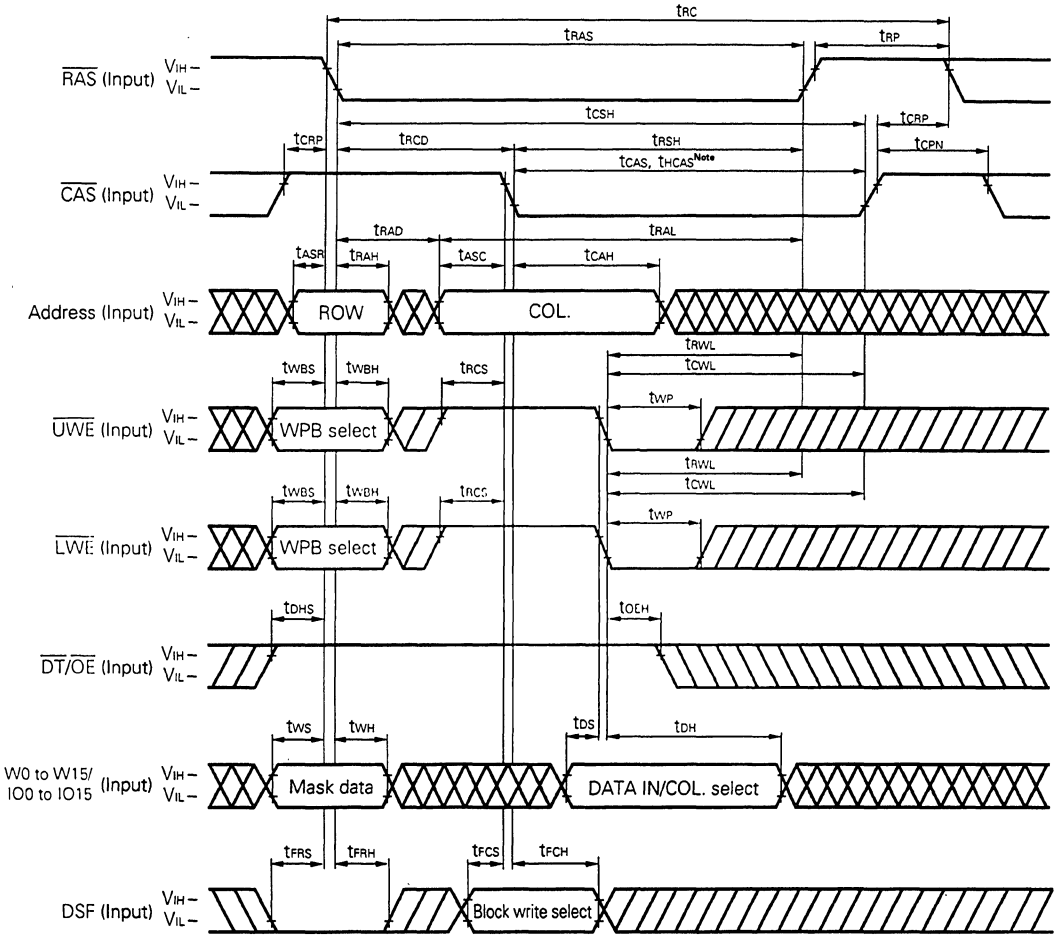
**Lower Byte Early Write Cycle/Lower Byte Early Block Write Cycle**




**Note** tCAS for the μPD482444
tHCAS for the μPD482445

**Remarks**
1. W8 to W15/IO8 to IO15 : Don't care
2. When DSF is high level : Block write cycle
   When DSF is low level : Write cycle
3. WPB : Write-per-bit
4. When block write cycle is selected, input the column selection data to DATA IN.
5. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

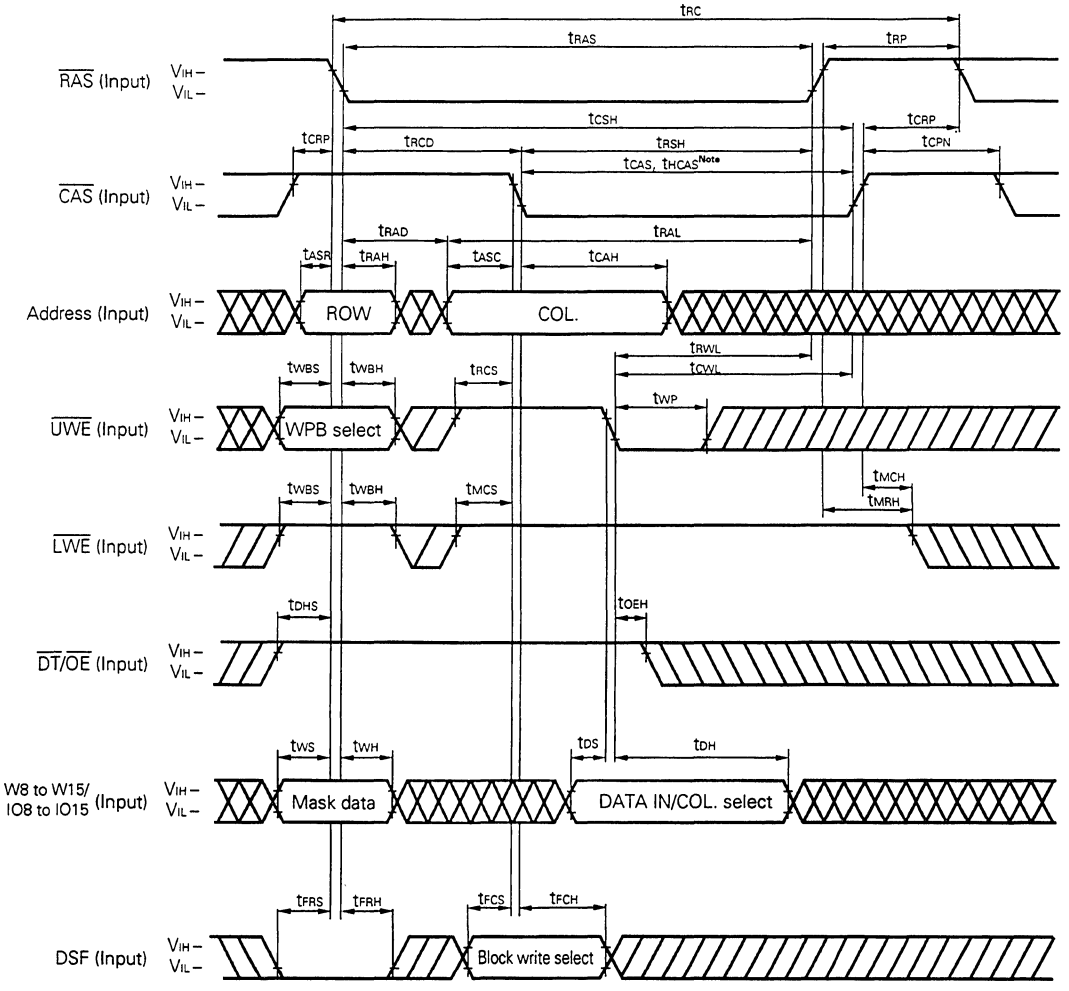
**Late Write Cycle/Late Block Write Cycle**




**Note** tCAS for the μPD482444
tHCAS for the μPD482445

**Remarks** **1.** When DSF is high level : Block write cycle
When DSF is low level : Write cycle
**2.** WPB : Write-per-bit
**3.** When block write cycle is selected, input the column selection data to DATA IN.
**4.** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

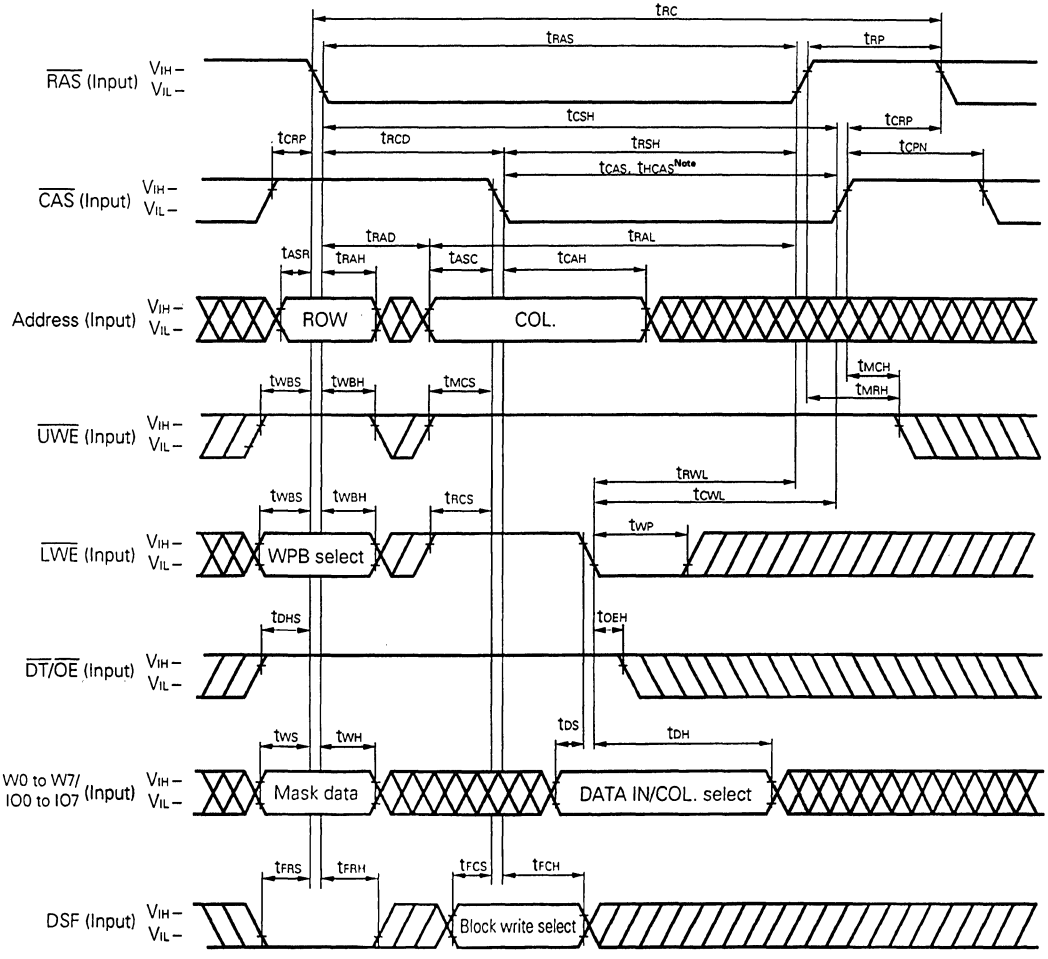
**Upper Byte Late Write Cycle/Upper Byte Late Block Write Cycle**




**Note** tCAS for the μPD482444
tHCAS for the μPD482445

**Remarks** 1. W0 to W7/IO0 to IO7 : Don't care
2. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
3. WPB : Write-per-bit
4. When block write cycle is selected, input the column selection data to DATA IN.
5. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

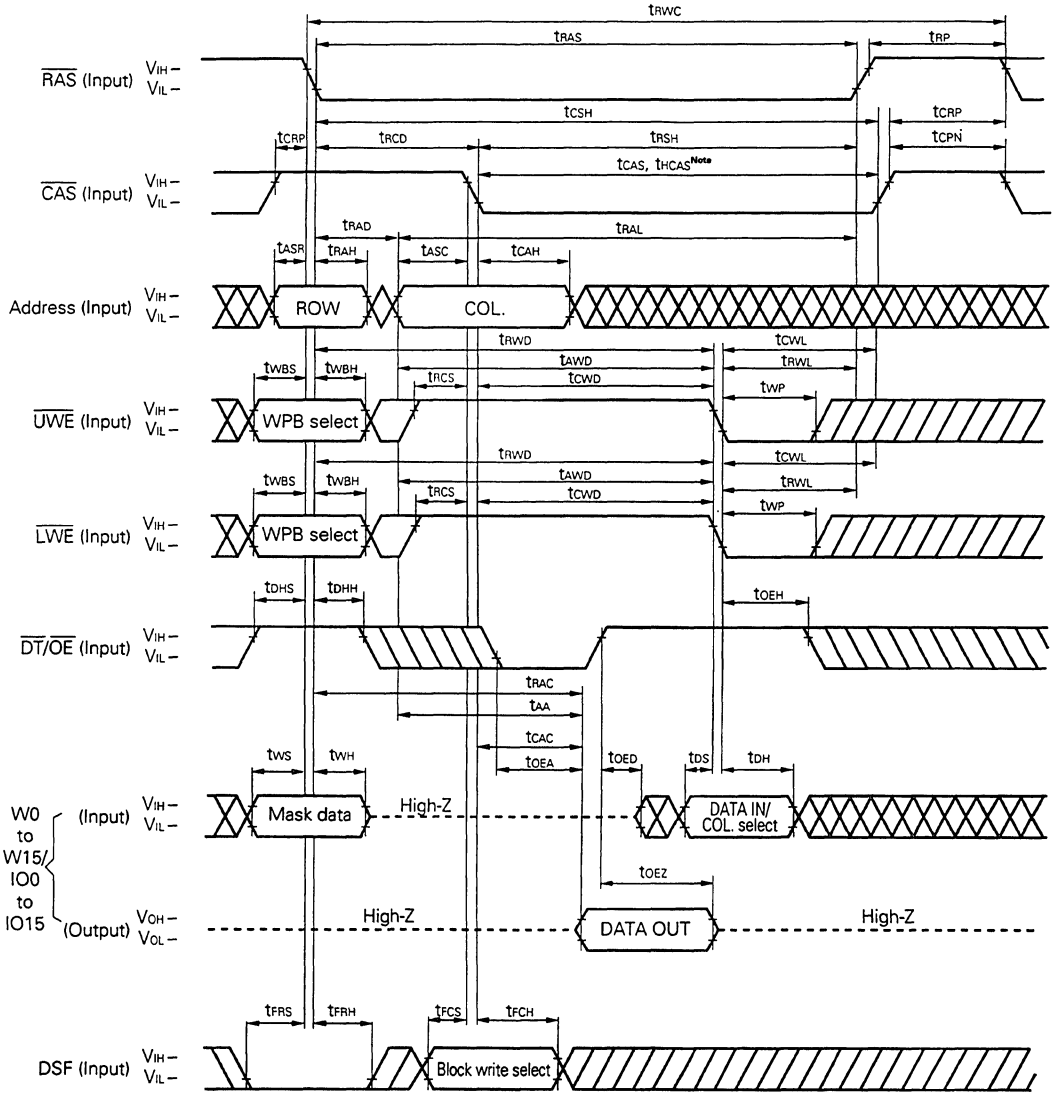
**Lower Byte Late Write Cycle/Lower Byte Late Block Write Cycle**




**Note** tCAS for the μPD482444
tHCAS for the μPD482445

**Remarks** 1. W8 to W15/IO8 to IO15 : Don't care
2. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
3. WPB : Write-per-bit
4. When block write cycle is selected, input the column selection data to DATA IN.
5. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

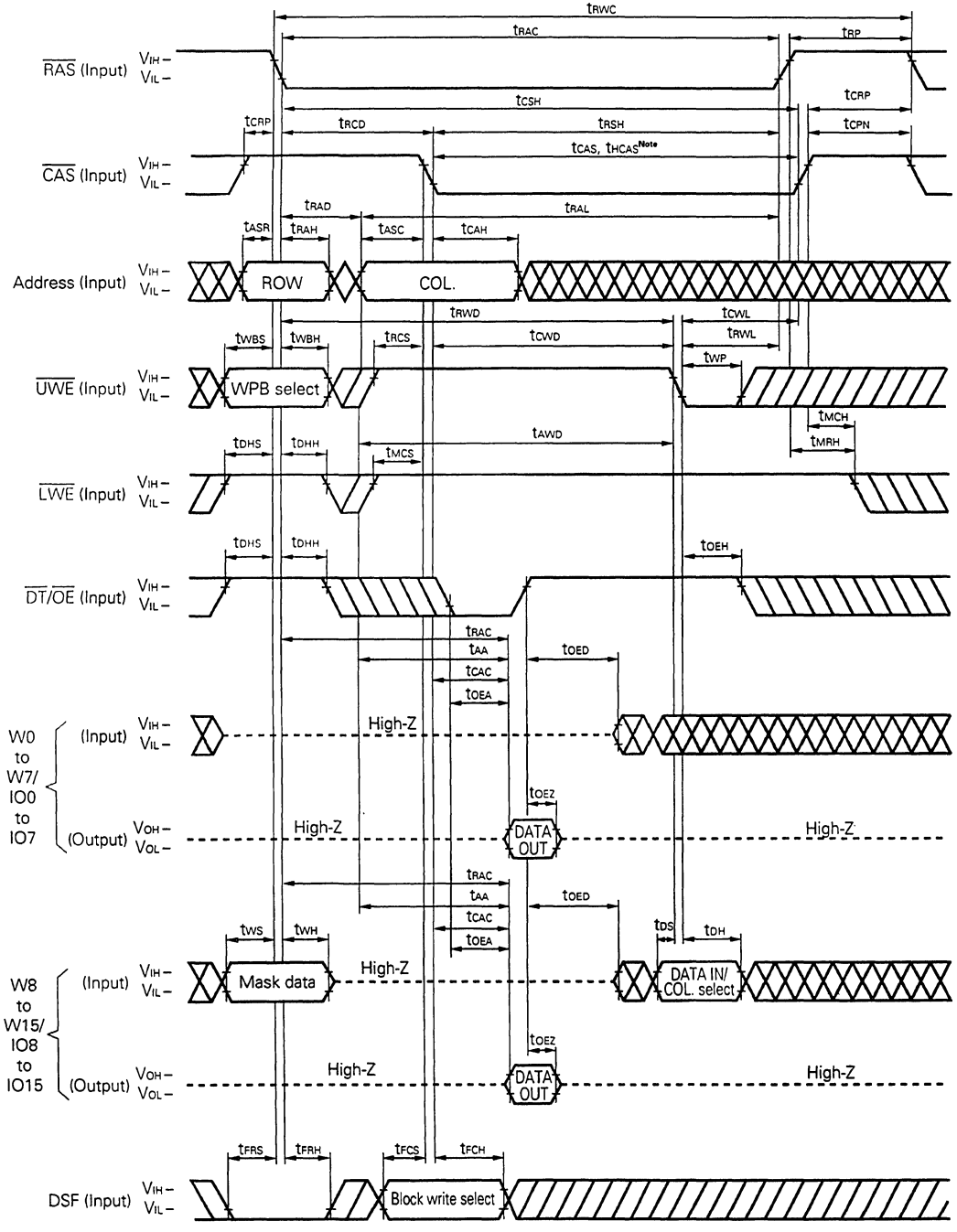
**Read Modify Write Cycle/Read Modify Block Write Cycle**




**Note** $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

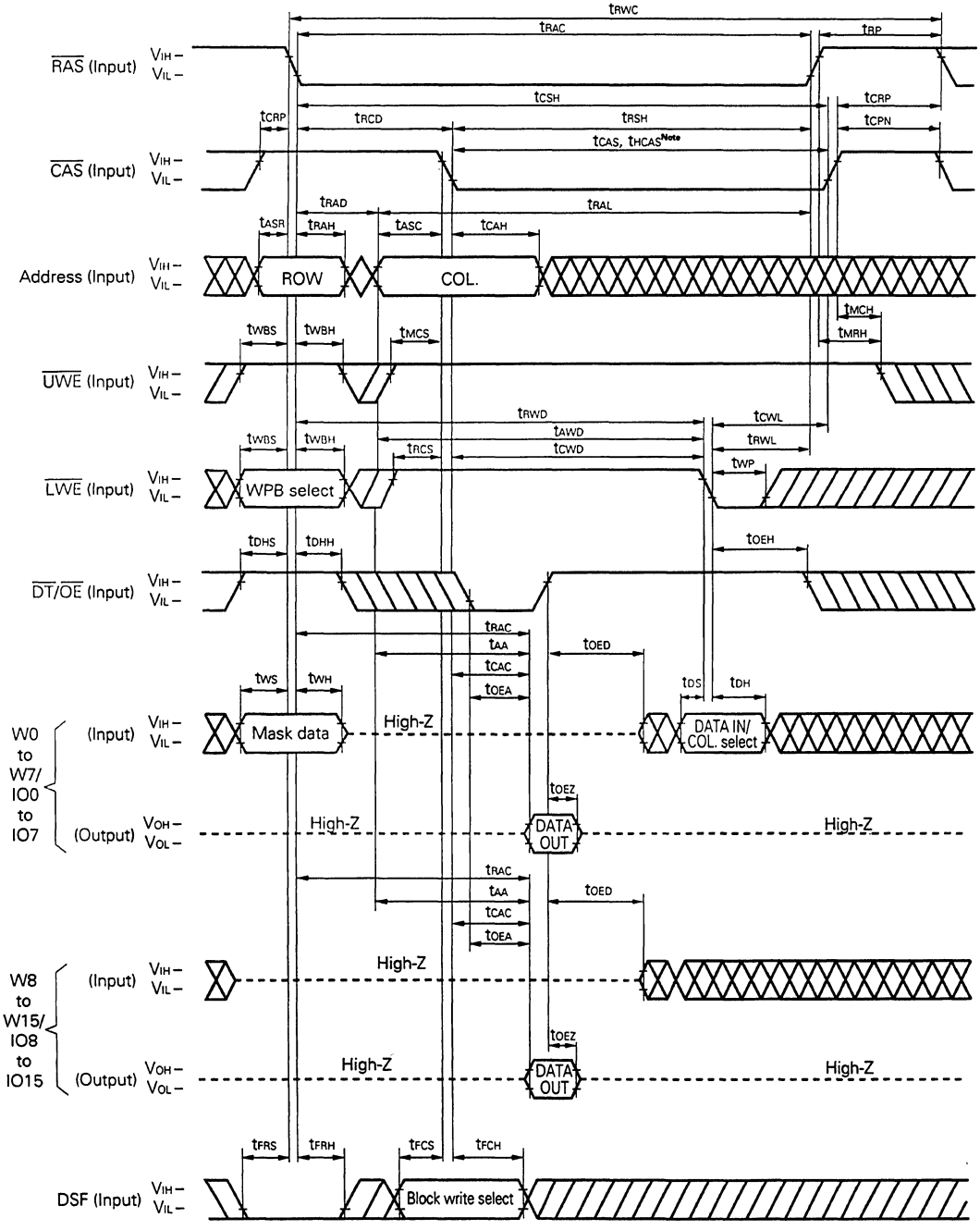
**Read Modify Upper Byte Write Cycle/Read Modify Upper Byte Block Write Cycle**



**Note** $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

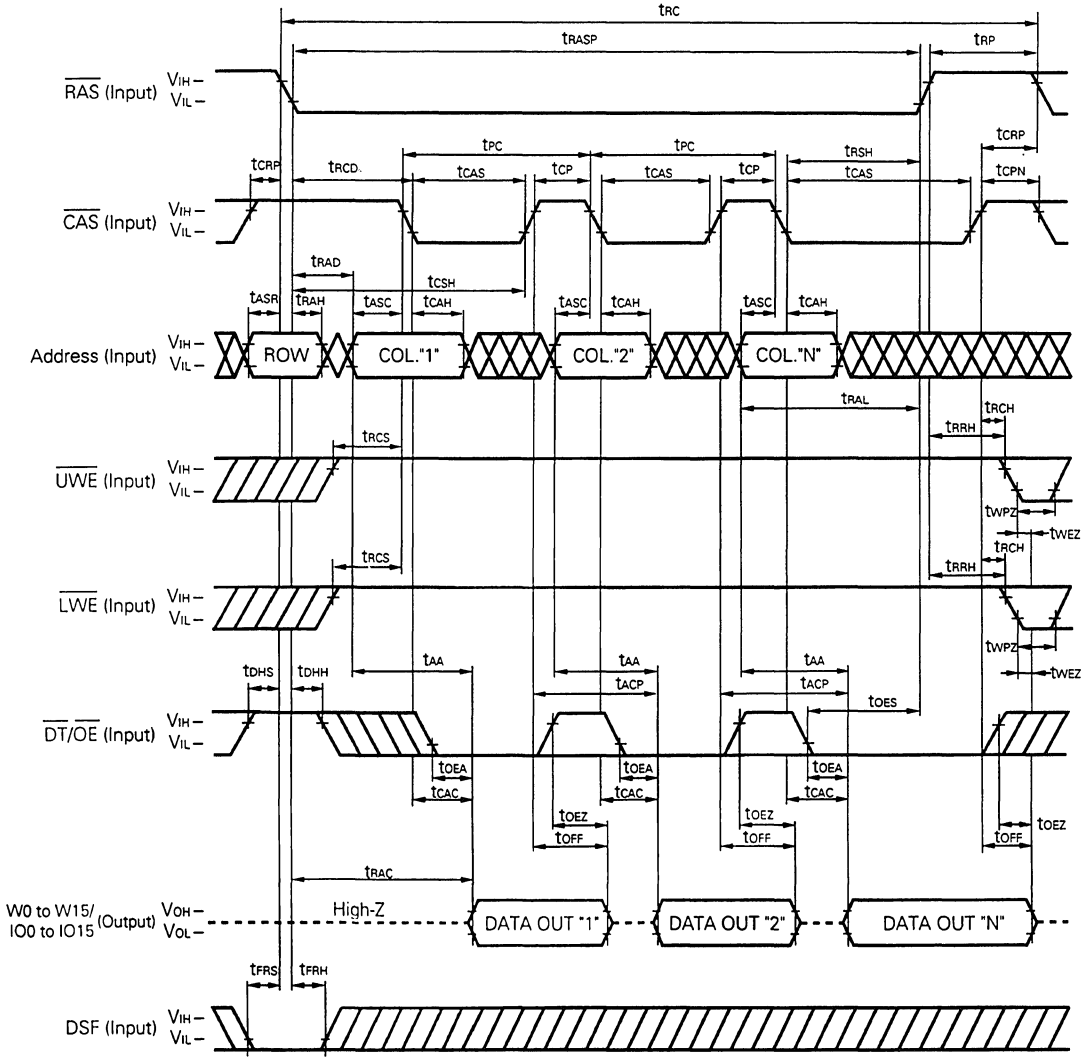
**Read Modify Lower Byte Write Cycle/Read Modify Lower Byte Block Write Cycle**



**Note** $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

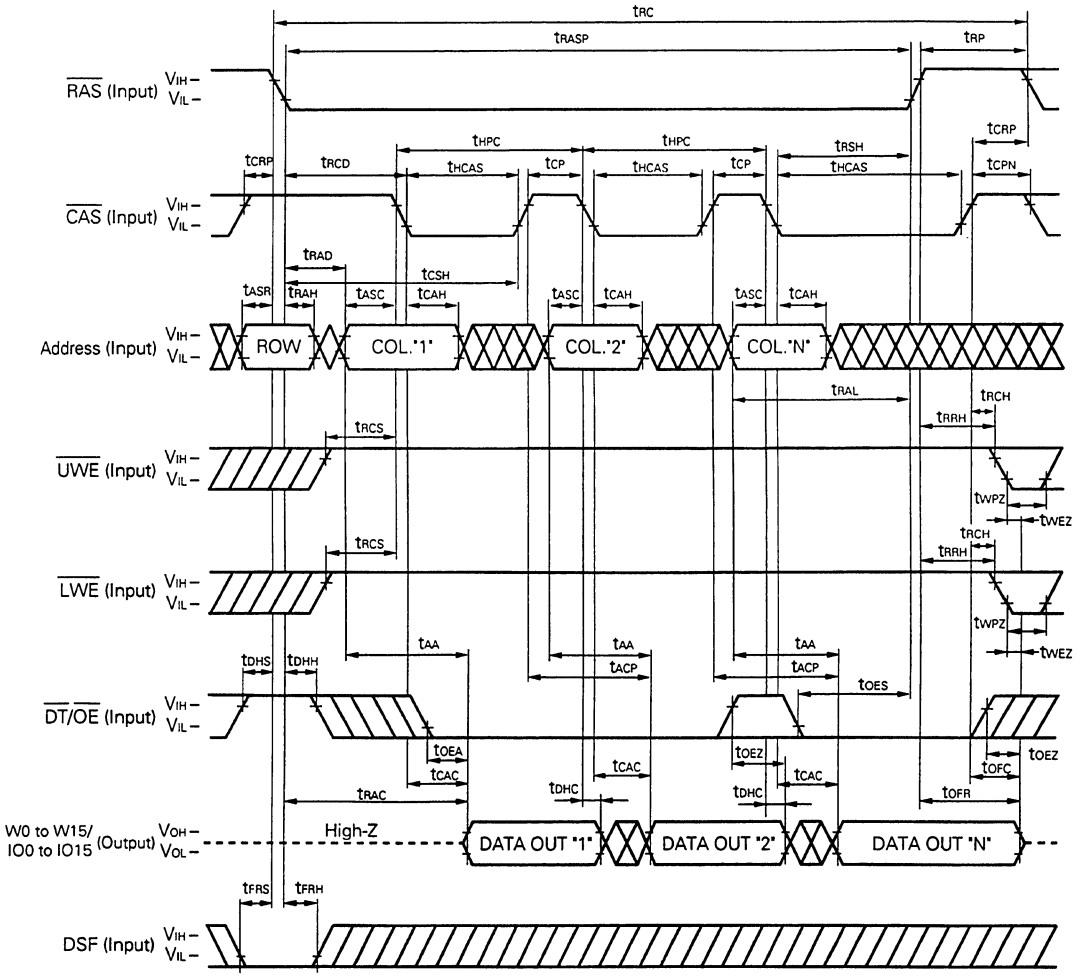
**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Fast Page Mode Read Cycle (μPD482444)**



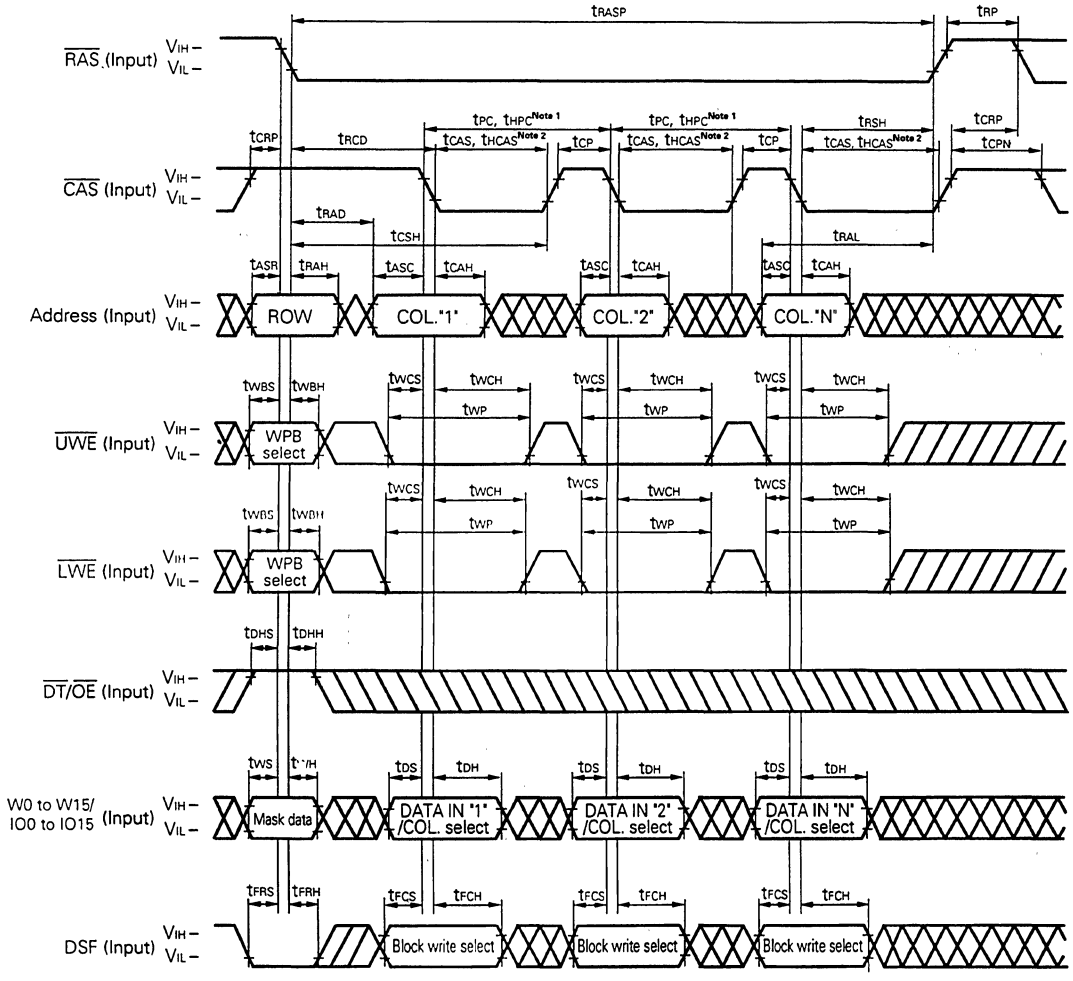

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Hyper Page Mode Read Cycle (Extended data output: μPD482445)**




**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

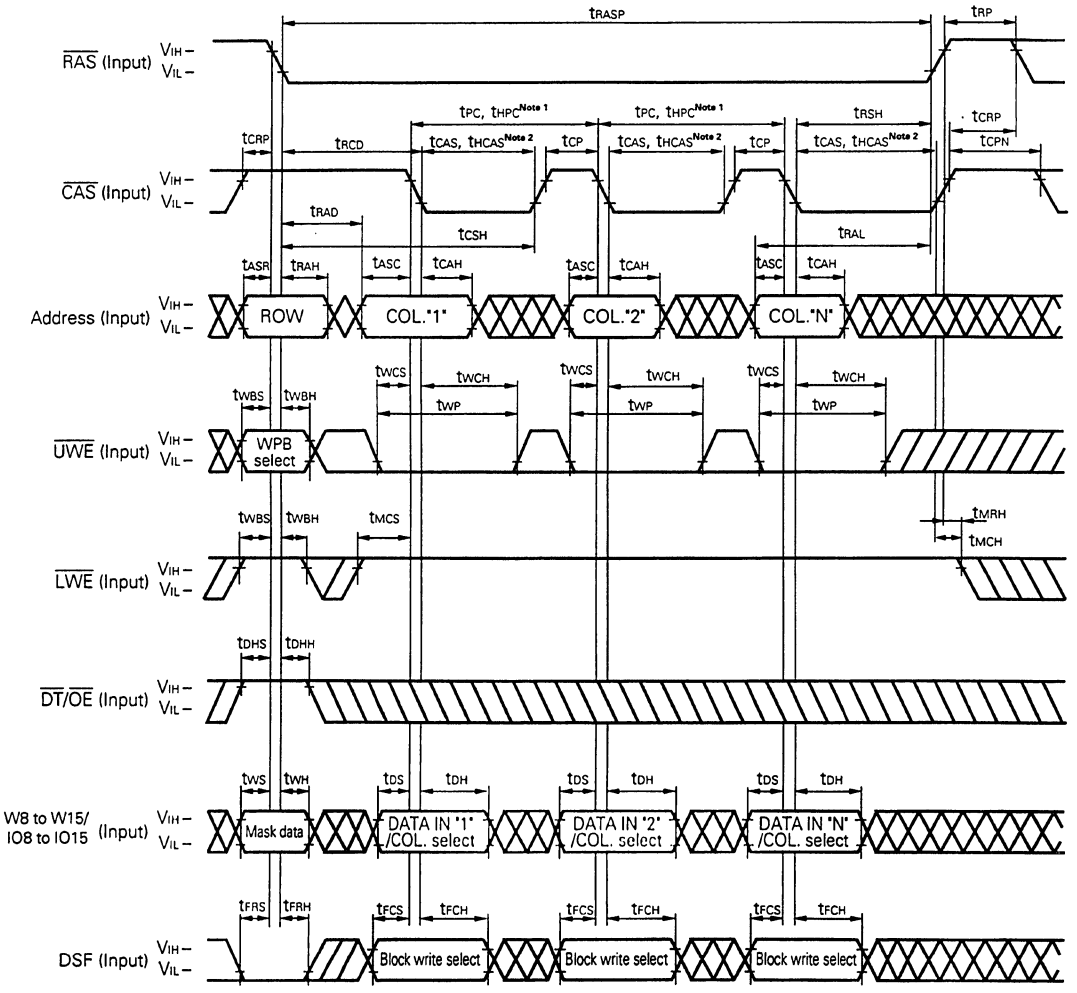
**Fast Page, Hyper Page Mode Early Write Cycle/Fast Page, Hyper Page Mode Early Block Write Cycle**




**Notes** 1. tPC for the μPD482444
tHPC for the μPD482445
2. tCAS for the μPD482444
tHCAS for the μPD482445

**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

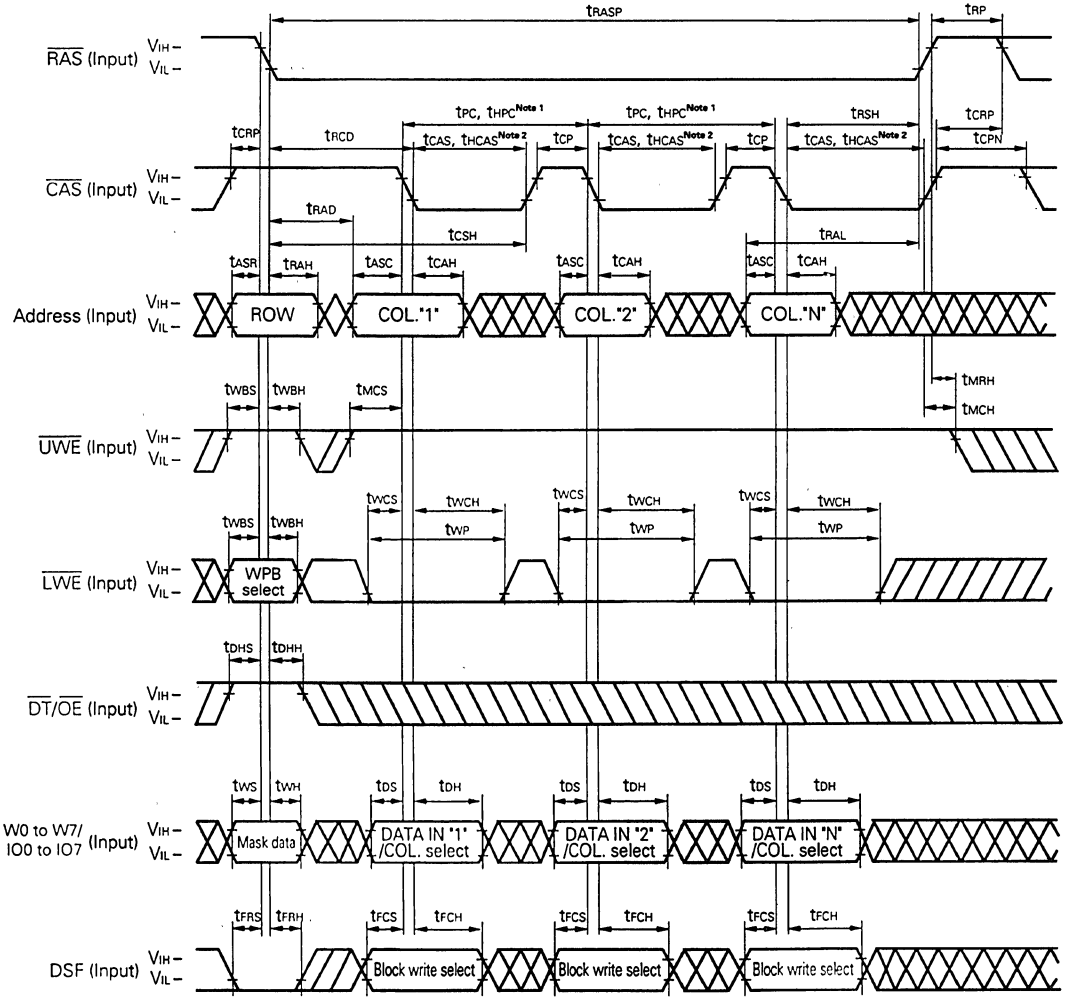
**Fast Page, Hyper Page Mode Upper Byte Early Write Cycle/**
**Fast Page, Hyper Page Mode Upper Byte Early Block Write Cycle**




**Notes 1.** tPC for the μPD482444
tHPC for the μPD482445
**2.** tCAS for the μPD482444
tHCAS for the μPD482445

**Remarks 1.** W0 to W7/IO0 to IO7 : Don't care
**2.** When DSF is high level : Block write cycle
When DSF is low level : Write cycle
**3.** WPB : Write-per-bit
**4.** When block write cycle is selected, input the column selection data to DATA IN.
**5.** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

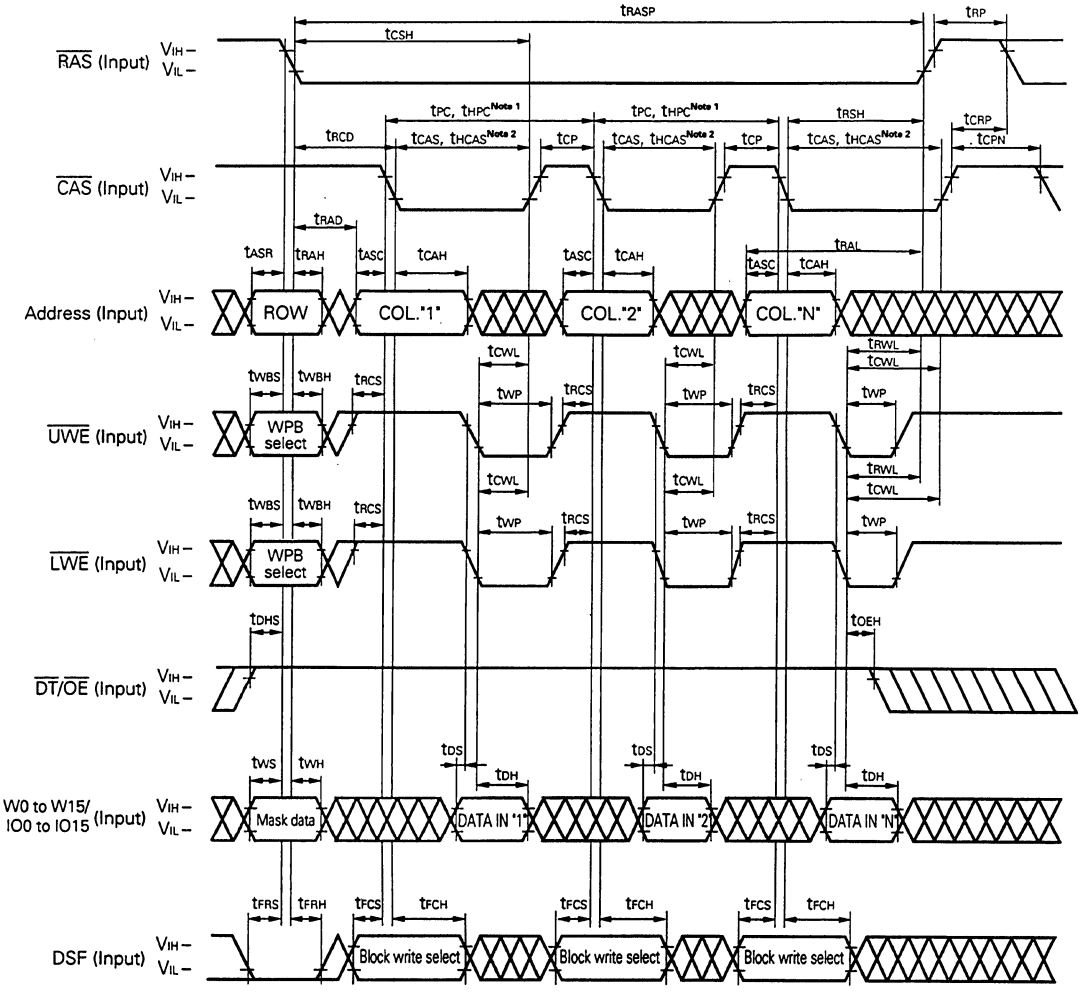
**Fast Page, Hyper Page Mode Lower Byte Early Write Cycle/**
**Fast Page, Hyper Page Mode Lower Byte Early Block Write Cycle**




**Notes** 1. $t_{PC}$ for the μPD482444
$t_{HPC}$ for the μPD482445
2. $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

**Remarks** 1. W8 to W15/IO8 to IO15 : Don't care
2. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
3. WPB : Write-per-bit
4. When block write cycle is selected, input the column selection data to DATA IN.
5. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Fast Page, Hyper Page Mode Late Write Cycle/Fast Page, Hyper Page Mode Late Block Write Cycle**




**Notes** 1. $t_{PC}$ for the μPD482444
$t_{HPC}$ for the μPD482445

2. $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

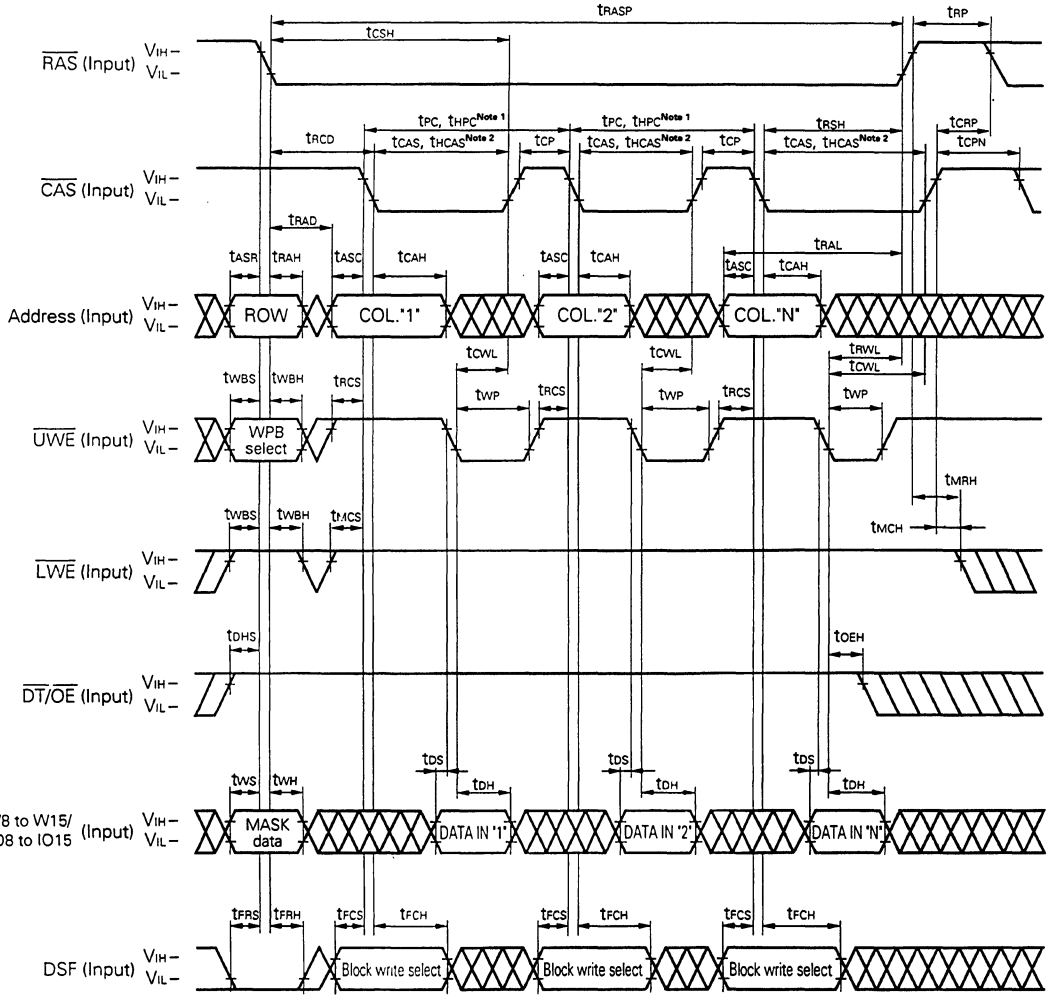
**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle

2. WPB : Write-per-bit

3. When block write cycle is selected, input the column selection data to DATA IN.

4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

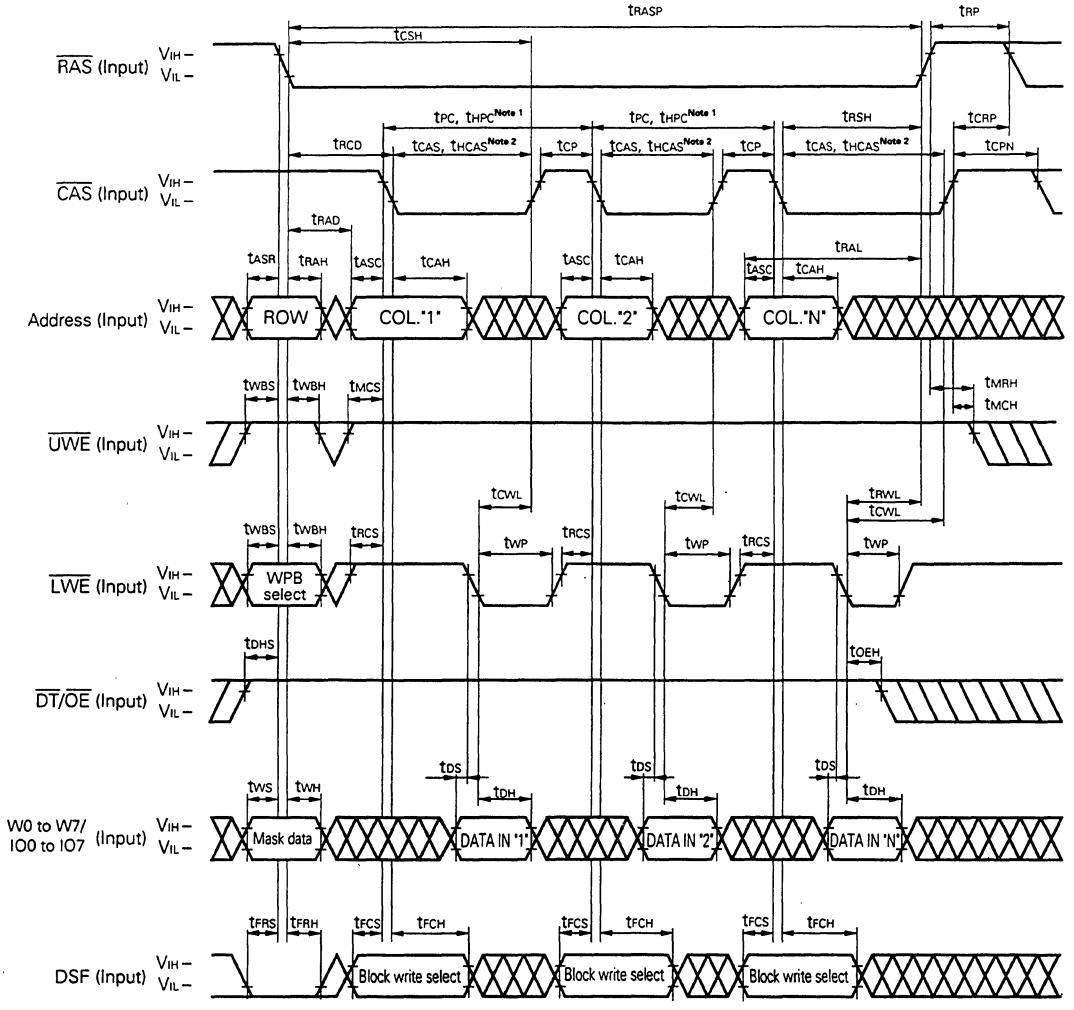
**Fast Page, Hyper Page Mode Upper Byte Late Write Cycle/**
**Fast Page, Hyper Page Mode Upper Byte Late Block Write Cycle**




**Notes** 1. $t_{PC}$ for the μPD482444
$t_{HPC}$ for the μPD482445
2. $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

**Remarks** 1. W0 to W7/IO0 to IO7 : Don't care
2. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
3. WPB : Write-per-bit
4. When block write cycle is selected, input the column selection data to DATA IN.
5. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

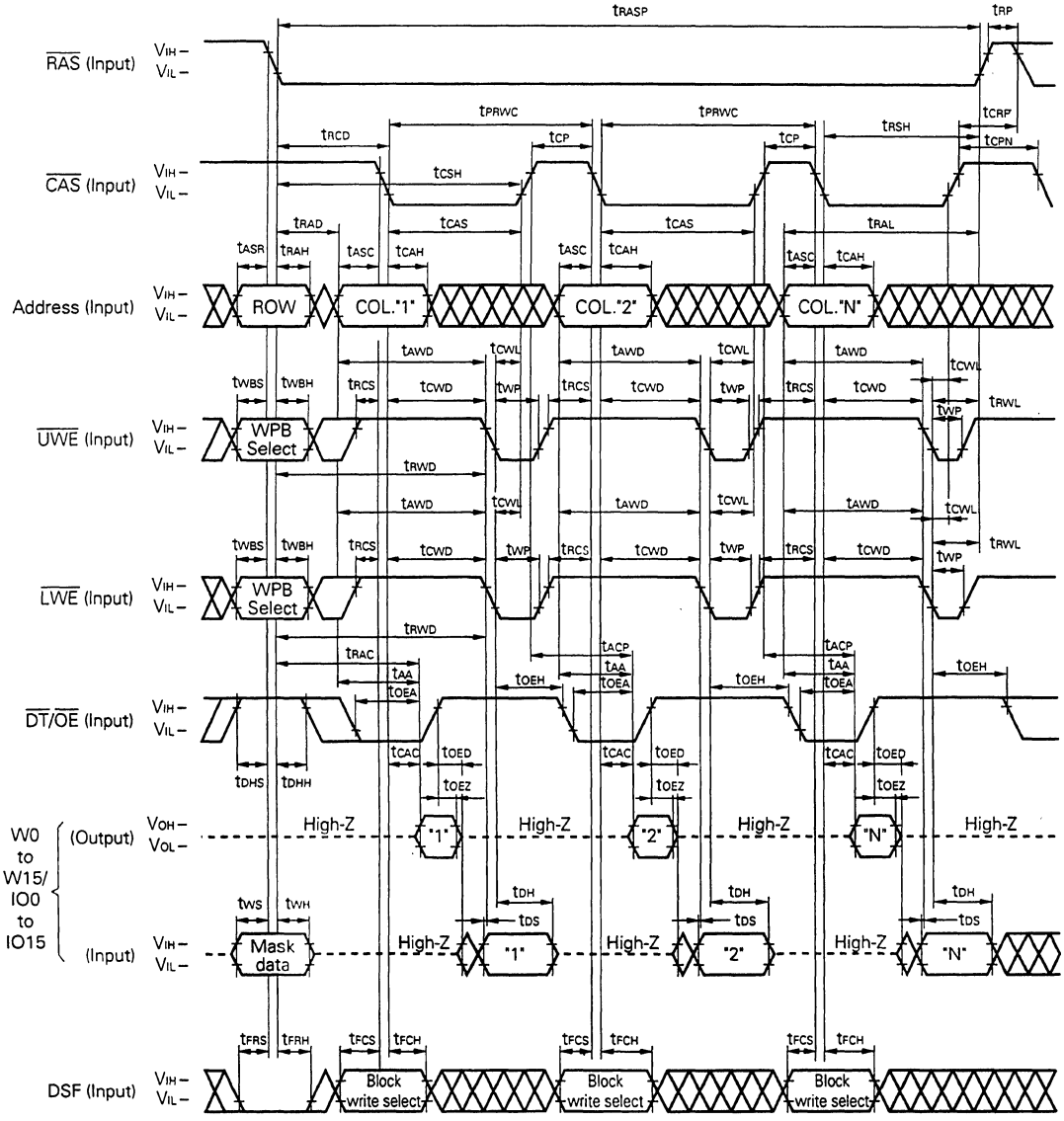
**Fast Page, Hyper Page Mode Lower Byte Late Write Cycle/**
**Fast Page, Hyper Page Mode Lower Byte Late Block Write Cycle**




**Notes** 1. tPC for the μPD482444
tHPC for the μPD482445
2. tCAS for the μPD482444
tHCAS for the μPD482445

**Remarks** 1. W8 to W15/IO8 to IO15 : Don't care
2. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
3. WPB : Write-per-bit
4. When block write cycle is selected, input the column selection data to DATA IN.
5. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

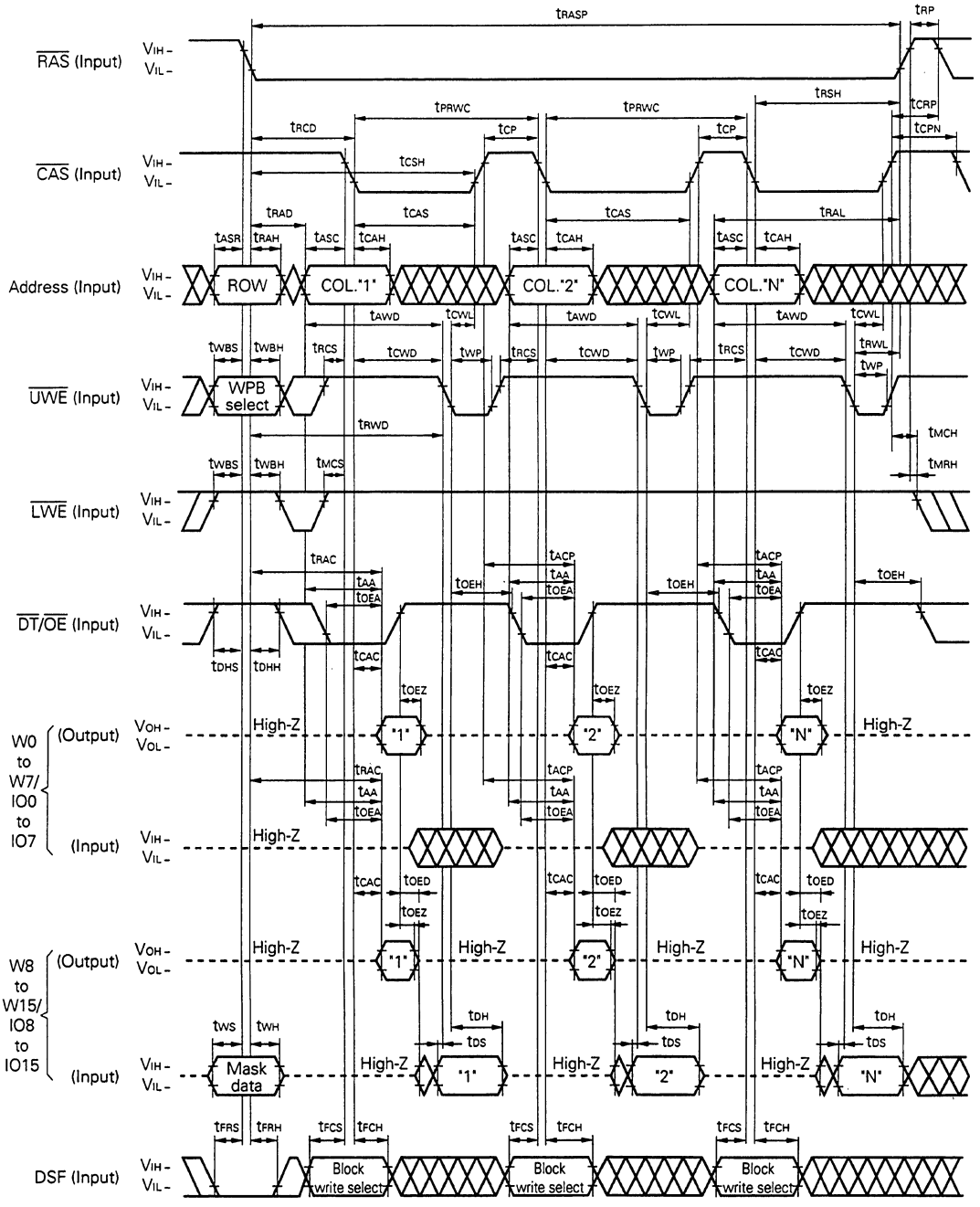
**Fast Page Mode Read Modify Write Cycle (μPD482444)/**
**Fast Page Mode Read Modify Block Write Cycle (μPD482444)**




**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle

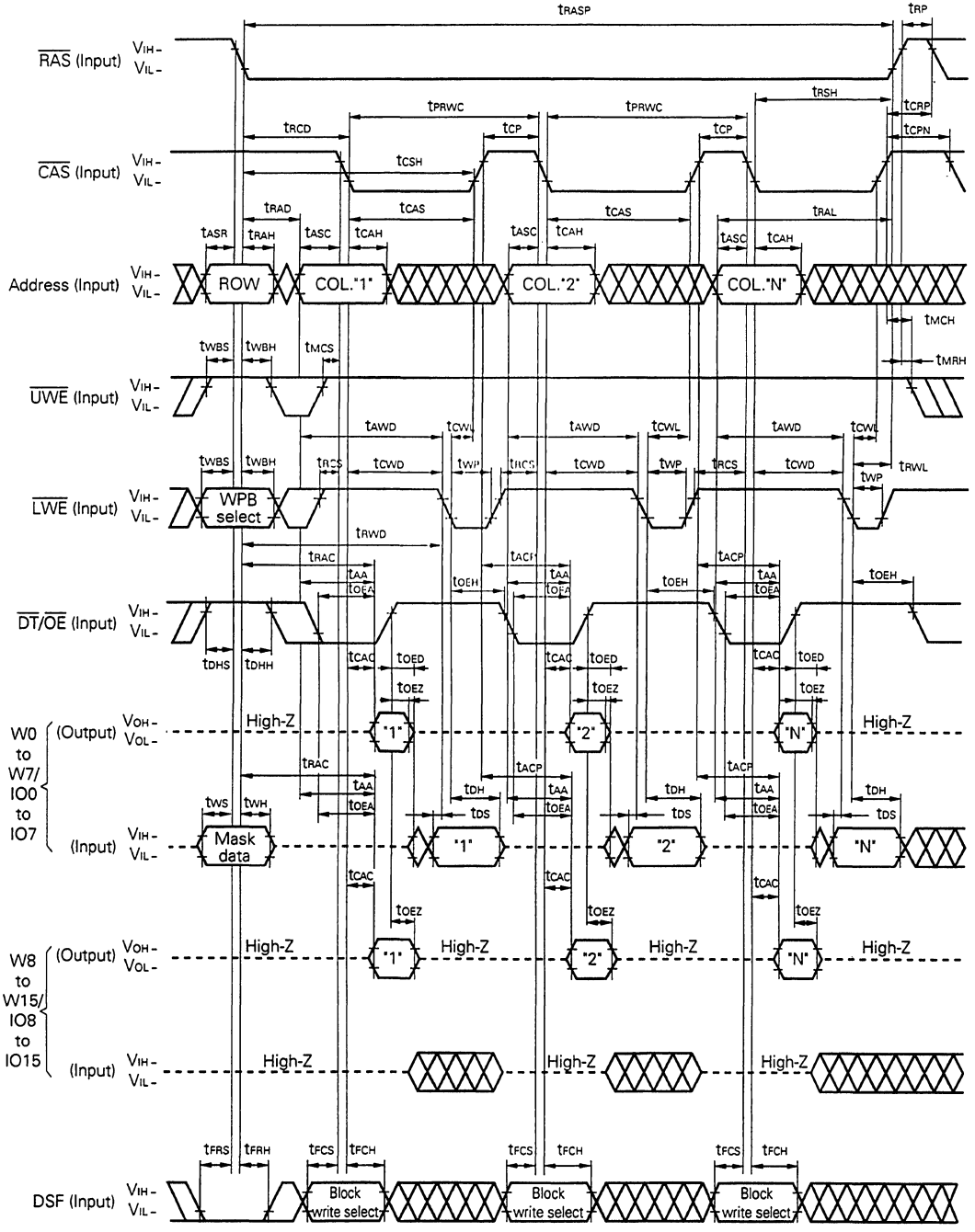
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Fast Page Mode Read Modify Upper Byte Write Cycle (μPD482444)/**
**Fast Page Mode Read Modify Upper Byte Block Write Cycle (μPD482444)**



**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

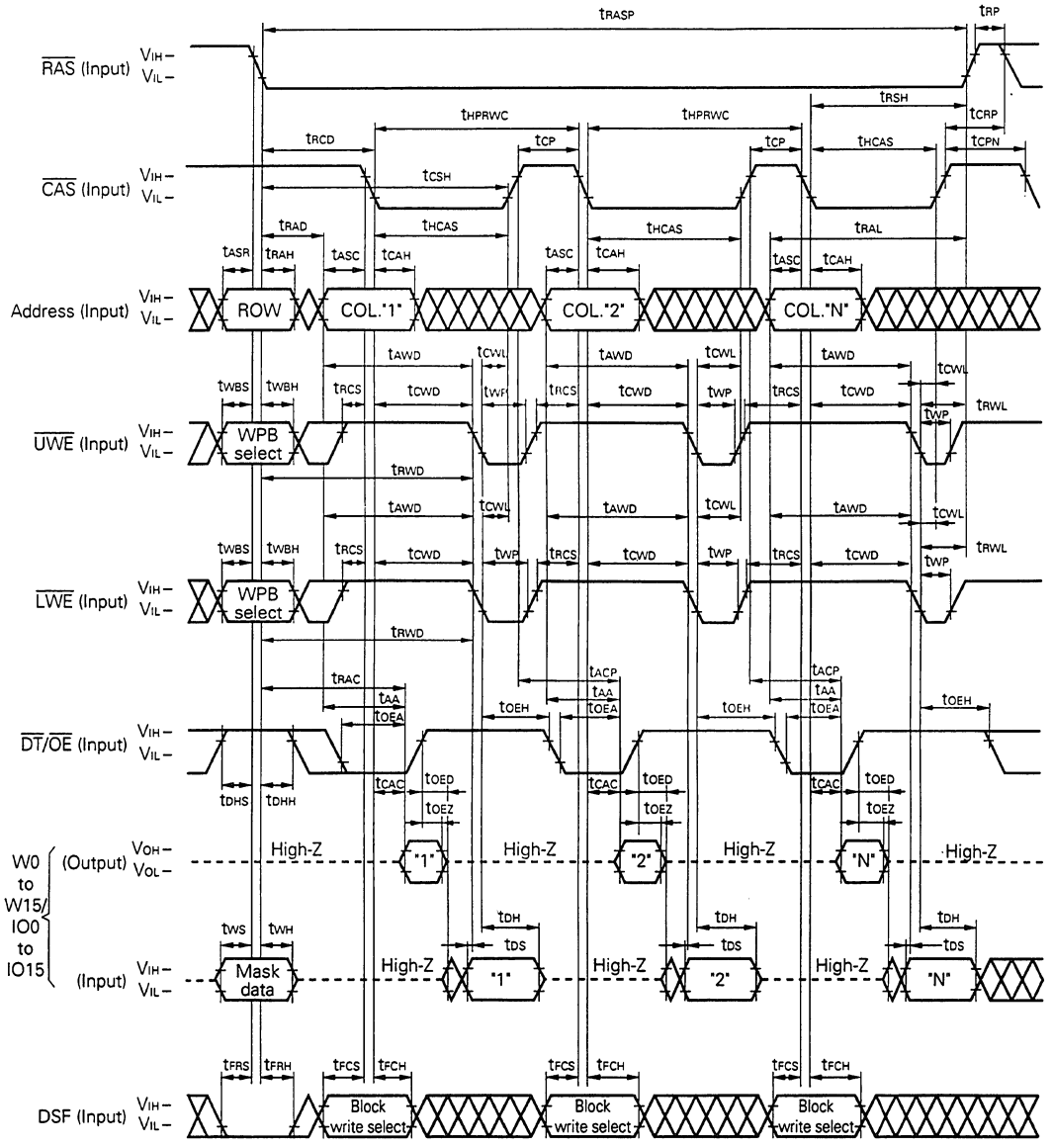
**Fast Page Mode Read Modify Lower Byte Write Cycle (μPD482444)/**
**Fast Page Mode Read Modify Lower Byte Block Write Cycle (μPD482444)**



**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Hyper Page Mode Read Modify Write Cycle (Extended data output: μPD482445)**

**Hyper Page Mode Read Modify Block Write Cycle (Extended data output: μPD482445)**

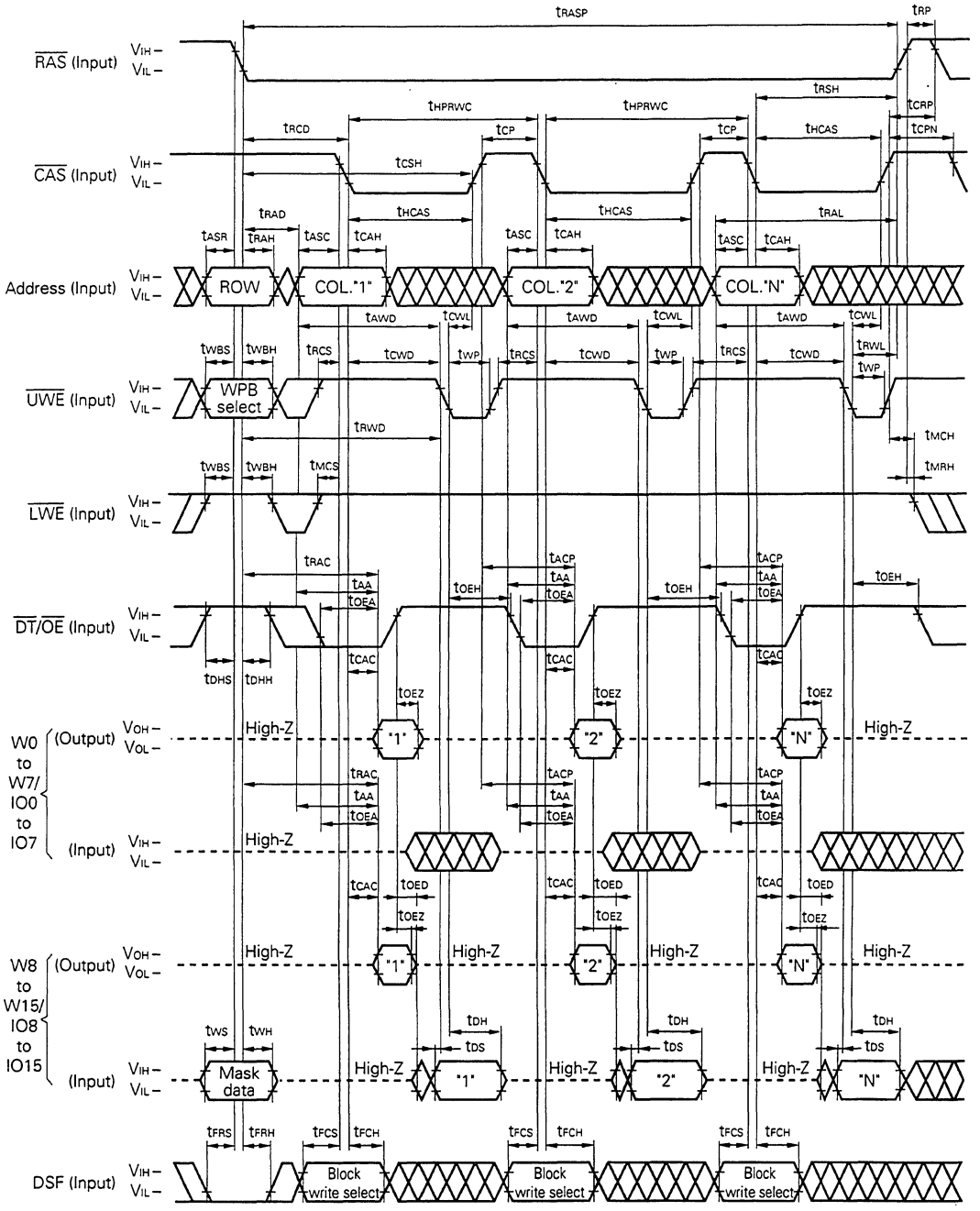



**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle

2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

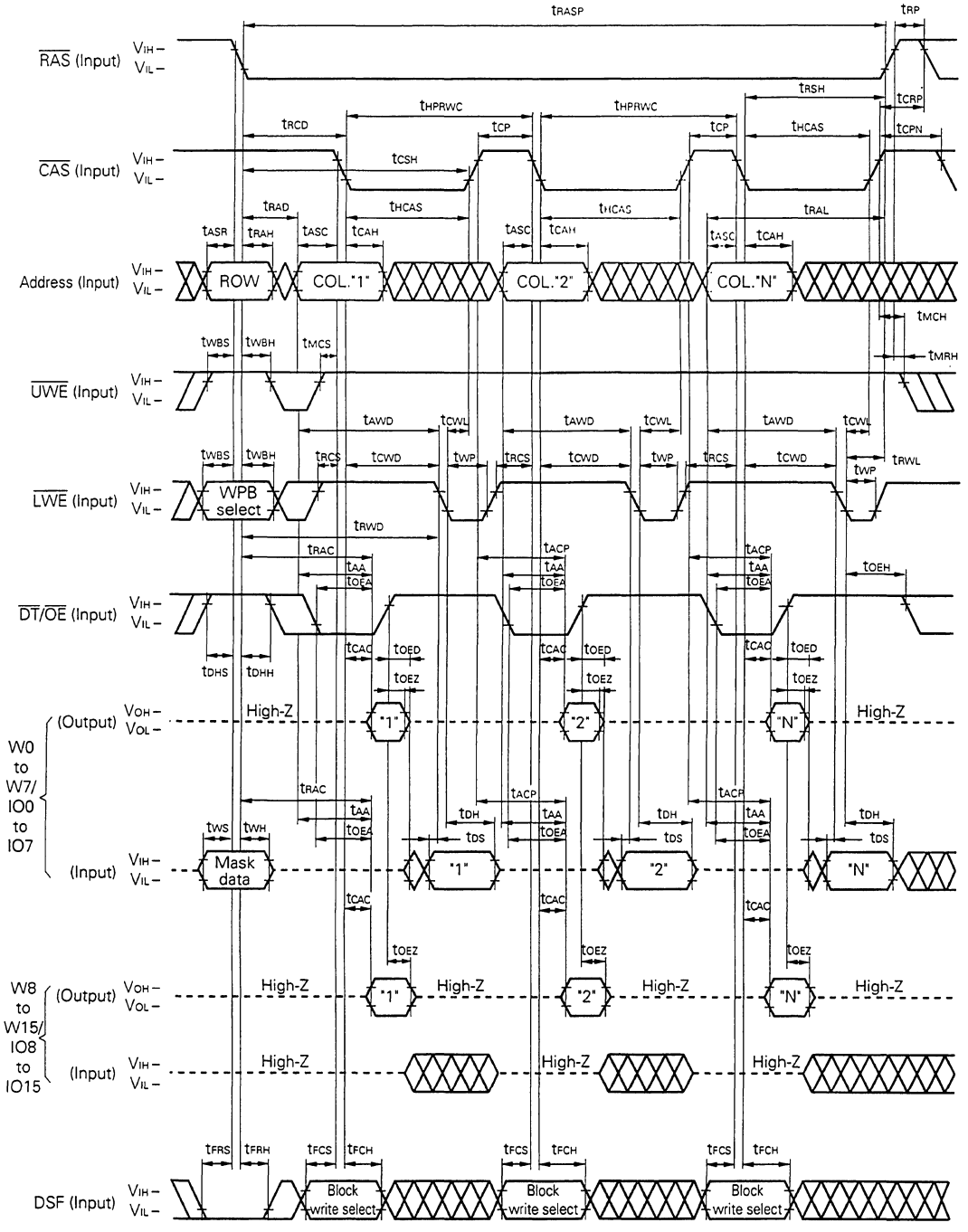
[MEMO]

**Hyper Page Mode Read Modify Upper Byte Write Cycle (Extended data output: μPD482445)**
**Hyper Page Mode Read Modify Upper Byte Block Write Cycle (Extended data output: μPD482445)**



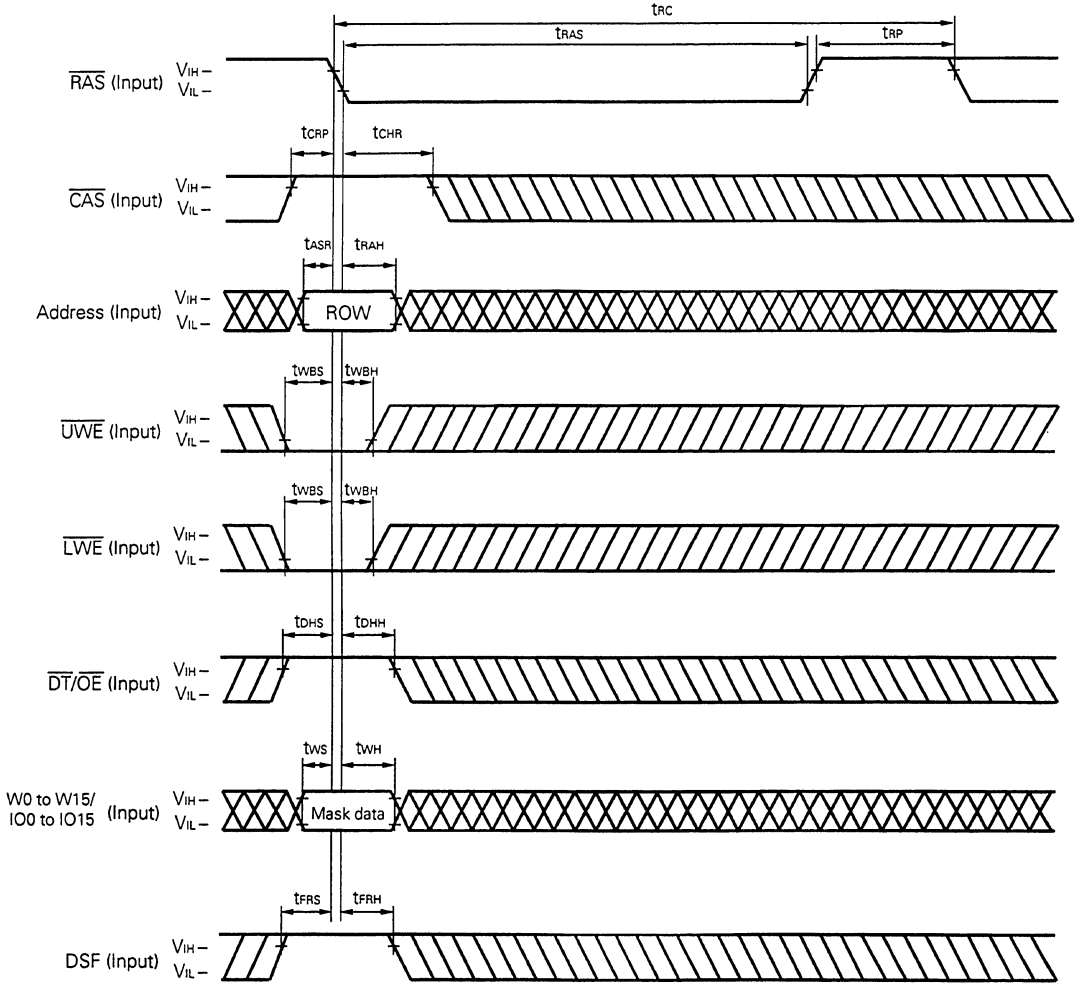
**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Hyper Page Mode Read Modify Lower Byte Write Cycle (Extended data output: μPD482445)**
**Hyper Page Mode Read Modify Lower Byte Block Write Cycle (Extended data output: μPD482445)**



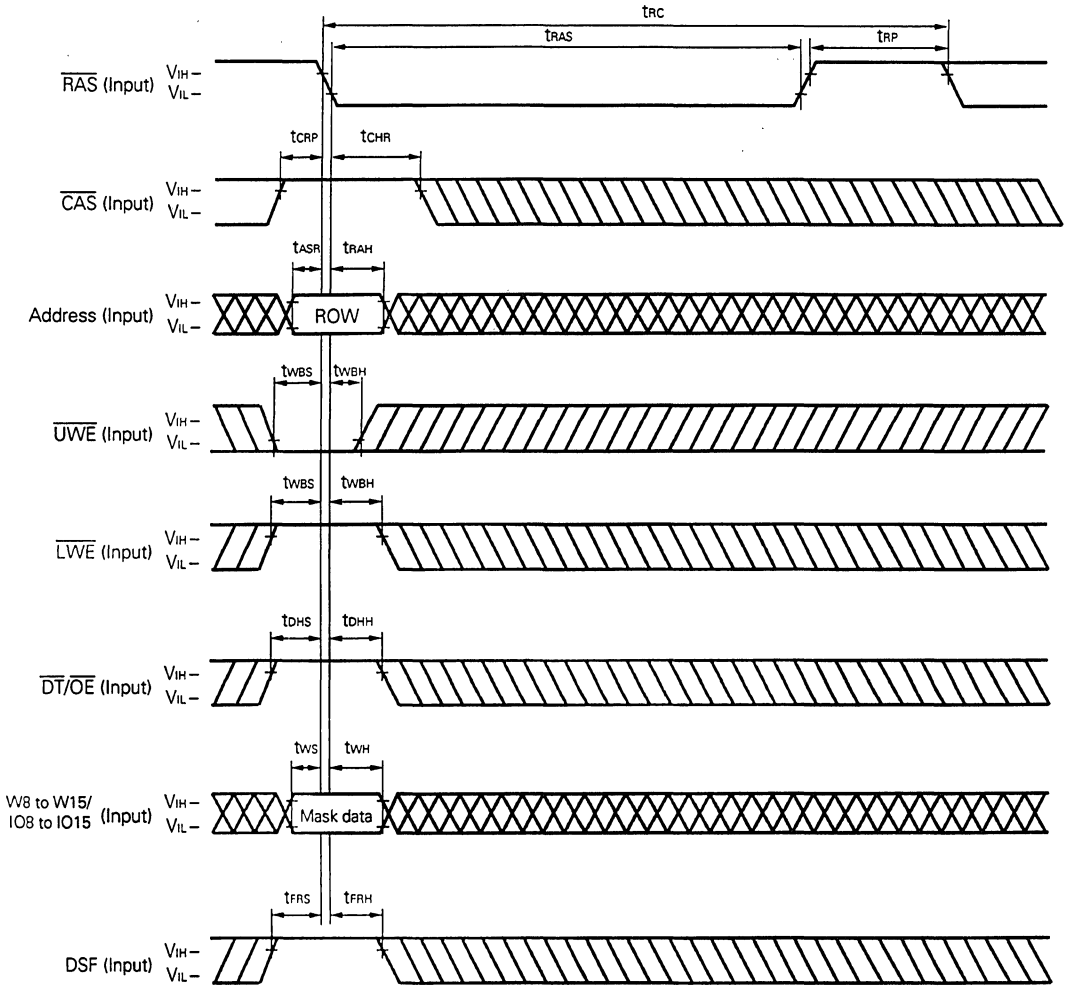
**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Flash Write Cycle**




**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

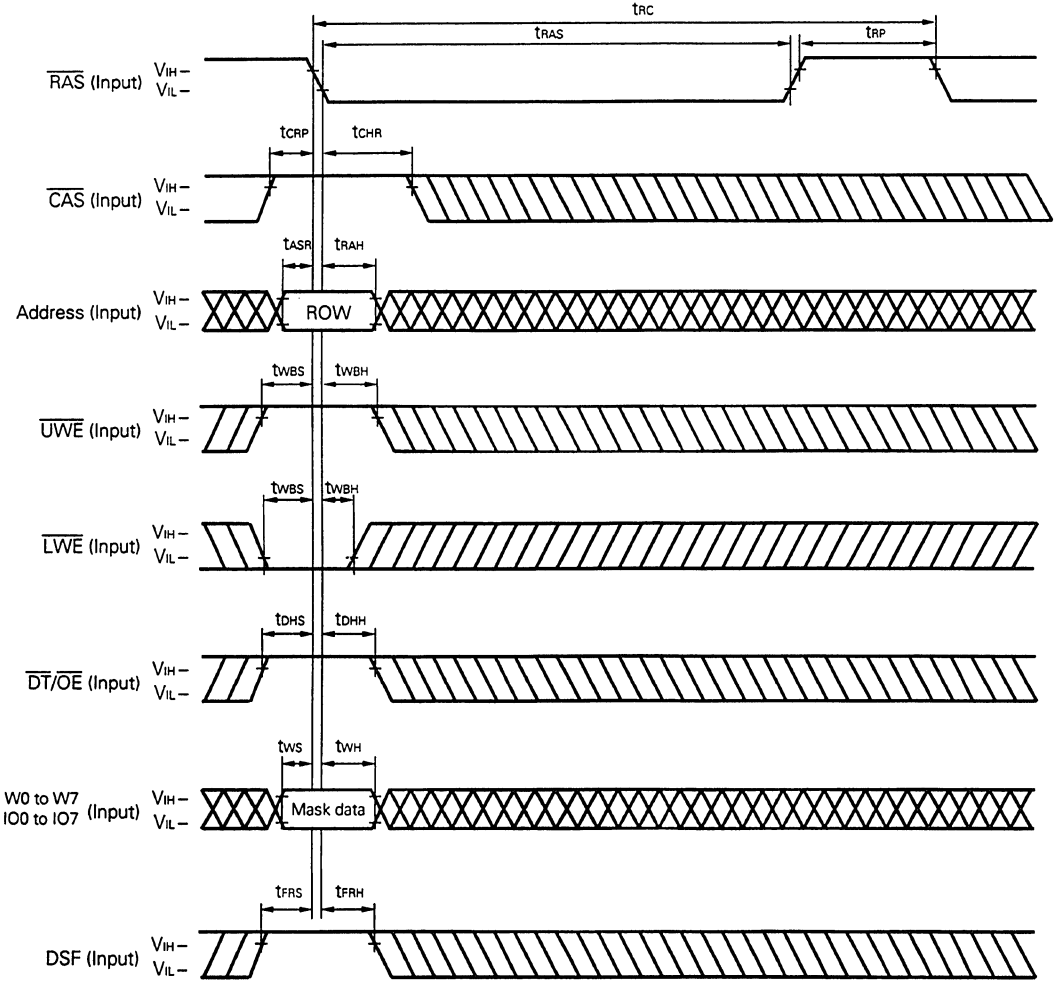
**Flash Write Cycle (Upper Byte Flash Write)**




**Caution After the falling edge of $\overline{RAS}$, the operations performed for the upper byte will be the same as the $\overline{RAS}$ only refresh cycle.**

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

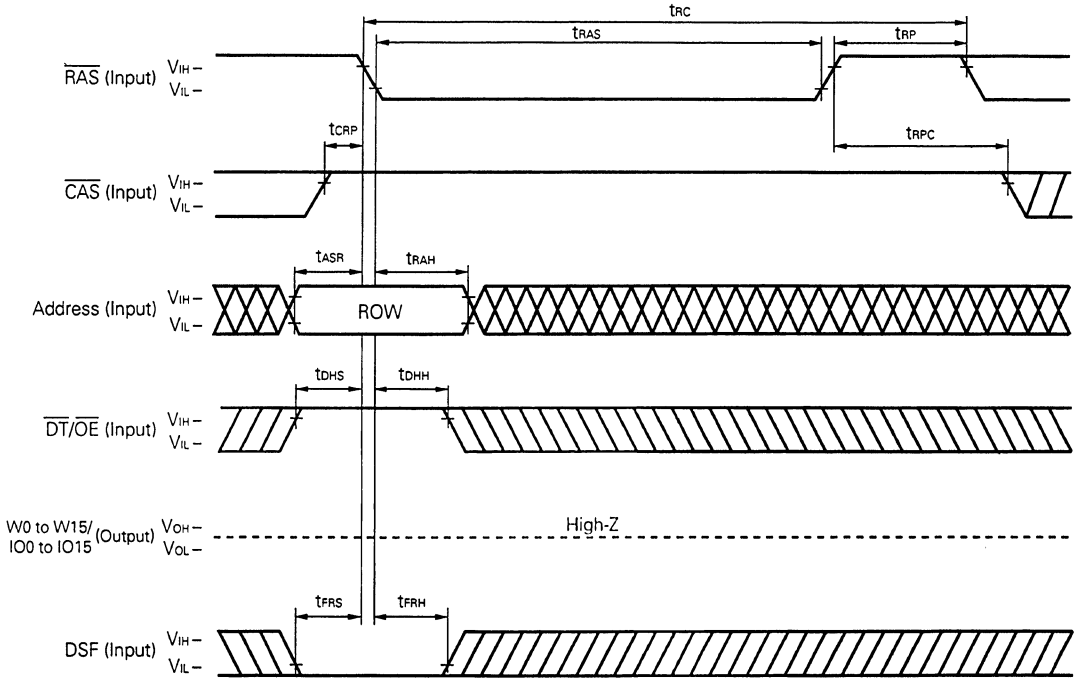
**Flash Write Cycle (Lower Byte Flash Write)**




**Caution After the falling edge of $\overline{RAS}$, the operations performed for the upper byte will be the same as the $\overline{RAS}$ only refresh cycle.**

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

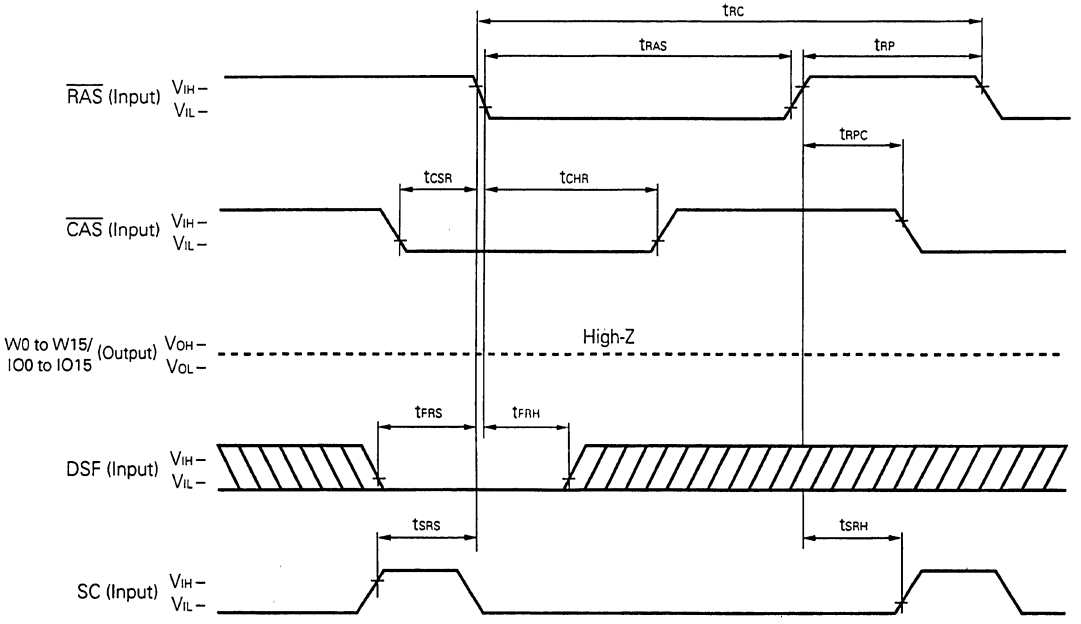
### $\overline{RAS}$ Only Refresh Cycle




**Remarks** 1. $\overline{UWE}$, $\overline{LWE}$ : Don't care

2. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

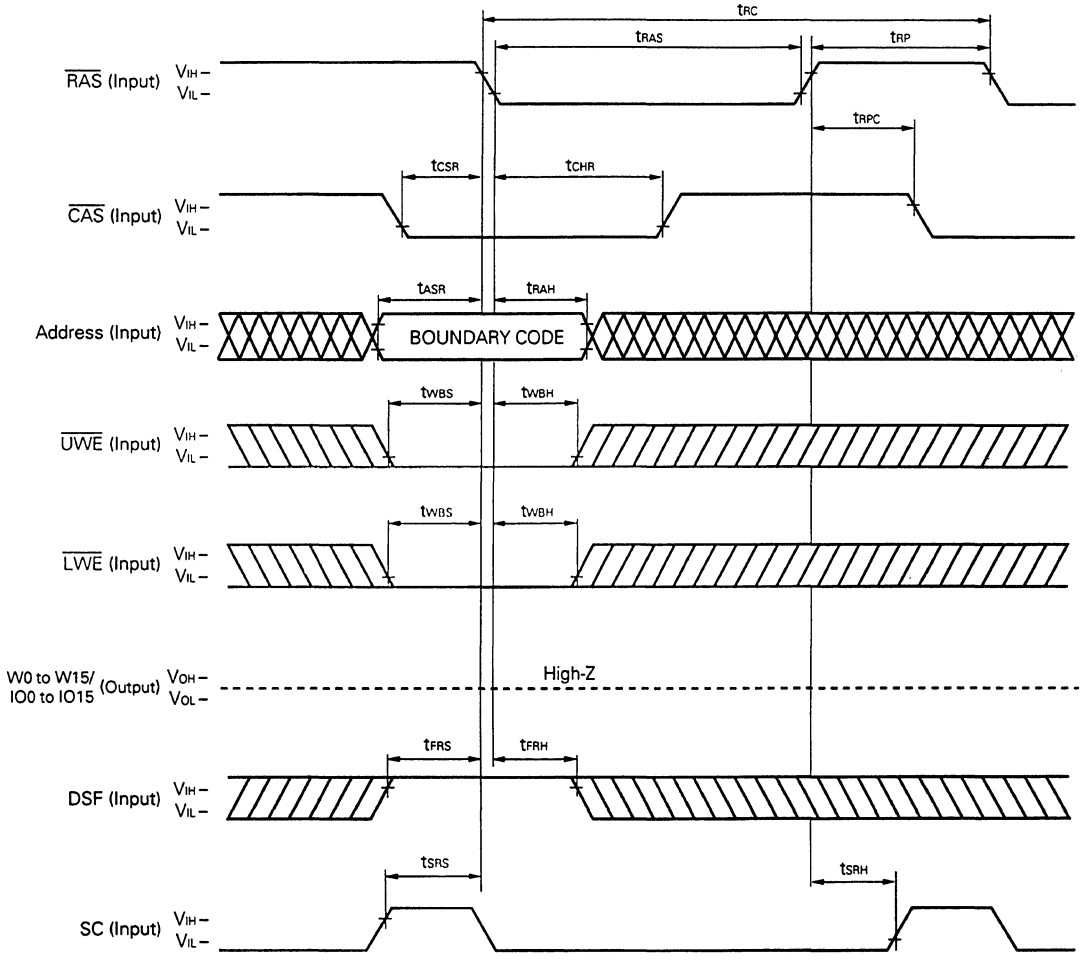
**$\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (Optional Reset)**




**Remarks** 1. A0 to A8, $\overline{UWE}$, $\overline{LWE}$, $\overline{DT}/\overline{OE}$ : Don't care

2. Because the serial access port operates independently of the random access port, there is no need to control the $\overline{SE}$, SIO pins in this cycle.

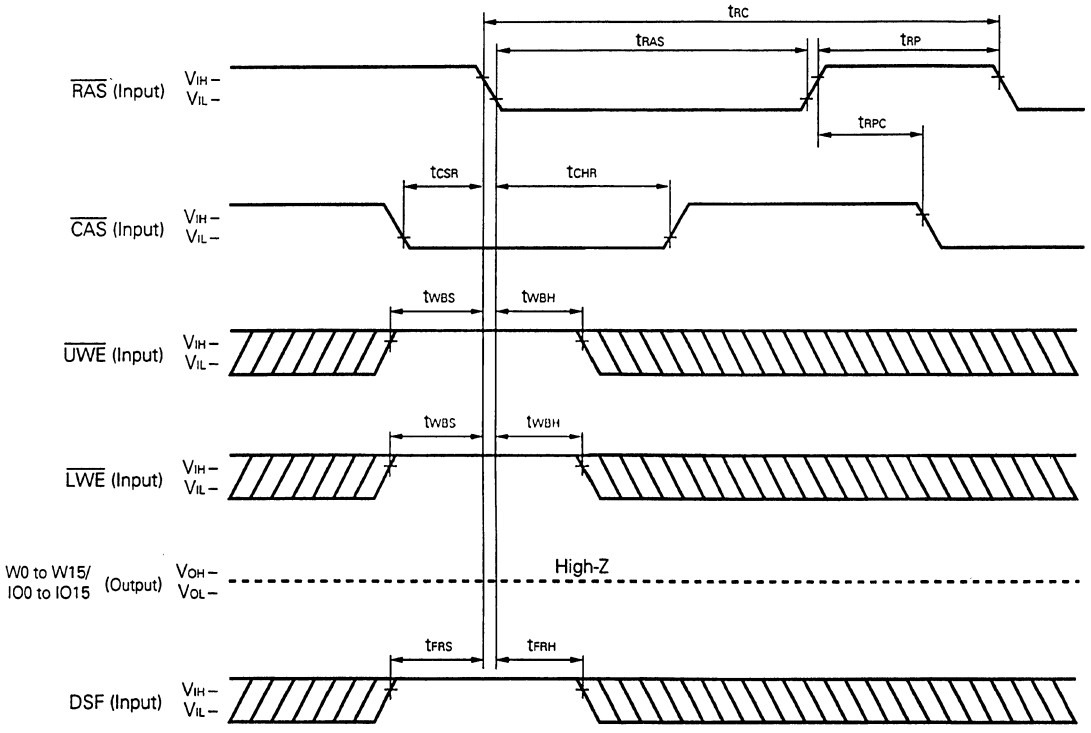
**$\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (STOP Register Set)**




**Remarks 1.** $\overline{DT}/\overline{OE}$ : Don't care

**2.** Because the serial access port operates independently of the random access port, there is no need to control the $\overline{SE}$, SIO pins in this cycle.

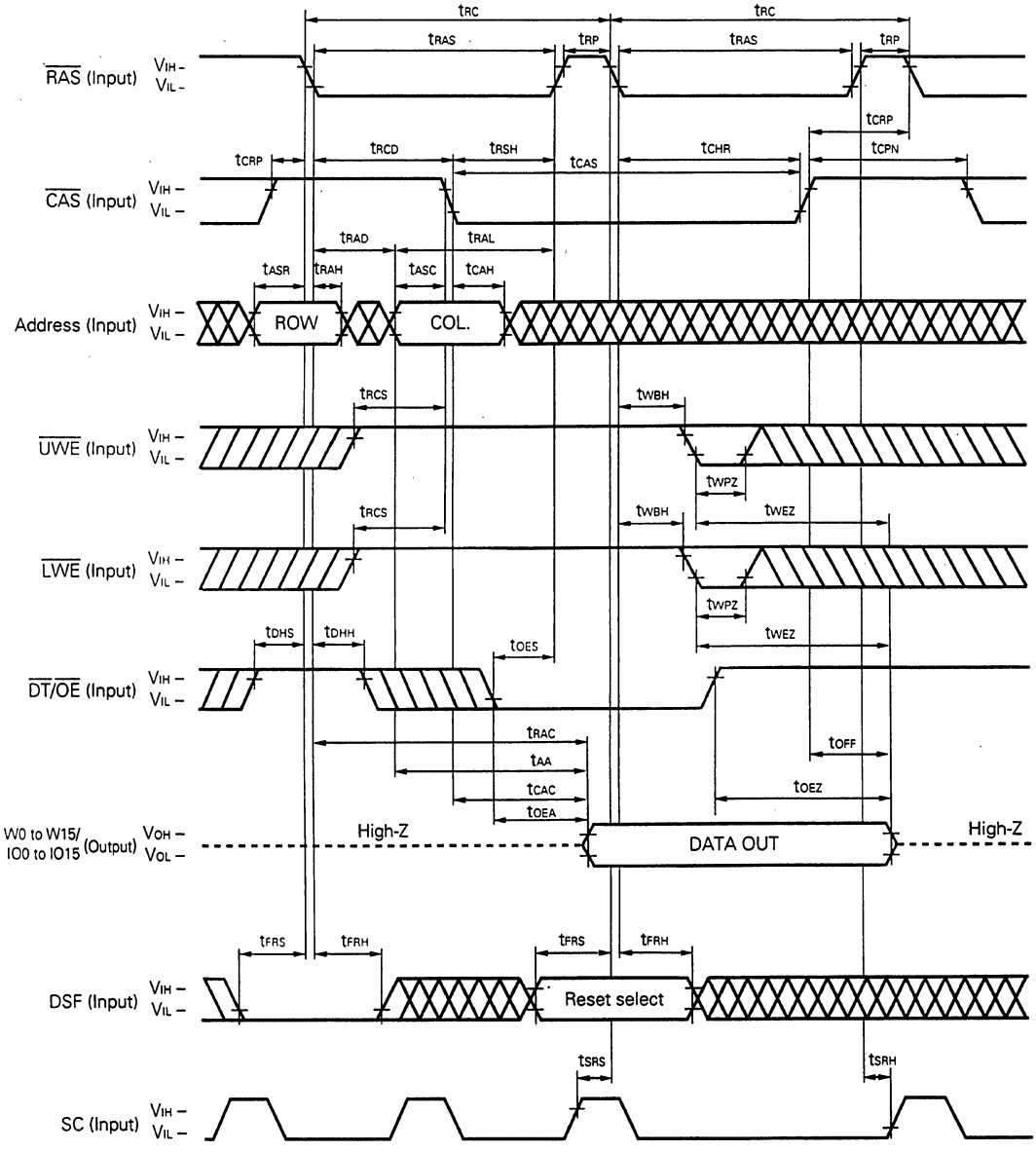
**$\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (No Reset)**




**Remarks** 1. A0 to A8, $\overline{DT}/\overline{OE}$ : Don't care

2. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

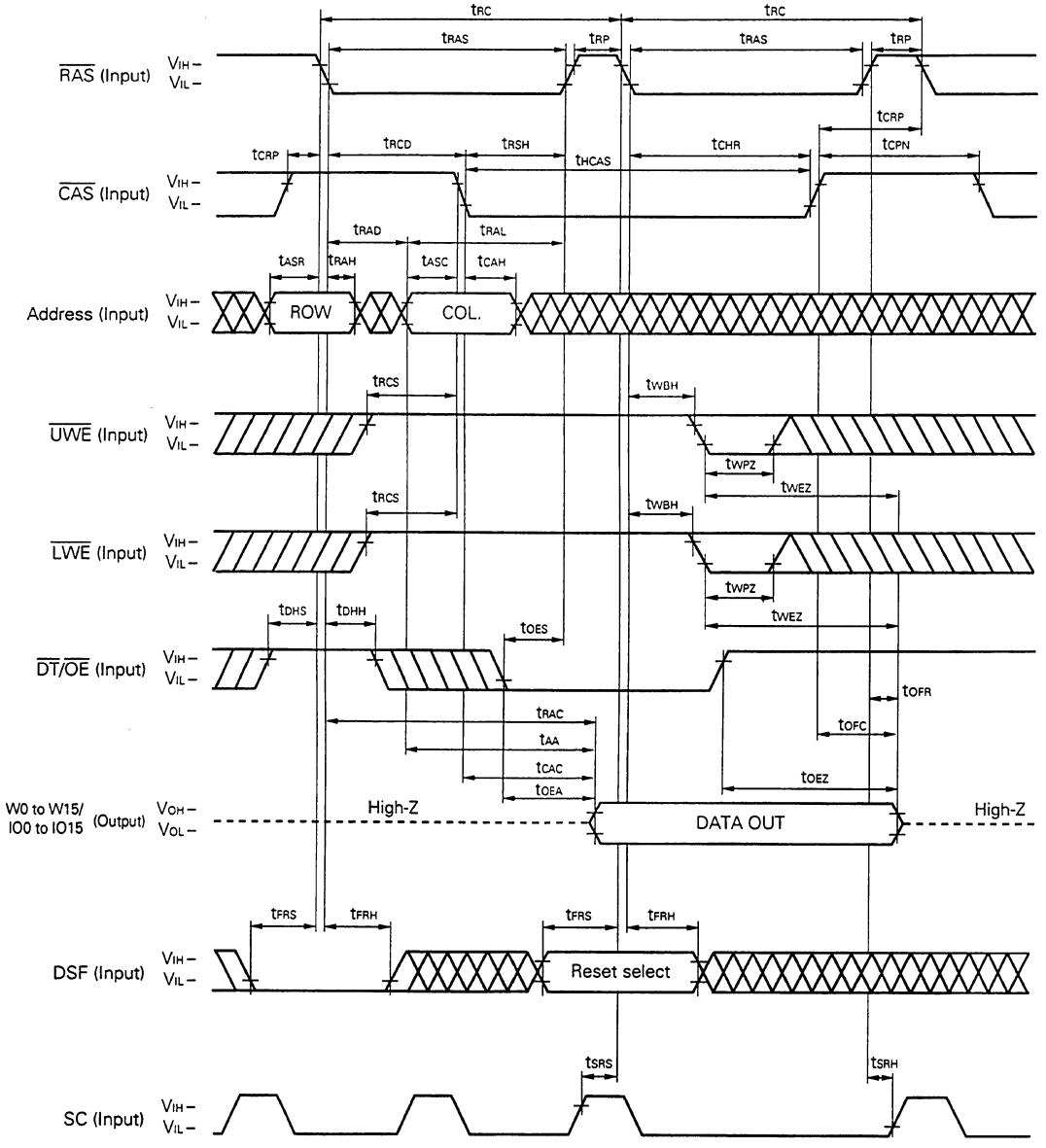
**Hidden Refresh Cycle (μPD482444)**




**Remarks** 1. When DSF is high level : Reset select = No Reset
When DSF is low level : Reset select = Optional Reset

2. Because the serial access port operates independently of the random access port, there is no need to control the $\overline{SE}$, SIO pins in this cycle.

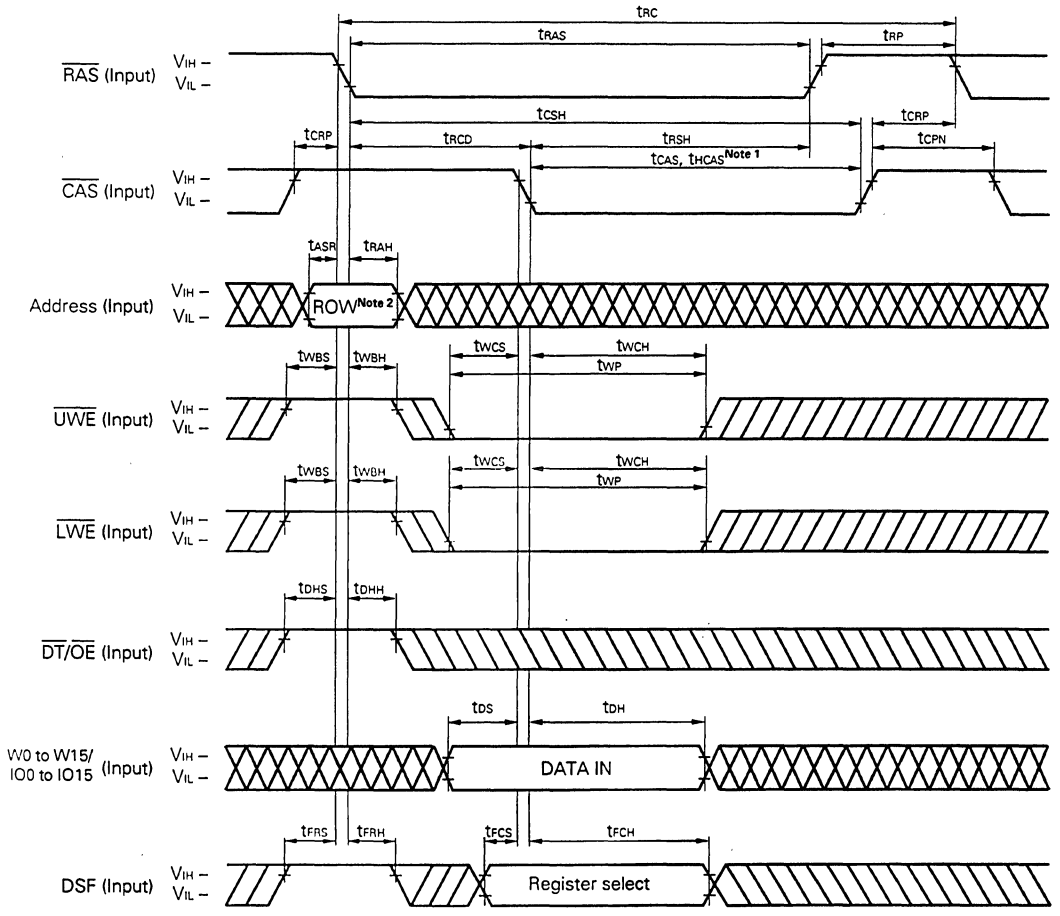
**Hidden Refresh Cycle (Extended data output: μPD482445)**




**Remarks 1.** When DSF is high level : Reset select = No Reset
When DSF is low level : Reset select = Optional Reset

**2.** Because the serial access port operates independently of the random access port, there is no need to control the $\overline{SE}$, SIO pins in this cycle.

**Register Set Cycle (Early Write)**



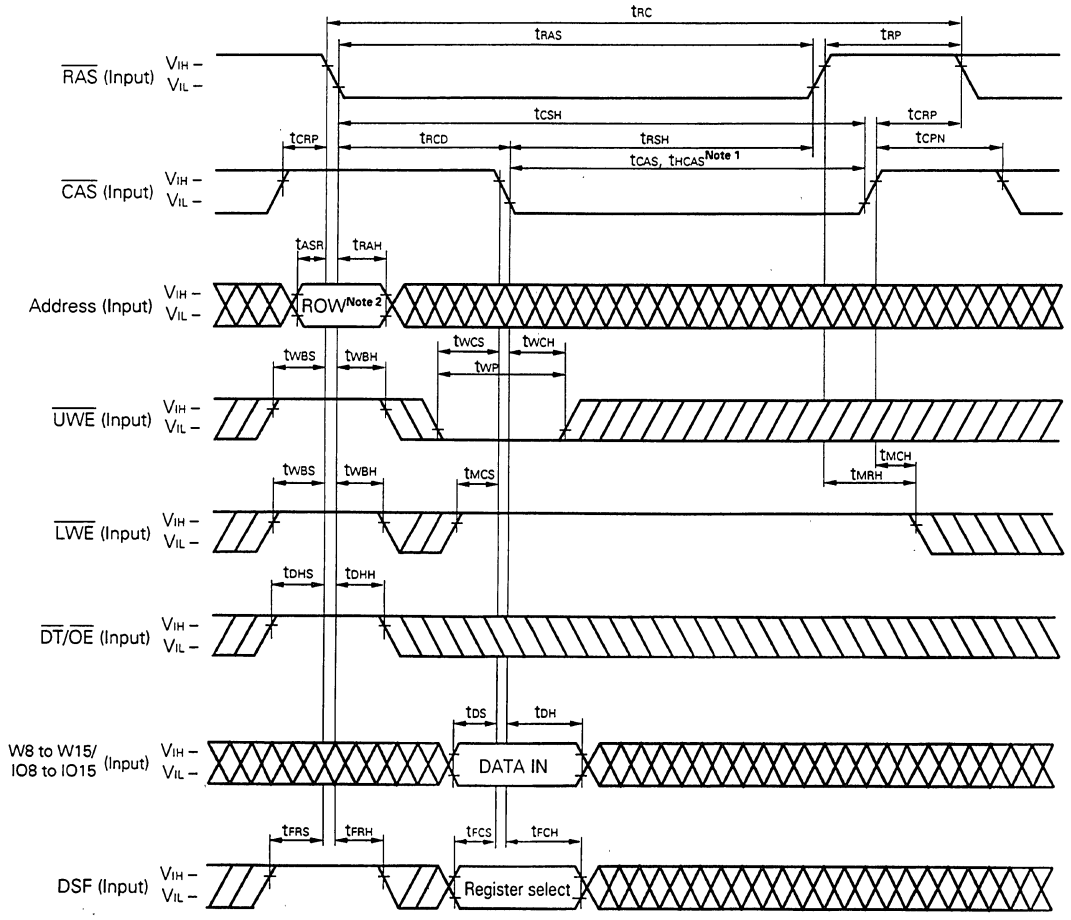

**Notes 1.** $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

**2.** Refresh address ($\overline{RAS}$ only refresh)

**Remarks 1.** When DSF is high level : Register select = Color Register Select
When DSF is low level : Register select = Write Mask Register Select

**2.** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

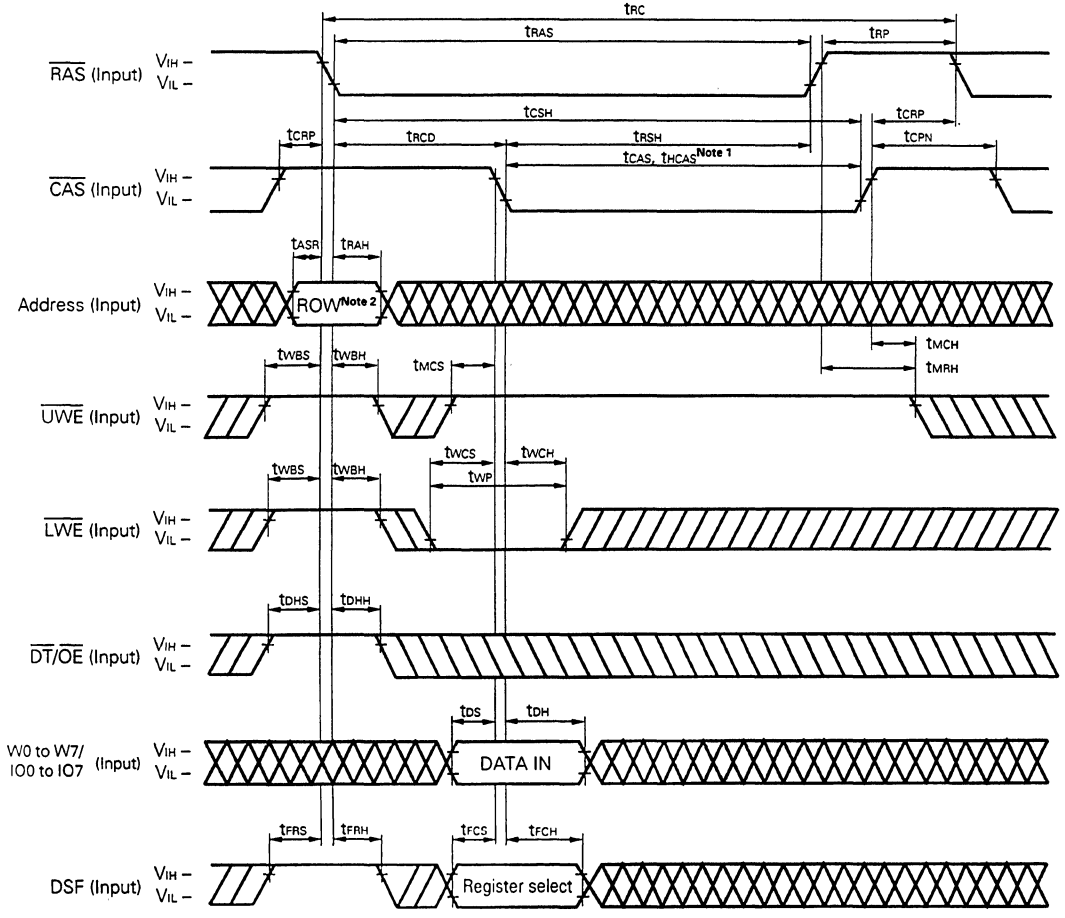
**Register Set Cycle (Upper Byte Early Write)**




**Notes** 1. tCAS for the μPD482444
tHCAS for the μPD482445
2. Refresh address ($\overline{RAS}$ only refresh)

**Remarks** 1. W0 to W7/IO0 to IO7 : Don't care
2. When DSF is high level : Register select = Color Register Select
When DSF is low level : Register select = Write Mask Register Select
3. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

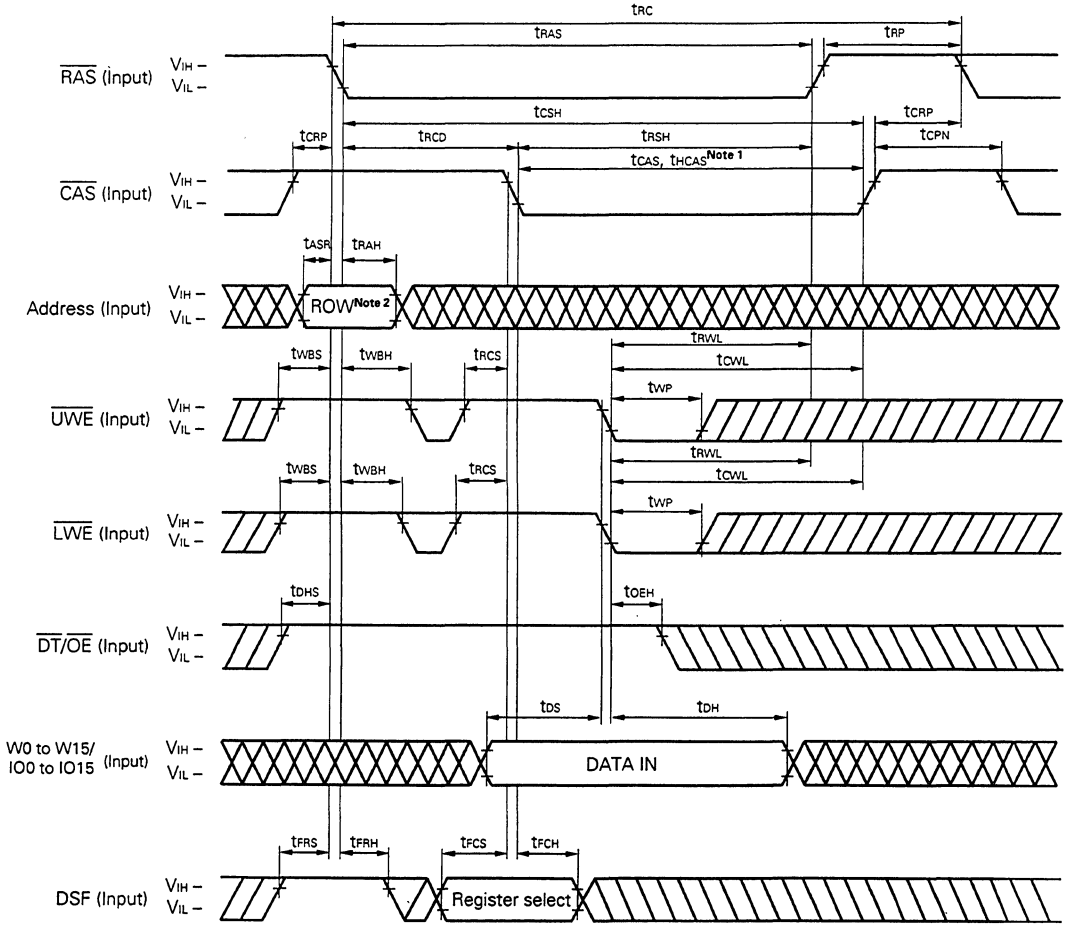
**Register Set Cycle (Lower Byte Early Write)**




**Notes** 1. tCAS for the μPD482444
tHCAS for the μPD482445
2. Refresh address ($\overline{RAS}$ only refresh)

**Remarks** 1. W8 to W15/IO8 to IO15 : Don't care
2. When DSF is high level : Register select = Color Register Select
When DSF is low level : Register select = Write Mask Register Select
3. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Register Set Cycle (Late Write)**



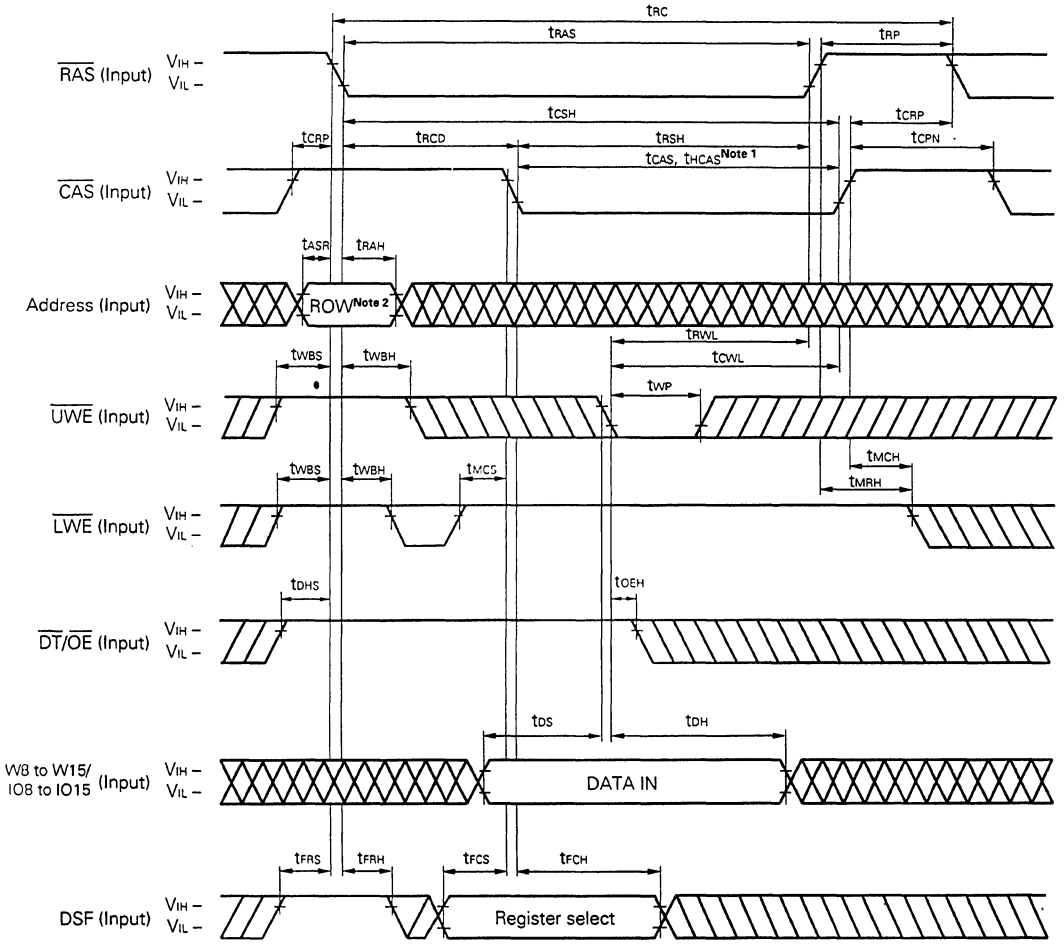

**Notes** 1. tCAS for the μPD482444
tHCAS for the μPD482445

2. Refresh address (RAS only refresh)

**Remarks** 1. When DSF is high level : Register select = Color Register Select
When DSF is low level : Register select = Write Mask Register Select

2. Because the serial access port operates independently of the random access port, there is no need to control the SC, SE, SIO pins in this cycle.

**Register Set Cycle (Upper Byte Late Write)**




**Notes 1.** tCAS for the μPD482444
tHCAS for the μPD482445

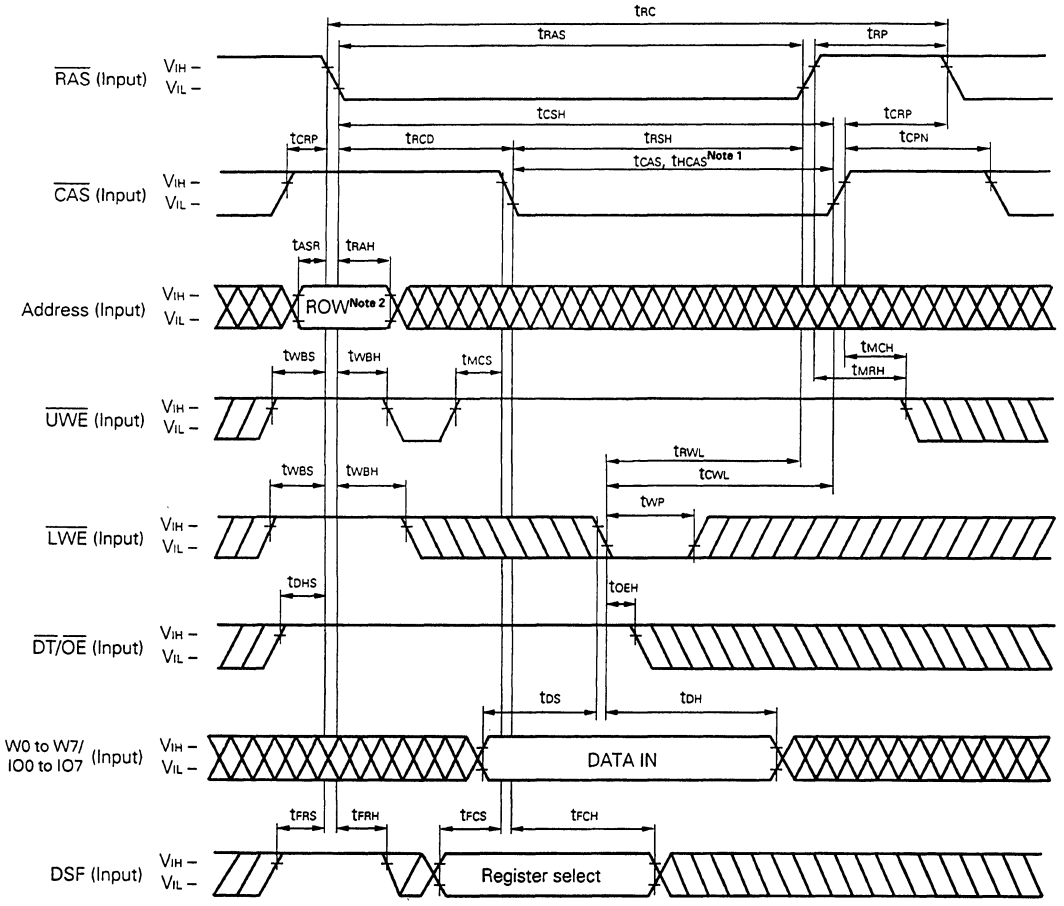
**2.** Refresh address ($\overline{RAS}$ only refresh)

**Remarks 1.** W0 to W7/IO0 to IO7 : Don't care

**2.** When DSF is high level : Register select = Color Register Select
When DSF is low level : Register select = Write Mask Register Select

**3.** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Register Set Cycle (Lower Byte Late Write)**




**Notes** 1. tCAS for the μPD482444
tHCAS for the μPD482445

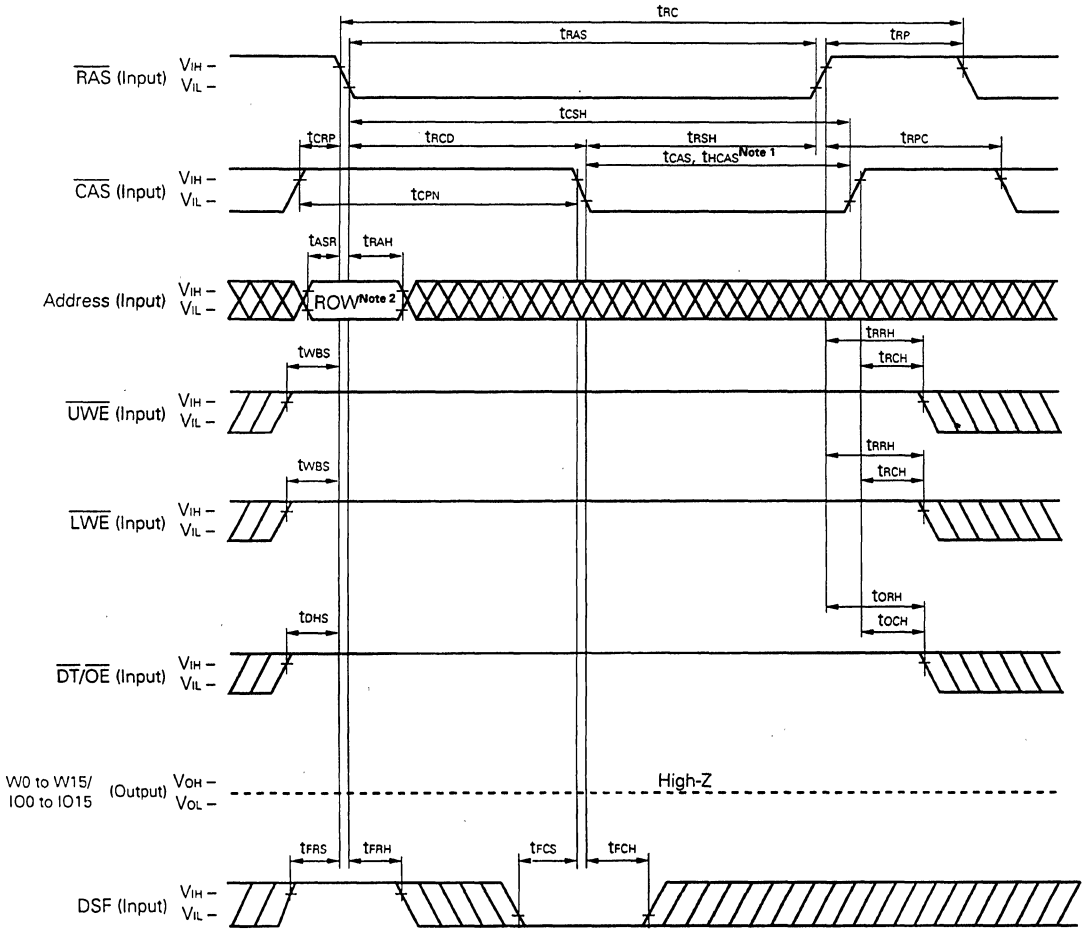
2. Refresh address ($\overline{RAS}$ only refresh)

**Remarks** 1. W8 to W15/IO8 to IO15 : Don't care

2. When DSF is high level : Register select = Color Register Select
When DSF is low level : Register select = Write Mask Register Select

3. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Mask Register Set Cycle (New Mask Selection)**

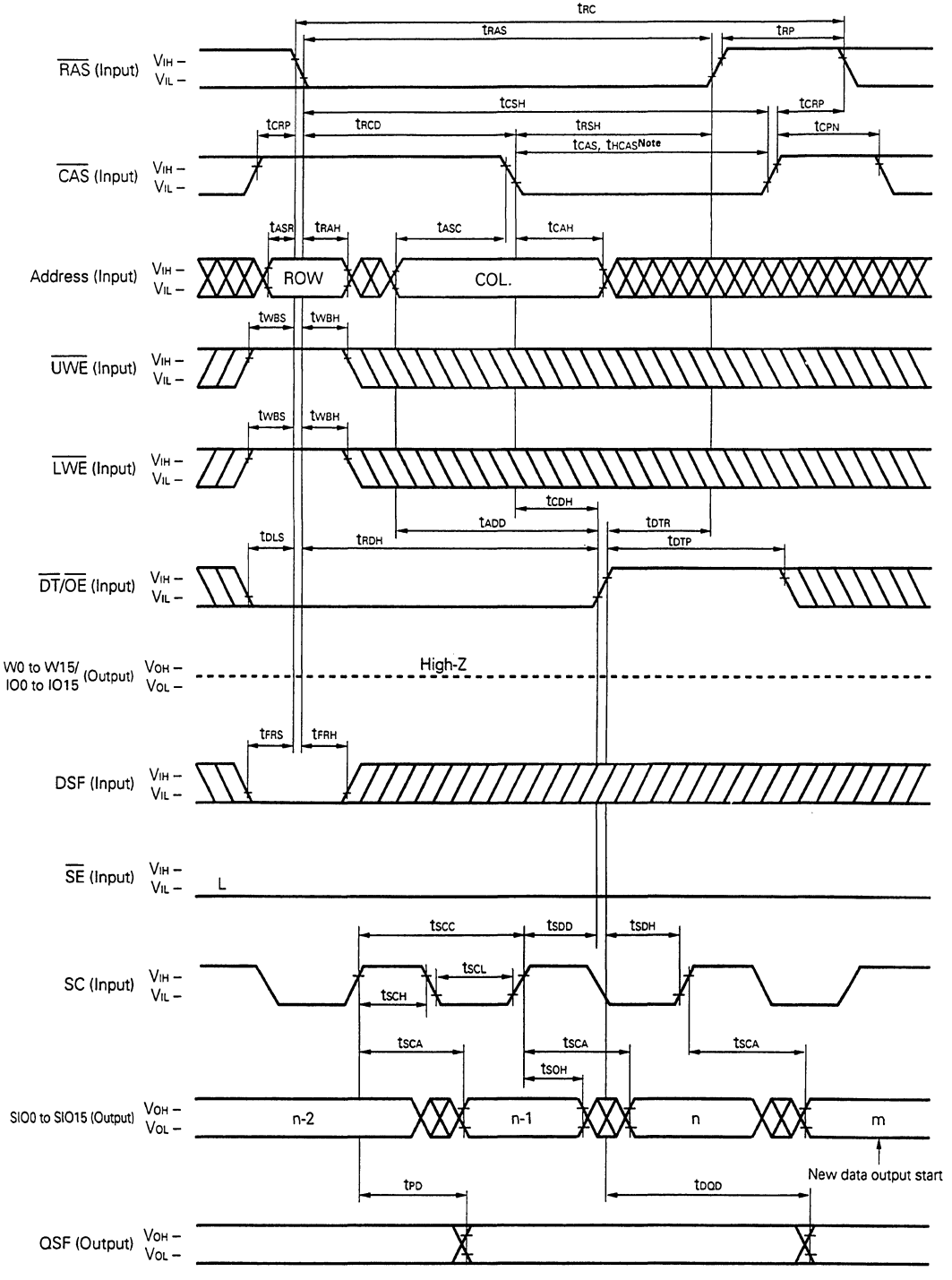



**Notes** 1. $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

2. Refresh address ($\overline{RAS}$ only refresh)

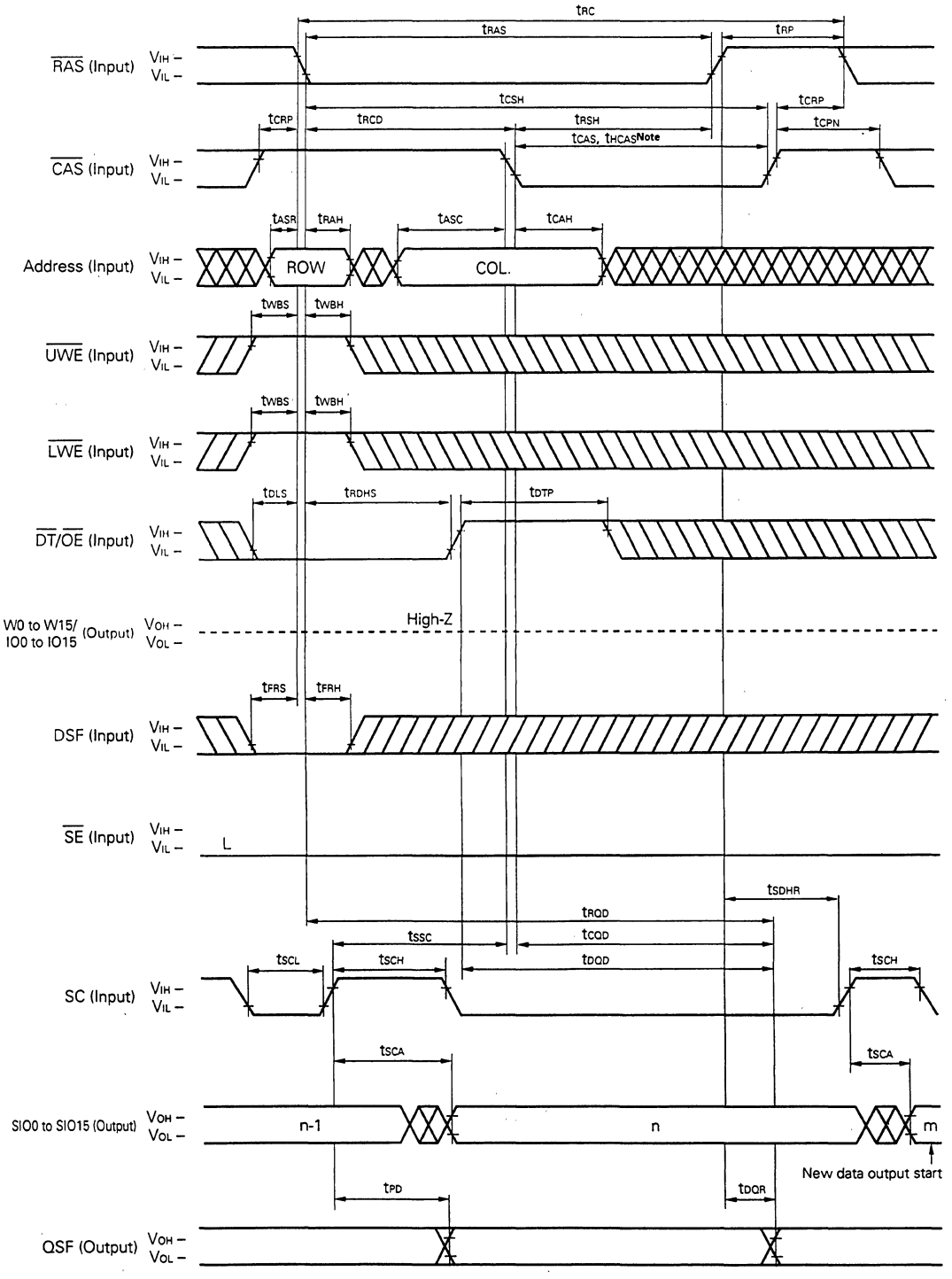
3. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Read Data Transfer Cycle (SC Active)**



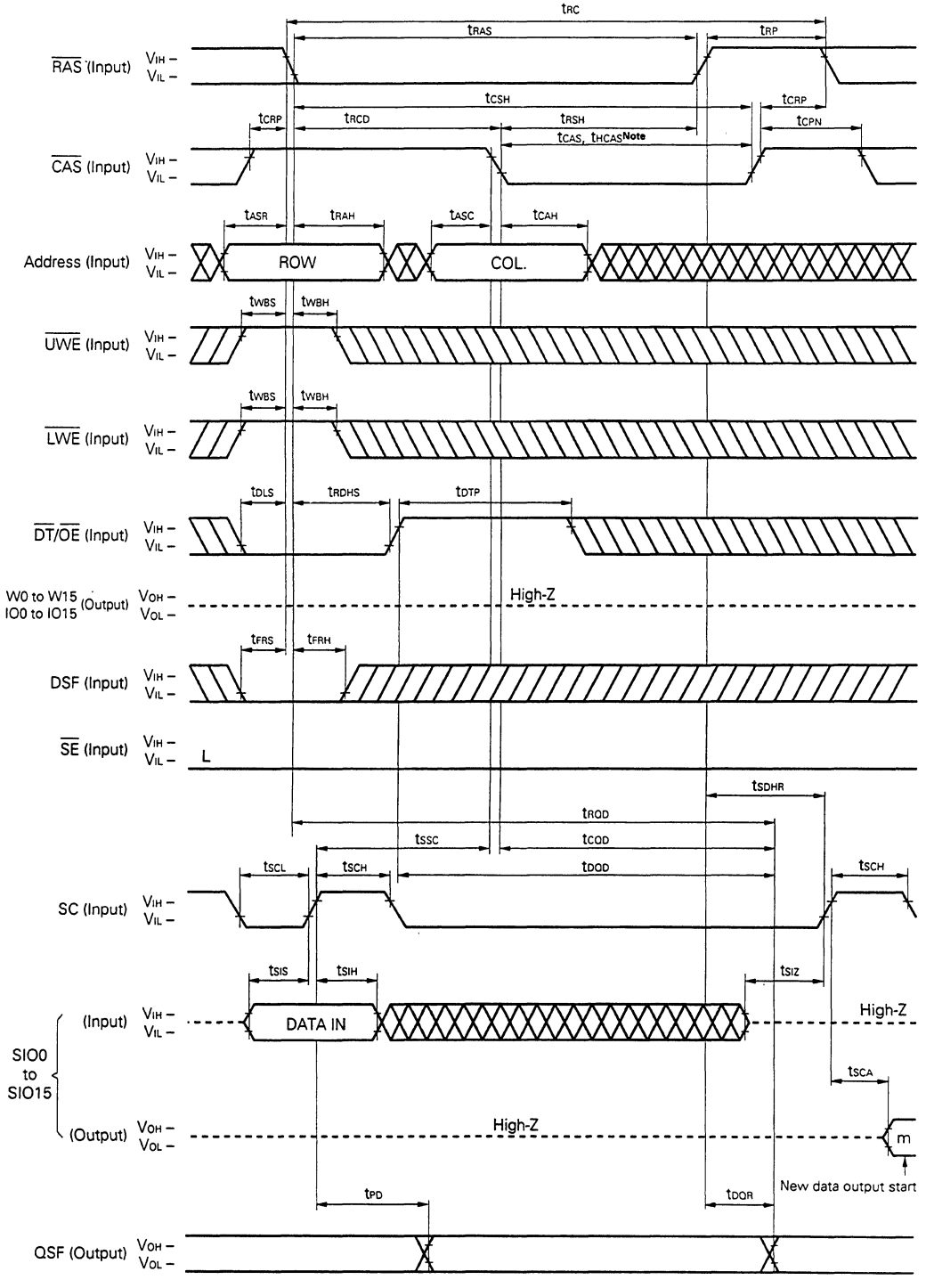

**Note** tCAS for the μPD482444
tHCAS for the μPD482445

**Read Data Transfer Cycle (SC Inactive)**




**Note** tCAS for the μPD482444
tHCAS for the μPD482445

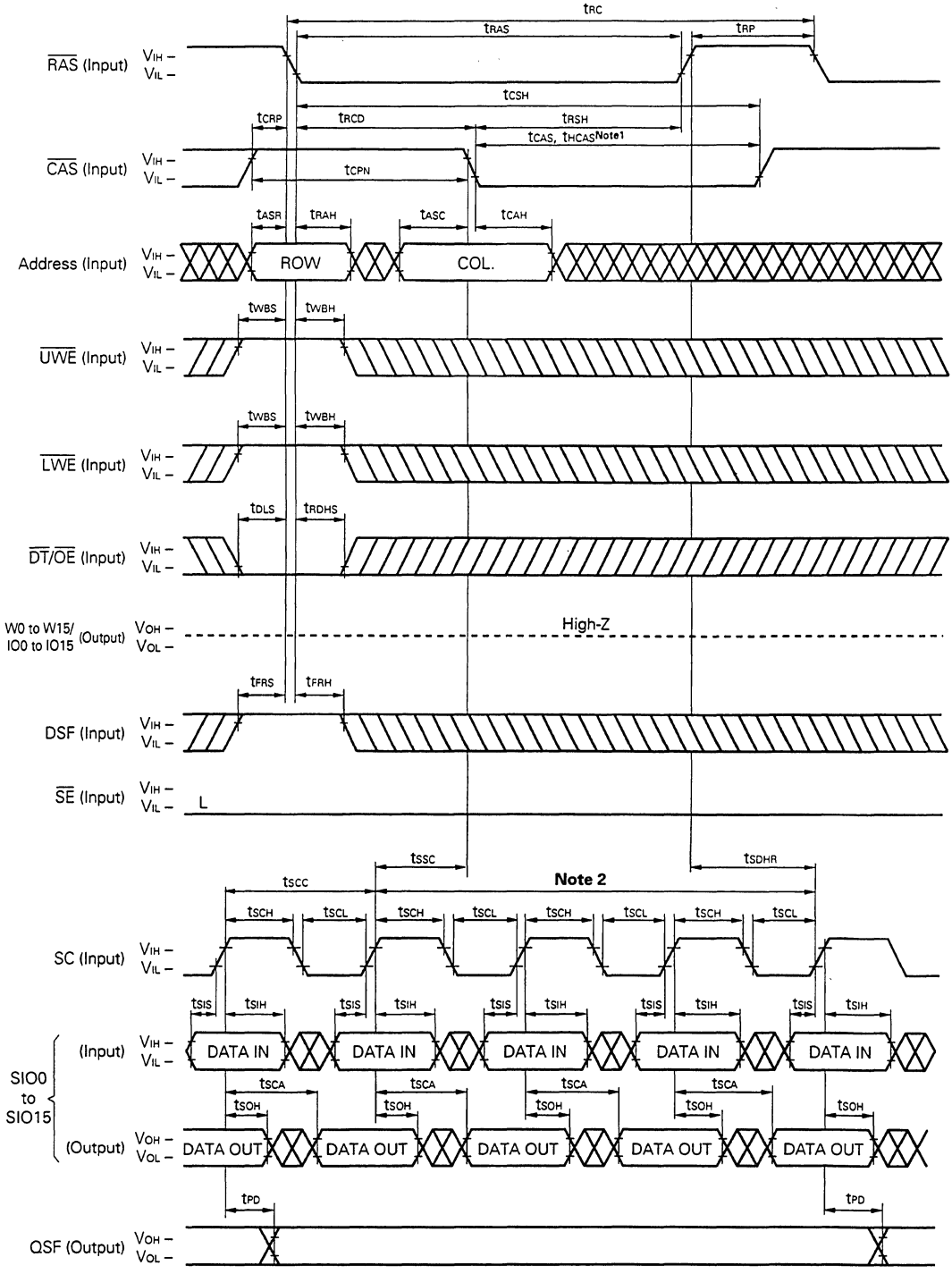
**Read Data Transfer Cycle (Serial Write → Serial Read Switching)**




**Note** $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

[MEMO]

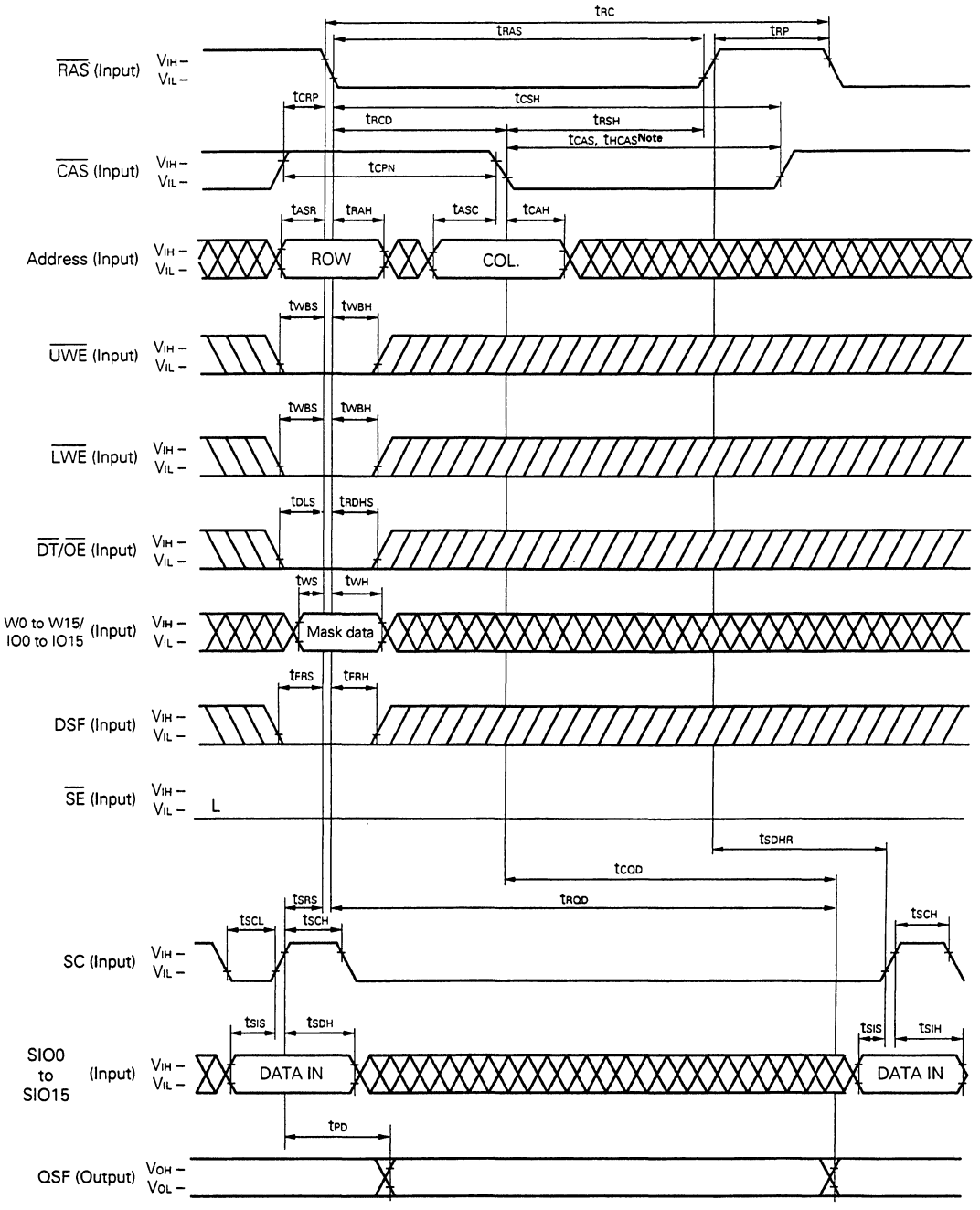
**Split Read Data Transfer Cycle**



**Notes 1.** $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

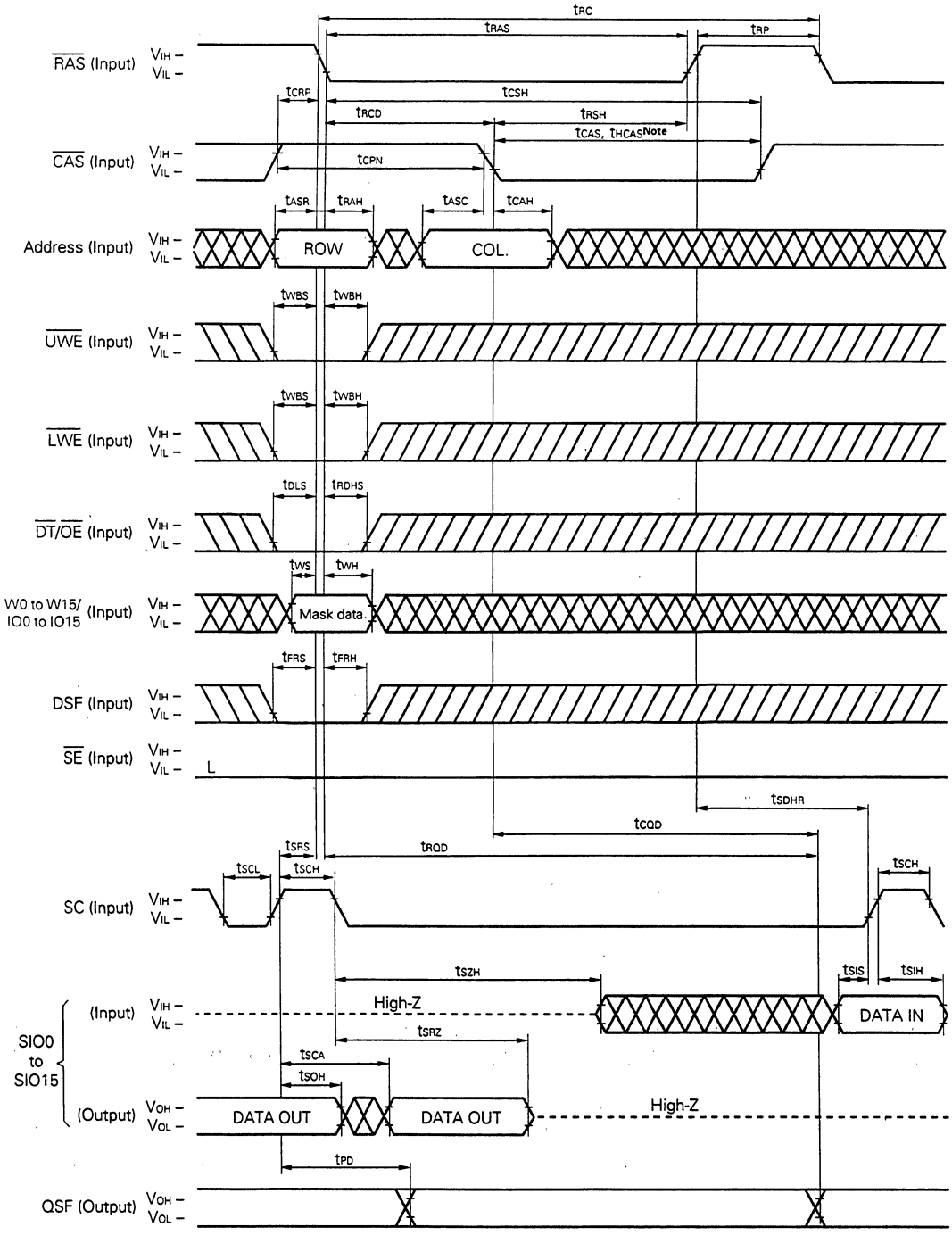
2. Do not perform the following two serial read/write during this period.
   - Serial read/write of jump source address set to the STOP register of the data register which does not perform the data transfer cycle.
   - Serial read/write of last address of data register (Address 255 or 511)

**Write Data Transfer Cycle**



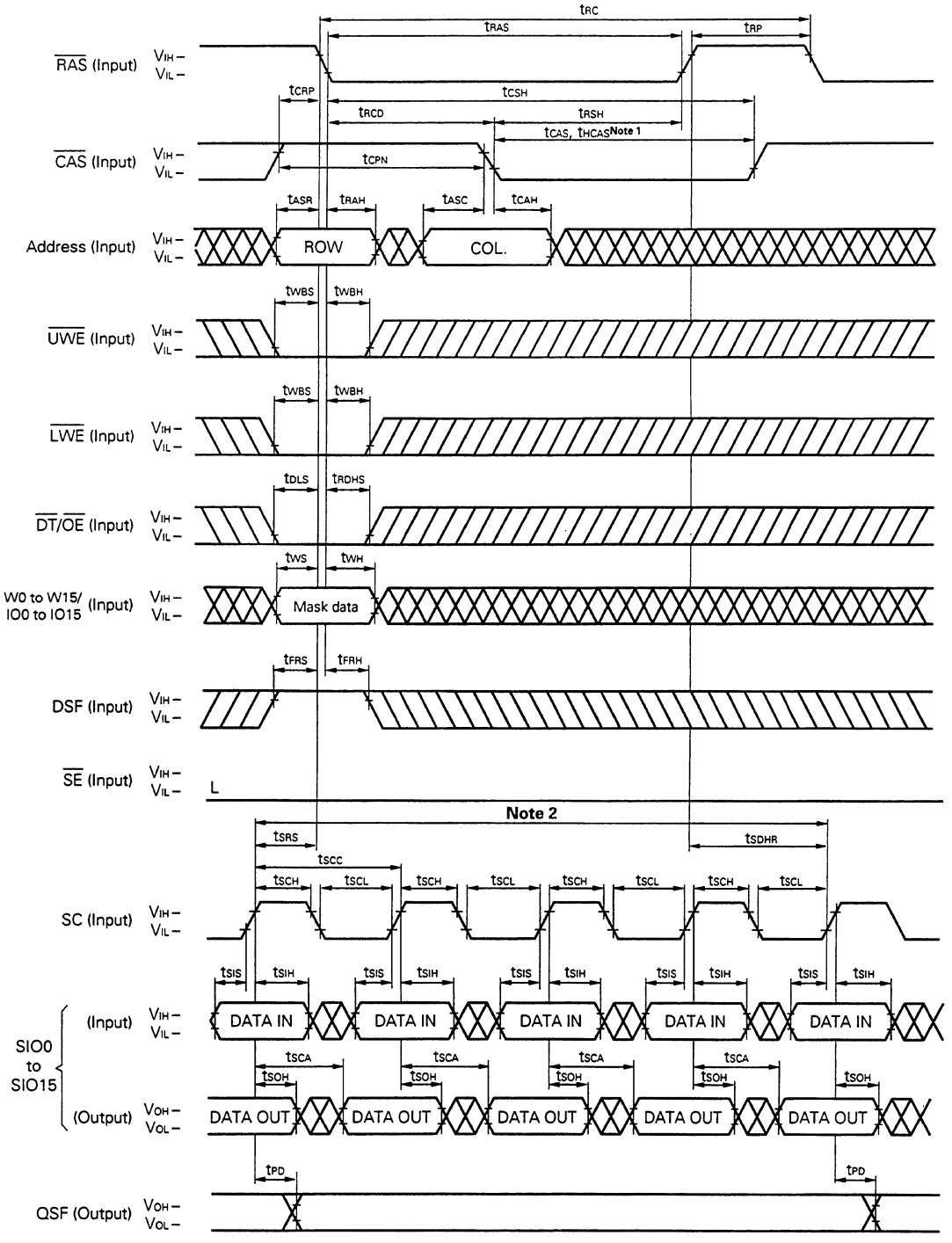

**Note** $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

**Write Data Transfer Cycle (Serial Read → Serial Write Switching)**




**Note** tCAS for the μPD482444
tHCAS for the μPD482445

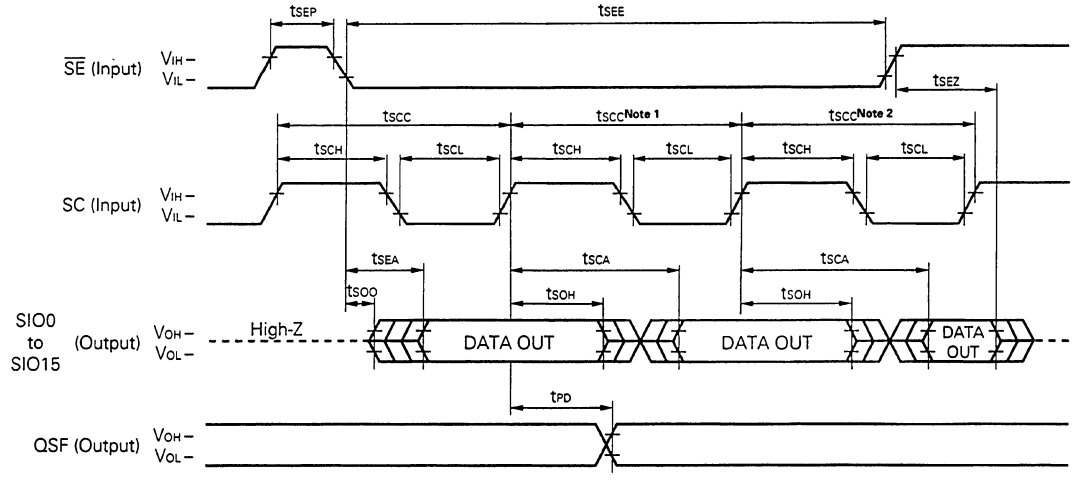
**Split Write Data Transfer Cycle**



**Notes** 1. $t_{CAS}$ for the μPD482444
$t_{HCAS}$ for the μPD482445

2. Do not perform the following two serial read/write during this period.
   - Serial read/write of jump source address set to the STOP register of the data register which does not perform the data transfer cycle.
   - Serial read/write of last address of data register (Address 255 or 511)

**Serial Read Cycle**

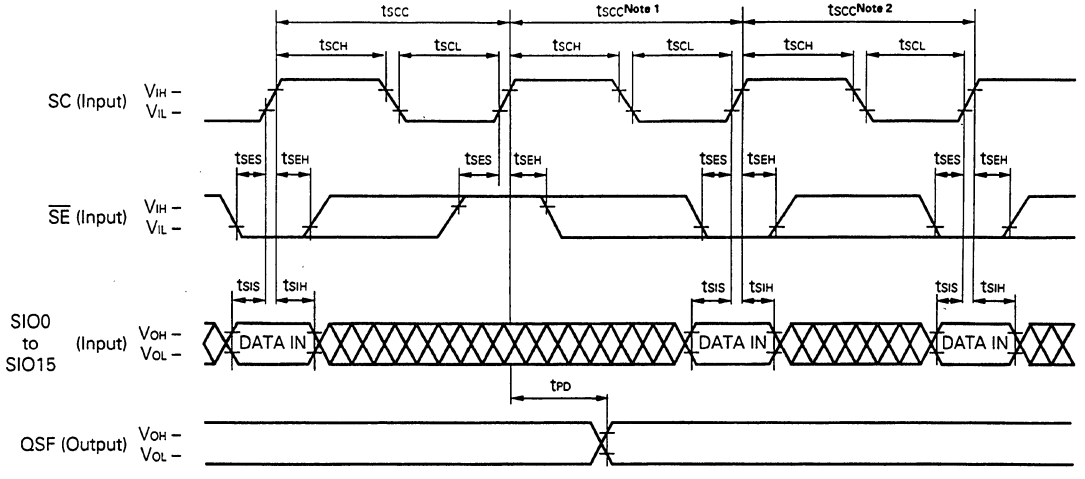



**Notes** 1. Last address of data register (Address 255 or 511)

2. Starting address of data register newly read (address is specified in the data transfer cycle).

**Remark** Because the random access port operates independently of the serial access port, there is no need to control the $\overline{RAS}$, $\overline{CAS}$, Address, $\overline{UWE}$, $\overline{LWE}$, $\overline{DT/OE}$, WI/O, DSF pins in this cycle.

**Serial Write Cycle**



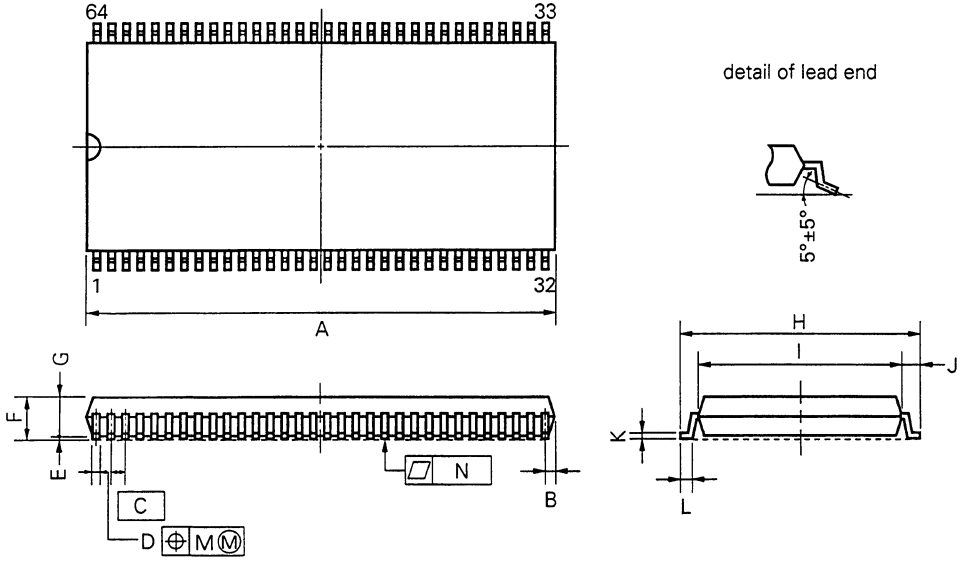

**Notes** 1. Last address of data register (Address 255 or 511)

2. Starting address of data register newly read (address is specified in the data transfer cycle).

**Remark** Because the random access port operates independently of the serial access port, there is no need to control the $\overline{RAS}$, $\overline{CAS}$, Address, $\overline{UWE}$, $\overline{LWE}$, $\overline{DT/OE}$, WI/O, DSF pins in this cycle.

## 5. Package Drawings

### 64 PIN PLASTIC SHRINK SOP (525 mil)




P64GW-80-525A-1

**NOTE**

Each lead centerline is located within 0.10 mm (0.004 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 26.30 MAX. | 1.036 MAX. |
| B | 0.75 MAX. | 0.030 MAX. |
| C | 0.8 (T.P.) | 0.031 (T.P.) |
| D | $0.35\pm0.05$ | $0.014^{+0.002}_{-0.003}$ |
| E | $0.15\pm0.05$ | $0.006\pm0.002$ |
| F | 2.3 MAX. | 0.091 MAX. |
| G | 2.0 | 0.079 |
| H | $13.8\pm0.3$ | $0.543^{+0.013}_{-0.012}$ |
| I | $11.8\pm0.1$ | $0.465^{+0.004}_{-0.005}$ |
| J | $1.0\pm0.2$ | $0.039^{+0.009}_{-0.008}$ |
| K | $0.20^{+0.10}_{-0.05}$ | $0.008^{+0.004}_{-0.002}$ |
| L | $0.5\pm0.2$ | $0.020^{+0.008}_{-0.009}$ |
| M | 0.10 | 0.004 |
| N | 0.10 | 0.004 |

## 6. Recommended Soldering Conditions

Please consult with our sales offices for soldering conditions of the μPD482444 and 482445 .

**Types of Surface Mount Device**

μPD482444GW : 64-Pin Plastic Shrink SOP (525 mil)
μPD482445GW : 64-Pin Plastic Shrink SOP (525 mil)

[MEMO]

NEC

MOS INTEGRATED CIRCUIT

# μPD482234, 482235

## 2M-Bit Dual Port Graphics Buffer

## 256K-WORD BY 8-BIT

### Description

The μPD482234 and μPD482235 have a random access port and a serial access port. The random access port has a 2M-bit (262, 144 words × 8 bits) memory cell array structure. The serial access port can perform clock operations of up to 55 MHz from the 4K-bit data register (512 words × 8 bits).

To simplify the graphics system design, the split data transfer function and binary boundary jump function have been adopted so that the number of split data registers can be programmed with the software during serial read/write operations.

The μPD482235 is provided with the hyper page mode, an improved version of the fast page mode of the μPD482234. The random access port can input and output data by $\overline{CAS}$ clock operations of up to 33 MHz.

### Features

Dual port structure (Random access port, Serial access port)

● Random access port (262, 144-word × 8-bit structure)

**μPD482234**

| | μPD482234-60 | μPD482234-70 | μPD482234-80 |
|---|---|---|---|
| $\overline{RAS}$ access time | 60 ns(MAX.) | 70 ns(MAX.) | 80 ns(MAX.) |
| Fast page mode cycle time | 40 ns(MIN.) | 45 ns(MIN.) | 50 ns(MIN.) |

**μPD482235**

| | μPD482235-60 | μPD482235-70 | μPD482235-80 |
|---|---|---|---|
| $\overline{RAS}$ access time | 60 ns(MAX.) | 70 ns(MAX.) | 80 ns(MAX.) |
| Hyper page mode cycle time | 30 ns(MIN.) | 35 ns(MIN.) | 40 ns(MIN.) |

- Flash write function[Note]
- Block write function (4 columns)[Note]
- Mask write (Write-per-bit function)
- 512 refresh cycles /8 ms
- $\overline{CAS}$ before $\overline{RAS}$ refresh, $\overline{RAS}$ only refresh, Hidden refresh

**Note** Write-per-bit can be specified.

● Serial access port (512 words × 8 bits organization)

- Serial read/write cycle time

| μPD482234-60, 482235-60 | μPD482234-70, 482235-70 | μPD482234-80, 482235-80 |
|---|---|---|
| 18 ns (MIN.) | 22 ns (MIN.) | 25 ns (MIN.) |

- Serial data read/write
- Split buffer data transfer
- Binary boundary jump function

**Version A, F, and E**

There are three versions, A, F, and E, to both the μPD482234 and μPD482235. These versions operate with different specifications.

- **Overview of each version**

**μPD482234**

The table below summarizes the operation of each version of the μPD482234.

| Item | Reference | Version A | Versions F and E |
|---|---|---|---|
| Specifying a column for data transfer during split data transfer cycle | **3.2 Split Data Transfer Method** | "MSB Don't Care" only | Version F: MSB Don't Care<br>Version E: MSB Care |
| Selecting a new mask data method during mask write cycle | **2.8.2 Selecting Mask Data** | Option reset cycle only | Both option reset cycle and new mask selection can be used. |

**μPD482235**

The table below summarizes the operation of each version of the μPD482235.

| Item | Reference | Version A | Versions F and E |
|---|---|---|---|
| Specifying a column for data transfer during split data transfer cycle | 3.2 **Split Data Transfer Method** | "MSB Don't Care" only | Version F: MSB Don't Care<br>Version E: MSB Care |
| Selecting a new mask data system during mask write cycle | 2.8.2 **Selecting Mask Data** | Option reset cycle only | Both option reset cycle and new mask selection can be used. |
| $\overline{OE}$ control system during hyper page mode cycle | **2.4.1 Setting the Output to the High Impedance State** | Latched control (conforming to JEDEC) | Non-latched control |

- **How to identify each version**

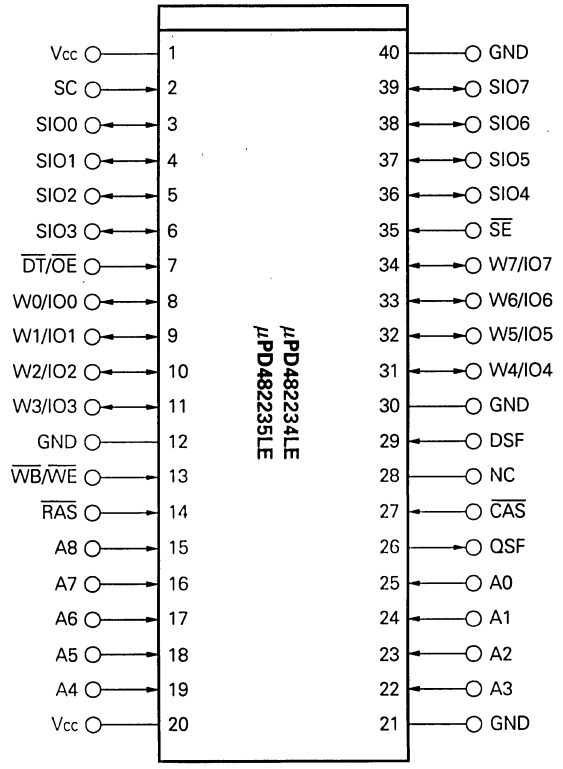
Each version is identified with its lot number (Refer to **7. Example of Stamping**).

**Ordering Information**

| Part Number | $\overline{RAS}$ Access Time ns (MAX.) | Package | Page Mode |
|---|---|---|---|
| μPD482234LE-60 | 60 | 40-pin plastic SOJ (400mil) | Fast page mode |
| μPD482234LE-70 | 70 | | |
| μPD482234LE-80 | 80 | | |
| μPD482234G5-60 | 60 | 44-pin plastic TSOP (II) (400mil) | |
| μPD482234G5-70 | 70 | | |
| μPD482234G5-80 | 80 | | |
| μPD482235LE-60 | 60 | 40-pin plastic SOJ (400mil) | Hyper page mode |
| μPD482235LE-70 | 70 | | |
| μPD482235LE-80 | 80 | | |
| μPD482235G5-60 | 60 | 44-pin plastic TSOP (II) (400mil) | |
| μPD482235G5-70 | 70 | | |
| μPD482235G5-80 | 80 | | |

## Pin Configurations (Marking Side)

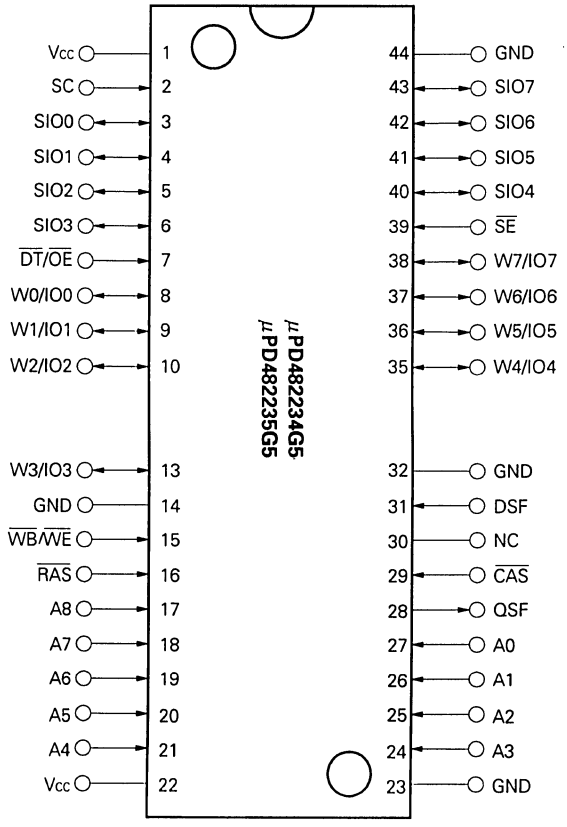
**40-pin plastic SOJ (400 mil)**




| | |
|---|---|
| A0 to A8 | : Address inputs |
| W0 to W7/IO0 to IO7 | : Mask data selects/Data inputs and outputs |
| SIO0 to SIO7 | : Serial data inputs and outputs |
| $\overline{RAS}$ | : Row address strobe |
| $\overline{CAS}$ | : Column address strobe |
| $\overline{DT}/\overline{OE}$ | : Data transfer/Output enable |
| $\overline{WB}/\overline{WE}$ | : Write-per-bit/Write enable |
| $\overline{SE}$ | : Serial data input/Output enable |
| SC | : Serial clock |
| QSF | : Special function output |
| DSF | : Special function enable |
| Vcc | : Power supply voltage |
| GND | : Ground |
| NC[Note] | : No connection |

**Note** Some signals can be applied because this pin is not connected to the inside of the chip.

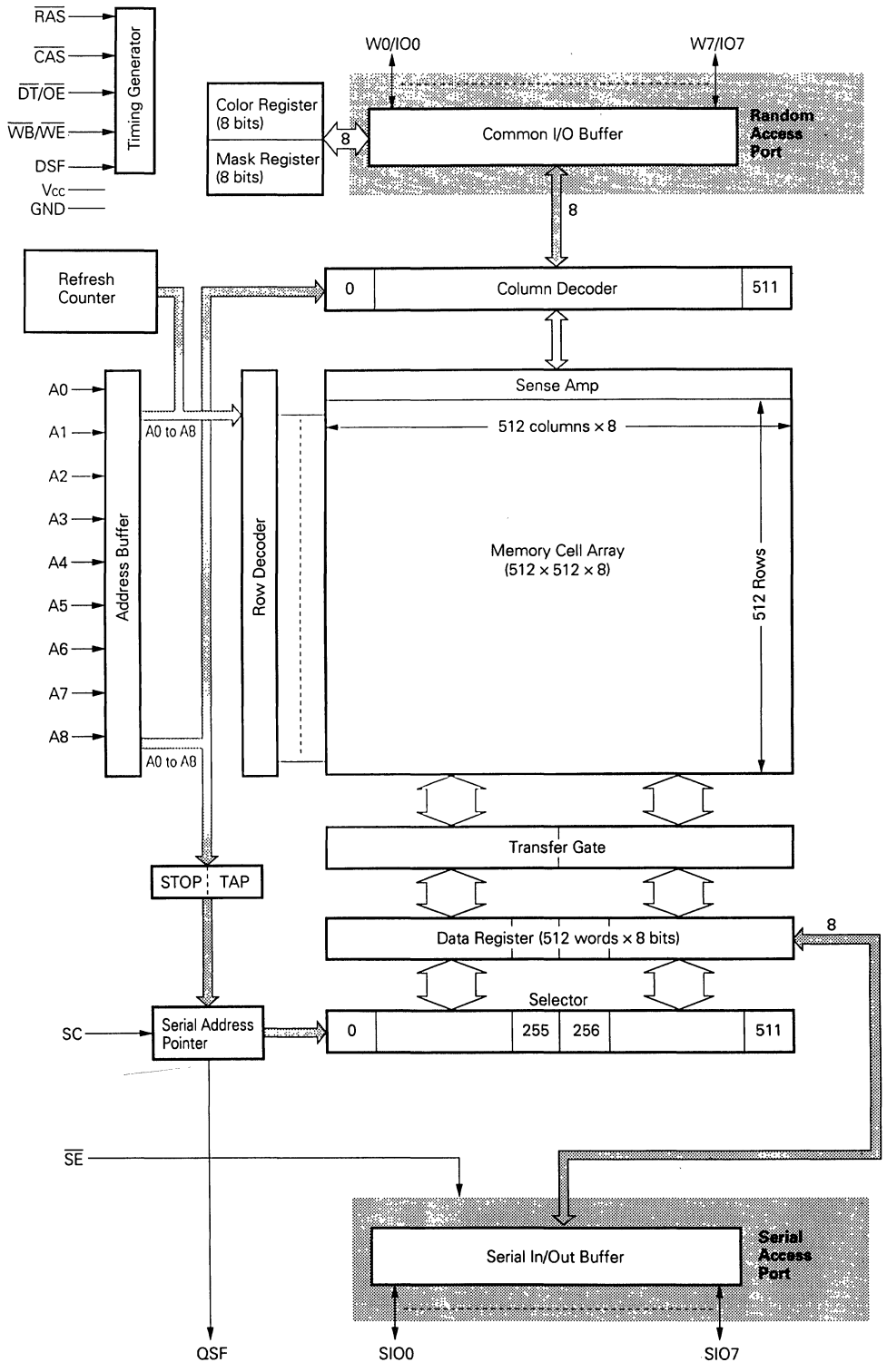
**44-pin plastic TSOP (II) (400 mil)**




| | | |
|---|---|---|
| A0 to A8 | : | Address inputs |
| W0 to W7/IO0 to IO7 | : | Mask data selects/Data inputs and outputs |
| SIO0 to SIO7 | : | Serial data inputs and outputs |
| $\overline{RAS}$ | : | Row address strobe |
| $\overline{CAS}$ | : | Column address strobe |
| $\overline{DT}/\overline{OE}$ | : | Data transfer/Output enable |
| $\overline{WB}$, $\overline{WE}$ | : | Write-per-bit/Write enable |
| $\overline{SE}$ | : | Serial data input/Output enable |
| SC | : | Serial clock |
| QSF | : | Special function output |
| DSF | : | Special function enable |
| Vcc | : | Power supply voltage |
| GND | : | Ground |
| NC[Note] | : | No connection |

**Note** Some signals can be applied because this pin is not connected to the inside of the chip.

## Block Diagram



## 1. Pin Functions

This product is equipped with the $\overline{RAS}$, $\overline{CAS}$, $\overline{WB}/\overline{WE}$, $\overline{DT}/\overline{OE}$, A0 to A8, DSF, SC, $\overline{SE}$ inputs, QSF output, and W0 to W7/IO0 to IO7, SIO0 to SIO7 input/output pins.

(1/3)

| Pin Name | Input/ Output | Function |
|---|---|---|
| $\overline{RAS}$ (Row address strobe) | Input | This signal latches the row addresses (A0 to A8), selects the corresponding word line, and activates the sense amplifier. It also refreshes the memory cell array of the one line (4,096 bits) selected from the row addresses (A0 to A8).<br><br>It also serves as the signal which selects the following operations.<br>• Write-per-bit • Flash write<br>• $\overline{CAS}$ before $\overline{RAS}$ refresh • Split data transfer |
| $\overline{CAS}$ (Column address strobe) | | This signal latches the column addresses (A0 to A8), selects the digit line connecting the sense amplifier, and activates the output circuit which outputs data to the random access port.<br><br>It also serves as the signal which selects the following operations.<br>• Read/write • Block write<br>• Color register set • Mask register set |
| A0 to A8 (Address inputs) | | These are the address input pins, TAP register input pins, and STOP register input pins.<br><br>Address Input<br>This is a 9-bit address bus. It inputs a total of 18 bits of the address signal, starting from the upper 9 bits (row address) and then followed by the lower 9 bits (column bits) (address multiplex method). Using these, one word memory cells (8 bits) are selected from the 262,144 words × 8 bits memory cell array.<br>During use, specify the row address, activate the $\overline{RAS}$ signal, latch the row address, switch to the column address, and activate the $\overline{CAS}$ signal. After activating the $\overline{RAS}$ and $\overline{CAS}$ signals, each address signal is taken into the device. For this reason, the address input setup time ($t_{ASR}$, $t_{ASC}$) and hold time ($t_{RAH}$, $t_{CAH}$) are specified for activating the $\overline{RAS}$ and $\overline{CAS}$ signals.<br><br>TAP Register Input<br>In the data transfer cycle, this TAP register input pin functions as the address input pin which selects the memory cell for transferring (9 bits are latched at the falling edge of $\overline{RAS}$) and the TAP register data input pin which specifies the start addresses of the serial read/write operation after data transfer (9 bits are latched at the falling edge of the $\overline{CAS}$).<br><br>STOP Register Input<br>This pin functions as the STOP register input pin when the STOP register is set (STOP register data (9 bits) are latched at the falling edge of the $\overline{RAS}$.) |

(2/3)

| Pin Name | Input/ Output | Function |
|---|---|---|
| $\overline{DT}/\overline{OE}$ (Data transfer/ output enable) | Input | These are the data transfer control signal and read operation control signal respectively. They have different functions in the data transfer cycle and read cycle.<br><br>**Data transfer control signal (in data transfer cycle)**<br>The data transfer cycle is initiated when a low level is input to this pin at the falling edge of $\overline{RAS}$.<br><br>**Read operations control signal (in read cycle)**<br>Read operation is performed when this signal, and the $\overline{RAS}$ and $\overline{CAS}$ signals are activated. The input/output pin is high impedance when this signal is not activated. When the $\overline{WB}/\overline{WE}$ signal is activated while the $\overline{DT}/\overline{OE}$ signal is activated, the $\overline{DT}/\overline{OE}$ signal is invalid in the memory and read operations cannot be performed. |
| $\overline{WB}/\overline{WE}$ (Write-per-bit/ Write enable) | | These are the write operation control signal and mask write cycle (write-per-bit function) mask data input control signal, respectively.<br>When this signal, $\overline{RAS}$ and $\overline{CAS}$ signals are activated, write operations or mask write can be performed. These mode are determined by the level of $\overline{WB}/\overline{WE}$ at the falling edge of $\overline{RAS}$.<br>• High level .....8-bit write cycle<br>• Low level.....Mask write cycle (Write-per-bit) |
| DSF (Special function enable) | | This signal controls the selection of functions.<br>The selection of functions is determined by the level of this signal at the falling edge of the $\overline{RAS}$ and $\overline{CAS}$. The functions will change as follows when this signal is high level.<br>• The data transfer cycle changes to a split data transfer cycle.<br>• The read/write cycle of each $\overline{RAS}$ clock changes to the flash write cycle.<br>• The write cycle of each $\overline{CAS}$ clock changes to the block write cycle. |
| W0 to W7/IO0 to IO7 (Mask data selects/ Data inputs, outputs) | Input/ Output | These are normally 8-bit data bus and are used for inputting and outputting data. (IO0 to IO7).<br>Function as the mask data input pins (W0 to W7) in the mask write cycle (write-per-bit function).<br>Write operations can be performed only for W0 to W7 that are input with a high level at the falling edge of $\overline{RAS}$ (new mask data).<br>Functions as the column selection data input pin in the block write cycle. |

(3/3)

| Pin Name | Input/ Output | Function |
|---|---|---|
| SC (Serial clock) | Input | This pin inputs the clock which controls the serial access port operation.<br><br>**Serial Read**<br>The data of the data register which is synchronized with the rising edge of the SC are output from the SIO0 to SIO7 pins and kept until the next SC rising edge.<br><br>**Serial Write**<br>The data from the SIO0 to SIO7 pins are latched at the rising edge of the SC and written in the data register. |
| $\overline{SE}$ (Serial data input/ output enable) | | This is a control pin for the serial access port input/output buffer.<br>It controls data output during serial reading and controls data input during serial writing.<br>By inputting the serial clock, the serial pointer will operate even if $\overline{SE}$ has not been activated (high level input). |
| SIO0 to SIO7 (Serial data inputs/ outputs) | Input/ Output | These are the serial data input and output pins of the serial access port. |
| QSF (Special function output) | Output | This is a position discrimination pin of the serial pointer (upper side or lower side).<br>Which side is being serial accessed (upper side or lower side) can be discriminated according to the output of this pin.<br>• High level..... Upper side (Addresses 256 to 511)<br>• Low level..... Lower side (Addresses 0 to 255) |

## 2. Random Access Port Operations

The operation mode is determined by the $\overline{CAS}$, $\overline{DT/OE}$, $\overline{WB/WE}$, and DSF level at the falling edge of $\overline{RAS}$ and DSF level at the falling edge of $\overline{CAS}$.

**Table 2-1. Operation Mode**

| $\overline{RAS}$ Falling Edge | | | | $\overline{CAS}$ Falling Edge | Operation Mode | |
|---|---|---|---|---|---|---|
| $\overline{CAS}$ | $\overline{DT/OE}$ | $\overline{WB/WE}$ | DSF | DSF | | |
| H | H | × | L | × | Read/Write Cycle | Read cycle |
| H | H | H | L | L | | Write cycle |
| H | H | H | L | H | | Block write cycle |
| H | H | L | L | L | | Mask write cycle (New mask/Old mask)[Note 1] |
| H | H | L | L | H | | Block mask write cycle (New mask/Old mask)[Note 1] |
| H | H | H | H | H | | Color register set cycle |
| H | H | H | H | L | | Write mask register set cycle |
| H | H | L | H | × | | Flash write cycle (New mask/Old mask)[Note 1] |
| H | L | H | L | × | Data Transfer Cycle | Single read data transfer cycle |
| H | L | H | H | × | | Split read data transfer cycle |
| H | L | L | L | × | | Single write data transfer cycle (New mask/Old mask)[Note 1] |
| H | L | L | H | × | | Split write data transfer cycle (New mask/Old mask)[Note 1] |
| L | × | × | L | × | Refresh Cycle | $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (Option reset)[Note 1, 2] |
| L | × | H | H | × | | $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (No reset) |
| L | × | L | H | × | | $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (STOP register set)[Note 2] |
| H | H | × | L | × | | $\overline{RAS}$ only refresh cycle |

**Notes 1.** Observe the following conditions when using the new mask data or old mask data in these cycles.

**(1) Old mask data**

Can be used after setting the mask data using the write mask register set cycle.

**(2) New mask data**

Can be used after selecting the new mask register using the write mask register set cycle (new mask selection) (Unusable in version A) and after the optional reset cycle.

**2.** The STOP register is set to "FFH (11111111)" by the optional reset cycle.

**Remark** H : High level, L : Low level, × : High level or low level

### 2.1 Random Read Cycle

This product has a common 8-bit input/output pin. To output data, specify the address using the $\overline{RAS}$ and $\overline{CAS}$ clocks and then set $\overline{DT}/\overline{OE}$ to low level.

The data output will be kept until one of the following conditions is set.

(1) Set $\overline{RAS}$ and $\overline{CAS}$ to high level
(2) Set $\overline{DT}/\overline{OE}$ to high level
(3) Set $\overline{WB}/\overline{WE}$ to low level

The read cycle and data transfer cycle are differentiated according to the level of $\overline{DT}/\overline{OE}$ at the falling edge of the $\overline{RAS}$ clock. If $\overline{DT}/\overline{OE}$ is set to low level at the falling edge of the $\overline{RAS}$ clock, data transfer cycle operations will be initiated. Therefore, to set the read cycle, input a high level above $t_{DHH}$ (MIN.) to $\overline{DT}/\overline{OE}$ from the falling edge of the $\overline{RAS}$ clock, and then input a low level.

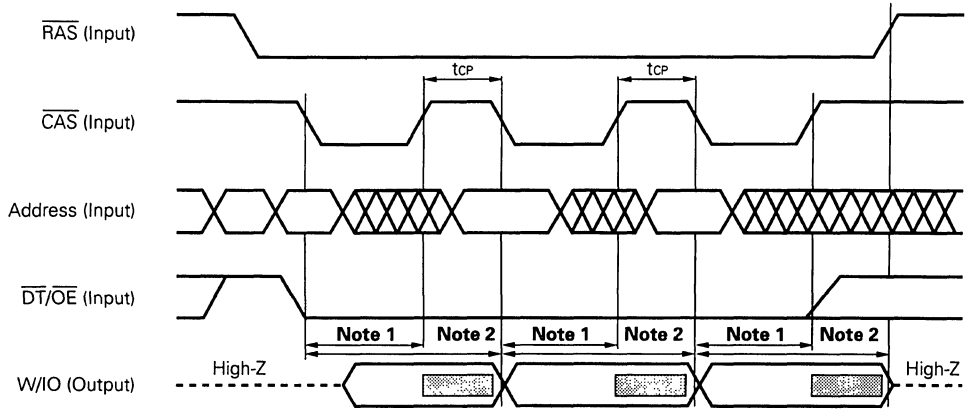
**Caution Set the DSF to low level at the falling edge of $\overline{RAS}$. If set to high level, the memory cell data cannot be output.**

#### 2.1.1 Extended Read Data Output (μPD482235)

The μPD482235 adopt the hyper page mode cycle which is a faster read/write cycle than the fast page mode of the μPD482234 (Hyper page mode cycle time: 30 ns (MIN.)).

With this cycle, the read data output can be kept until the next $\overline{CAS}$ cycle, and because the output is extended, the minimum cycle can easily be used. For example, by fixing $\overline{DT}/\overline{OE}$ at low level after dropping $\overline{RAS}$ and executing the hyper page read cycle, each time the column address is latched at the falling edge of $\overline{CAS}$, the data output will be updated and kept until the next falling edge of $\overline{CAS}$. As a result, the output will be extended only during $\overline{CAS}$ precharge time ($t_{CP}$) as compared to the normal fast page mode.

**Figure 2-1. Extended Data Output of Hyper Page Mode**




**Notes 1.** Time during which the output data is kept in the fast page read cycle.

**2.** Time during which the output data is kept in the hyper page read cycle (▒ part: Extended data output).

### 2.2 Random Write Cycle (Early Write, Late Write, Read Modify Write)

There are three types of random write cycles-the early write, late write, and read modify write. To use these cycles, activate the $\overline{RAS}$ and $\overline{CAS}$ clocks and set $\overline{WB}/\overline{WE}$ to low level.

The $\overline{WB}/\overline{WE}$ also controls the mask data for the write-per-bit function (mask write cycle). Therefore, when performing the normal write cycle which does not use the write-per-bit function, set this pin to high level at the falling edge of the $\overline{RAS}$ clock.

#### 2.2.1 Early Write Cycle

The early write cycle controls data writing according to the $\overline{CAS}$ clock.

To execute this cycle, set $\overline{WB}/\overline{WE}$ to low level earlier than the $\overline{CAS}$ clock. The write data is taken into the device at the falling edge of the $\overline{CAS}$ clock.

#### 2.2.2 Late Write Cycle

The late write cycle controls data writing according to the $\overline{WE}$ clock.

To execute this cycle, set $\overline{WB}/\overline{WE}$ to low level later than the $\overline{CAS}$ clock. The write data is taken into the device at the falling edge of $\overline{WB}/\overline{WE}$. To set the output to high impedance at this time, keep $\overline{DT}/\overline{OE}$ at high level until $\overline{WB}/\overline{WE}$ is input.

#### 2.2.3 Read Modify Write Cycle

The read modify write cycle performs data reading and writing in one $\overline{RAS}$ and $\overline{CAS}$ cycle.

To execute this cycle, delay $\overline{WB}/\overline{WE}$ from the late write cycle by $t_{RWD}$ (MIN.), $t_{CWD}$ (MIN.), and $t_{AWD}$ (MIN.). Follow the $t_{OEZ}$ and $t_{OED}$ specifications so that the output data and input data do not clash in the data bus. The data after modification can be input after more than $t_{OED}$ (MIN.) from the rising edge of $\overline{DT}/\overline{OE}$.

### 2.3 Fast Page Mode Cycle (μPD482234)

The μPD482234 adopts the fast page mode. This mode accesses memory cells in the same row array in about 1/3 of the time taken by the normal random read/write cycle. This fast page mode cycle is executed by repeating the $\overline{CAS}$ clock cycle more than two times while the $\overline{RAS}$ clock is being activated. In this mode read, write and read modify write cycles are available for each of the consecutive $\overline{CAS}$ cycles within the same $\overline{RAS}$ cycle.

### 2.4 Hyper Page Mode Cycle (μPD482235)

The μPD482235 adopts a hyper page mode cycle which is a faster read/write cycle than the fast page mode of the μPD482434 (Hyper page mode cycle time: 30 ns (MIN.)).

In this cycle, because the read data output is kept until the following $\overline{CAS}$ cycle and as a result, the output is extended, the minimum cycle can easily be used. The output is extended compared to the normal fast page mode of μPD482234. Refer to **2.1.1 Extended Read Data Output.**

#### 2.4.1 Setting the Output to the High Impedance State

The hyper page mode can use one of three methods of setting the output pin to the high impedance state depending on the version; these methods are $\overline{WE}$ control, $\overline{OE}$ control (latched control), and $\overline{OE}$ control (non-latched control).

**Table 2-2** lists the correspondence between the output control types and versions A, F, and E.

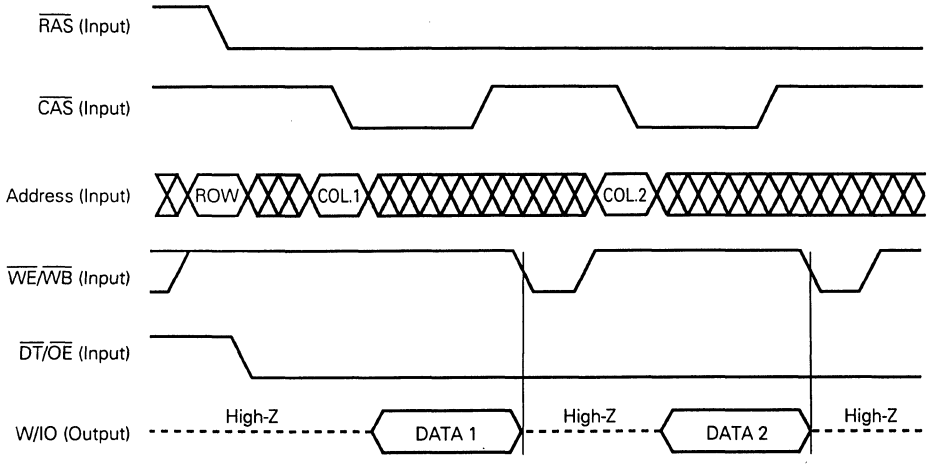
**Table 2-2. Correspondence between Each Output Control Method and Versions A, F, and E**

| Output control | Version A | Version F | Version E |
|---|---|---|---|
| $\overline{WE}$ control | usable | usable | usable |
| $\overline{OE}$ control (latched control), conforming to JEDEC | usable | unusable | unusable |
| $\overline{OE}$ control (non-latched control) | unusable | usable | usable |

**(1) $\overline{WE}$ control (usable in all versions)**

After a high level is input to $\overline{CAS}$, when a pulse conforming to the $t_{WEZ}$ specification is supplied to the $\overline{WE}$ pin ($\overline{WE}$ = enable) during the same $\overline{CAS}$ cycle, the W/IO pin is held in the high impedance state until the next $\overline{CAS}$ cycle.

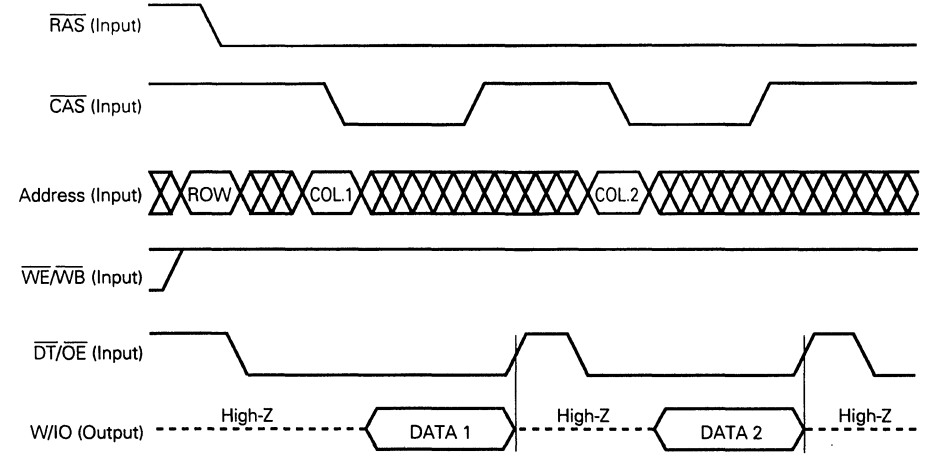
**Figure 2-2. $\overline{WE}$ Control Output Control**




**(2) $\overline{OE}$ control (latched control) (usable in version A)**

After a high level is input to $\overline{CAS}$, when a high level is supplied to the $\overline{OE}$ pin ($\overline{OE}$ = disable) during the same $\overline{CAS}$ cycle, the W/IO pin is held in the high impedance state until the next $\overline{CAS}$ cycle. This specification enables efficient use of $\overline{OE}$ interleaving during parallel connection.

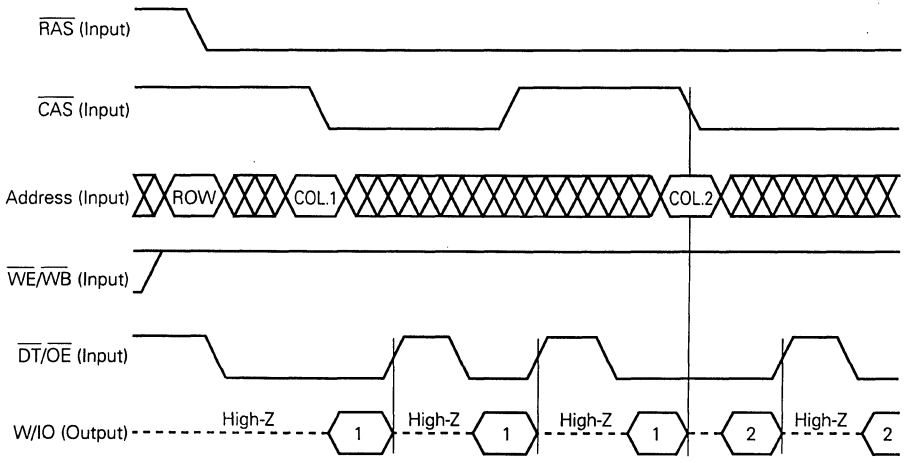
**Figure 2-3. $\overline{OE}$ Control Output Control (Latched Control)**



**(3) $\overline{OE}$ control (non-latched control) (usable in versions F and E)**

When a high level is input to the $\overline{OE}$ pin ($\overline{OE}$ = disable) during the same $\overline{CAS}$ cycle, the W/IO pin enters the high impedance state. If a low level is input to the $\overline{OE}$ pin again before the next $\overline{CAS}$ cycle ($\overline{OE}$ = enable), the data at the same address is output to the W/IO pin again.

**Figure 2-4. $\overline{OE}$ Control Output Control (Non-Latched Control)**



### 2.5 Flash Write Cycle

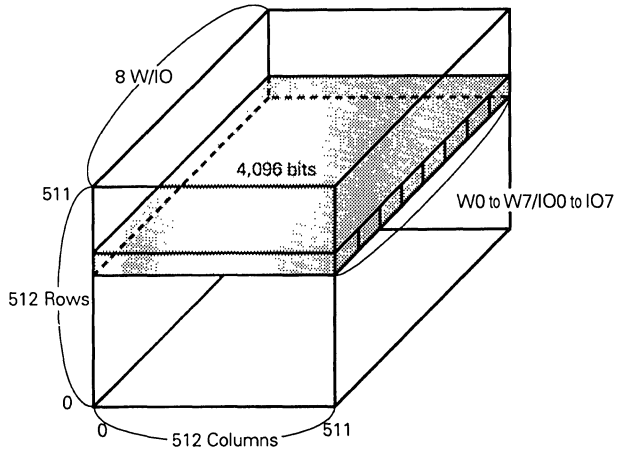
This cycle writes the color register data in a 4,096-bit memory cell in one cycle. The memory cell range for one flash write cycle is 512 columns on the same row address (512-column × 8 · IO = 4,096 bits).

#### 2.5.1 Execution of Flash Write Cycle

To execute the flash write cycle, set $\overline{WB}/\overline{WE}$ to low level at the falling edge of $\overline{RAS}$.

By using the write-per-bit function (new mask data/old mask data), only the required W/IO can be selected and written.

**Figure 2-5. Memory Cell Range That Can be Written with Flash Write Cycle**




**Remark** ▭ is the memory cell range that can be written in one flash write cycle.

### 2.6 Block Write Cycle

This cycle writes the color register data in 32-bit memory cell in one cycle. The memory cell range in which data can be written in one block write cycle is four continuous columns on one row address (4-column × 8 · IO = 32 bits).

Any column of the four columns can be selected and writing prohibited. Determine whether to write or prohibit writing according to the data selected for column.

#### 2.6.1 Free Column Selection

Determine which column to select according to the W/IO pin to which the data selected for the column is to be input.

The four columns (1st to 4th) correspond to W0 to W3/IO0 to IO3 to which the data selected for column will be input (The following table shows the 1st to 4th columns specified by A0 and A1 and the corresponding W/IO pins to which the data selected will be input.).

#### 2.6.2 Column Select Data

Input column select data for every four columns at the 32 bits (4-column × 8 · IO). The data will be written if the column select data is "1". Writing will be prohibited if the column select data is "0"

#### 2.6.3 Execution of Block Write Cycle

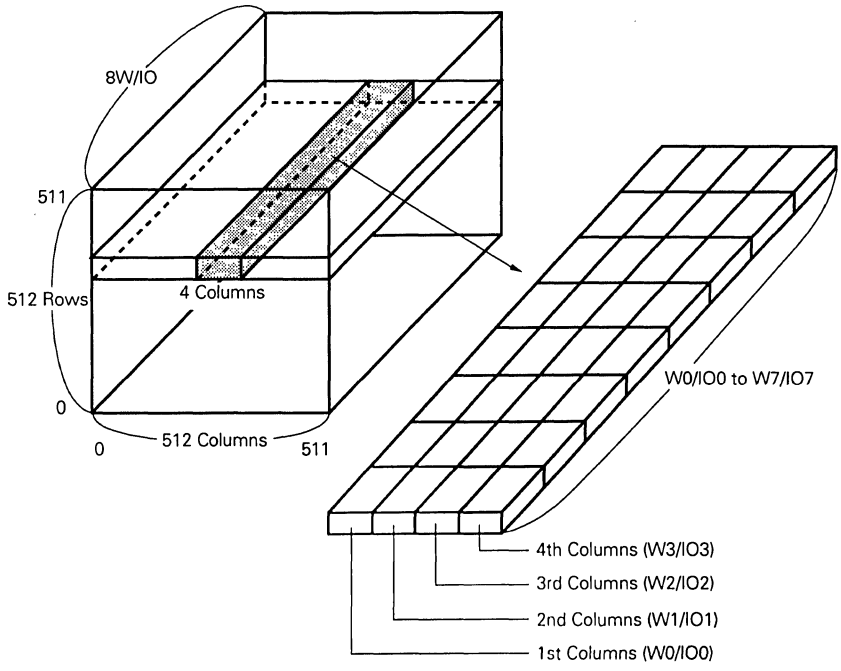
At the falling edge of the slowest signal ($\overline{CAS}$, $\overline{WB}/\overline{WE}$), input the "1" column select data or "0" column select data to W0 to W3/IO0 to IO3 corresponding to columns 1st to 4th.

By using the write-per-bit (new mask data/old mask data) function, only the required W/IO can be selected and written.

**Table 2-3. I/O Pins Input with Column Select Data Corresponding to Columns 1st to 4th**

| Selected 4 Columns | Column Address and Corresponding W/IO Pin | | | Column Select Data | Writing |
|---|---|---|---|---|---|
| | A1 | A0 | IO | | |
| 1st column | 0 | 0 | IO0 | 1 | Yes |
| | | | | 0 | No |
| 2nd column | 0 | 1 | IO1 | 1 | Yes |
| | | | | 0 | No |
| 3rd column | 1 | 0 | IO2 | 1 | Yes |
| | | | | 0 | No |
| 4th column | 1 | 1 | IO3 | 1 | Yes |
| | | | | 0 | No |

**Figure 2-6. Memory Cell Range That Can be Written in Block Write Cycle**




**Remarks 1.** ▒ is the memory cell range that can be written in one block write cycle.

**2.** ( ) is the W/IO pin input with the column select data.

### 2.7 Register Set Cycle (Color Register, Write Mask Register)

This cycle writes data in the color register and write mask register. To execute the register set cycle, set $\overline{CAS}$, $\overline{DT/OE}$, $\overline{WB/WE}$ and DSF to high level at the falling edge of $\overline{RAS}$. Determine which register to select according to the DSF level at the falling edge of $\overline{CAS}$.

The register set cycle also serves as the $\overline{RAS}$ only refresh cycle.

**Table 2-4. Register Selection**

| DSF level at $\overline{CAS}$ falling edge | Selected register |
|---|---|
| High level | Color register |
| Low level | Write mask register |

**Caution After selecting the write mask register and writing the mask data, the write-per-bit function in the mask write cycle will be set for the old mask register. Refer to 2.8.1 Write-Per-Bit Function.**

### 2.8 Mask Write Cycle

Cycles that use the write-per-bit function during the random write cycle, flash write cycle, block write cycle, write data transfer cycle, are called mask write cycles. In the fast page/hyper page mode write cycle, the mask data cannot be changed during the $\overline{CAS}$ cycle.

#### 2.8.1 Write-Per-Bit Function

The write-per-bit function writes data using the mask data only in the required IO-pin. It writes when the mask data is "1" and prohibits writing when the data is "0".

**Table 2-5. Mask Data Selection**

| W Pin | Mask Data | Writing |
|---|---|---|
| W0 to W7 | 1 | Yes |
| | 0 | No |

#### 2.8.2 Selecting Mask Data

There are two ways of selecting mask data. One is the new mask data method and the other is the old mask data method.

With the new mask data method, new mask data is set in the cycle writing. With the old mask data, mask data set in the write mask register is used.

**(1) New Mask Data Method**

The new mask data method consists of the write mask register set cycle (new mask selection) method and $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (optional reset cycle) method.

However, only the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (option reset cycle) can be used in version A. **Table 2-6** lists how to select the new mask data method for each version.

**Table 2-6. New Mask Data Selection Method for Each Version**

| New mask data selection method | Version A | Version F | Version E |
|---|---|---|---|
| Method to use the write mask register set cycle (new mask selection) | unusable | usable | usable |
| Method to use the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (option reset cycle.) | usable | usable | usable |

**(a) Method Using Write Mask Register Set Cycle (New Mask Selection) (Versions F and E)**

This method is usable in both version F and version E.

To switch to the mode using new mask data, keep the $\overline{DT}/\overline{OE}$, $\overline{WB}/\overline{WE}$ DSF to high level and set the $\overline{CAS}$ and DSF to high level at the falling edge of $\overline{RAS}$, the DSF to low level at the falling edge of $\overline{CAS}$, and start up the next $\overline{RAS}$ and $\overline{CAS}$ after the $t_{CAS}$ and $t_{RAS}$.

As a result, the write-per-bit function can be used using the new mask data from the next mask write cycle.

**(b) Method Using $\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (Optional Reset Cycle) (All versions)**

This method is usable in all versions.

To switch to the mode using new mask data, set the DSF to low level at the falling edge of $\overline{CAS}$ in the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle.

As a result, the write-per-bit function can be used using the old mask data from the next mask write cycle.

**(2) Old Mask Data Method**

To switch to the mode using old mask data, set the DSF to low level at the falling edge of $\overline{CAS}$ in the write mask register set cycle, and write the mask data in the write mask register.

As a result, the write-per-bit function can be used using the old mask data from the next mask write cycle.

### 2.8.3 Execution of Mask Write Cycle

To execute the write-per-bit function, select the new mask data method or old mask data method, and set $\overline{WB}/\overline{WE}$ to low level at the falling edge of $\overline{RAS}$ of each write cycle. At this time, input the mask data to the W pin in the write cycle using the new mask data. In the write cycle using the old mask data, as the mask data set to the write mask register will be used, there is no need to input the mask data to the W pin.

This function is valid only at the falling edge of $\overline{RAS}$. In the fast page/hyper page mode write cycle, the mask data determined in the first $\overline{RAS}$ cycle for moving onto the next fast page/hyper page mode will be valid while the fast page/hyper page mode write cycle continues.

### 2.9 Refresh Cycle

The refresh cycle of this product consists of the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle and refresh cycle using external address inputs ($\overline{RAS}$ only refresh and read/write refresh). The refresh period is the same as the DRAM (Standard), 512 cycles/8 ms.

#### 2.9.1 Refresh Cycle Using External Address Input ($\overline{RAS}$ Only Refresh and Read/Write Refresh)

By specifying the row address using the 9 bits between A0 to A8 at the falling edge of $\overline{RAS}$, setting $\overline{CAS}$ and $\overline{DT}/\overline{OE}$ to high level, and keeping $\overline{CAS}$ at high level while $\overline{RAS}$ is low level, the memory cells on the specified row address (512 × 8 bits) can be refreshed. At this time, refresh is executed, W0 to W7/IO0 to IO7 pins are kept at high impedance, and information such as memory contents, register data, function settings, etc. are all also kept.

At the falling edge of $\overline{RAS}$, all cycles whose $\overline{CAS}$ are high level input the external address. Therefore, in addition to the read/write cycle operations, etc. refresh operations similar to the $\overline{RAS}$ only refresh operations will be performed. For this reason, in systems in which addresses in the memory are always increased or decreased, it may not be necessary to perform refresh again.

When several devices exist on one bus, data will clash in the bus during the above read/write operations unless each device is equipped with a buffer. Consequently, as it is necessary to set the I/O line to high impedance beforehand during refresh, normally the $\overline{RAS}$ only refresh operation is used.

#### 2.9.2 $\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (Including Hidden Refresh)

When $\overline{CAS}$ is set to low level at the falling edge of $\overline{RAS}$, the refresh address is supplied from the internal refresh address counter. The internal refresh address counter is increased automatically each time this refresh cycle is executed.

During this refresh cycle, functions of random access port and serial access port are selected as follows according to the DSF and $\overline{WB}/\overline{WE}$ levels at the falling edge of $\overline{RAS}$.

**(1) When DSF is low level: Optional reset**

All STOP register data become "1" and the mask write cycle switches to the new mask data method.

**(2) When DSF is high level and $\overline{WB}/\overline{WE}$ is low level: STOP register set**

The STOP register data is input from the A0 to A7 pins at the falling edge of $\overline{RAS}$.

**(3) When DSF, $\overline{WB}/\overline{WE}$ is high level: No reset**

Only refresh operations are performed and the function selection state is kept.

In all cases, the W/IO pin is kept at high impedance. When $\overline{CAS}$ and $\overline{DT}/\overline{OE}$ are kept low level while the mode is changed to the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle following the read cycle, and $\overline{RAS}$ is activated, the hidden refresh cycle will be initiated. In this cycle, the W/IO pin does not become high impedance and the data read in the former read cycle will be kept as it is.

Because internal memory operations are equivalent to $\overline{CAS}$ before $\overline{RAS}$ refresh, no external addresses are required.

Like $\overline{CAS}$ before $\overline{RAS}$ refresh, in the hidden cycle, functions will be selected according to the level of DSF, $\overline{WB}/\overline{WE}$ at the falling edge of $\overline{RAS}$. Operations are guaranteed when DSF is low level and when DSF, $\overline{WB}/\overline{WE}$ are high level.

### 3. Serial Access Port Operations

There are two types of data transfer cycles-data transfer from the random access port to the serial access port (read data transfer) and data transfer from the serial access port to the random access port (write data transfer). There are also two types of data transfer methods-single data transfer and split data transfer.

To set the data transfer cycle, input high level to $\overline{CAS}$ and input low level to $\overline{DT}/\overline{OE}$ at the falling edge of $\overline{RAS}$. The data transfer type differs according to the input levels of $\overline{WB}/\overline{WE}$, and DSF at the falling edge of $\overline{RAS}$.

**Table 3-1. Serial Access Port Operation Mode**

| At $\overline{RAS}$ Falling Edge | | | | Data Transfer Type | Transfer Direction | |
|---|---|---|---|---|---|---|
| $\overline{CAS}$ | $\overline{DT}/\overline{OE}$ | $\overline{WB}/\overline{WE}$ | $\overline{DSF}$ | | Transfer Source | Transfer Destination |
| H | L | H | L | Single read data transfer | Random access port | Serial access port |
| H | L | H | H | Split read data transfer | | |
| H | L | L | L | Single mask write data transfer[Note] | Serial access port | Random access port |
| H | L | L | H | Split mask write data transfer[Note] | | |

**Note** Write-per-bit function can be specified.

**Remark** H: High level, L: Low level

### 3.1 Single Data Transfer Method

With this method, 512 words × 8 bits (whole memory range of serial access port) data is transferred at one time. This method can be used in both write data transfer and read data transfer.

#### 3.1.1 Single Read Data Transfer Cycle

This cycle transfers the 4K-bit (512 words × 8 bits) data of the random access port to the serial access port in one cycle.

**(a) Setting of Single Read Data Transfer Cycle**

To set the data transfer cycle, input a high level to $\overline{CAS}$ and $\overline{WB}/\overline{WE}$ and low level to $\overline{DT}/\overline{OE}$ and DSF at the falling edge of $\overline{RAS}$.

Using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$, the memory cells (512 words × 8 bits) of the transfer source of the random access port can be selected. The address data input to A0 to A8 at the falling edge of $\overline{CAS}$ will be latched as the TAP register data of serial access port. Refer to **3.4 TAP Register.**

**(b) Execution of Single Read Data Transfer Cycle**

To execute the data transfer cycle, set the single read data transfer cycle and then input a high level to $\overline{DT}/\overline{OE}$ and $\overline{RAS}$.

When SC is active (edge control), data transfer will be executed at the rising edge of $\overline{DT}/\overline{OE}$. When SC is inactive (self control), it will be executed at the rising edge of $\overline{RAS}$. At the same time, the serial address pointer jumps to the start column (TAP) of the next serial read cycle, and the TAP register will be set the empty state.

After the transfer is completed, the new serial access port data is output after $t_{SCA}$ following the rise of the SC clock that occurs after $t_{SDH}$ if the SC is active, and after $t_{SDHR}$ if SC is inactive.

**Caution When the single read data transfer cycle is executed while the serial access port is performing serial write operations, the serial access port will start serial read operations at the rising edge of $\overline{RAS}$. Refer to 4. Electrical Characteristics Read Data Transfer Cycle (Serial Write → Serial Read Switching) Timings.**

### 3.1.2 Single Mask Write Data Transfer Cycle

This cycle transfers 4K-bit (512 word × 8 bits) data of the serial access port to the random access port in one cycle. Because $\overline{WB}/\overline{WE}$ is low level at the falling edge of $\overline{RAS}$, the write-per-bit function always functions in this transfer cycle. Refer to **2.8 Mask Write Cycle.**

**(a) Setting of Single Mask Write Data Transfer Cycle**

To set this cycle, latch the data to be transferred to the serial access port, and then input a high level to $\overline{CAS}$ and low level to $\overline{DT}/\overline{OE}$, $\overline{WB}/\overline{WE}$, and DSF at the falling edge of $\overline{RAS}$. Because the write-per-bit function functions in this transfer operation, for the new mask data method, the mask data must be supplied to W0 to W7 at the falling edge of $\overline{RAS}$, and for the old mask data method, there is no need to control the mask data.

The memory cells (512 words × 8 bits) of the transfer destination of the random access port are selected using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$. The address data input to A0 to A8 at the falling edge of $\overline{CAS}$ is input as the TAP register data of serial access port. Refer to **3.4 TAP Register.**

**(b) Execution of Single Mask Write Data Transfer Cycle**

To execute this cycle, set the single write data transfer cycle and then input high level to $\overline{RAS}$. Data will be transferred at the rising edge of $\overline{RAS}$. At the same time, the serial address pointer jumps to the start column (TAP) of the next serial write cycle, and the TAP register will be set the empty state.

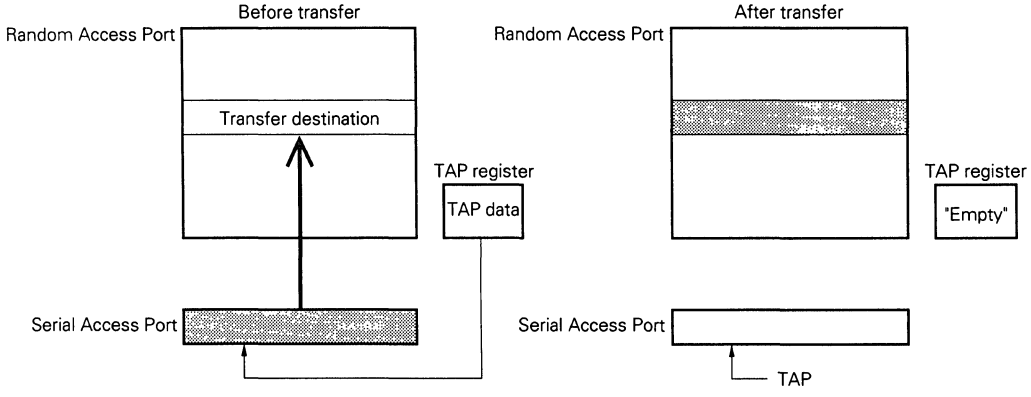
After the transfer is completed, the new serial access port data is latched at the rising edge of the SC clock that occurs after $t_{SDHR}$.

**Caution 1. When the single mask write data transfer cycle is executed while the serial access port is performing serial read operations, the serial access port will start serial write operations at the rising edge of $\overline{RAS}$. Refer to 4. Electrical Characteristics Write Data Transfer Cycle (Serial Read › Serial Write Switching) Timings.**

**2. Always make $\overline{CAS}$ low level in the write data transfer cycle and latch TAP. If write data transfer is performed without setting TAP, serial access port operations cannot be ensured until either one of the following points. If the SC clock is input during this time, the serial register value also cannot be guaranteed.**

- **Until the falling edge of $\overline{CAS}$ during the write data transfer cycle**
- **Until the read data transfer cycle is executed again**

**Figure 3-1. Single Write Data Transfer and TAP Operation**



### 3.2 Split Data Transfer Method

With this method, the 512 words × 8 bits (whole memory range of serial access port) data is divided into the lower column (0 to 255) and upper column (256 to 511), each consisting of 256 words × 8 bits.

Because the columns are divided into upper and lower columns with this method, data transfer can be performed on lower column (or upper column) while performing read/write operations in the upper column (or lower column). For this reason, transfer timing design is easy. This transfer method can be used in both write data transfer and read data transfer.

This transfer method uses either of two modes, "MSB Care" and "MSB Don't Care," depending on the method used to select the column for data transfer (See the table below.). The mode to be used varies from one version to another. Refer to **7. Example of Stamping** for how to identify each version.

**Table 3-2. Differences between the MSB Don't Care and MSB Care Modes**

| Mode | Version | Function |
|---|---|---|
| MSB Don't Care | Versions A and F | A8 input as TAP is ignored, and an inactive column is selected automatically for data transfer. |
| MSB Care | Version E | The column (upper or lower) is specified for data transfer according to A8 input as TAP. |

#### 3.2.1 Split Read Data Transfer Cycle (Versions A and F: For "MSB Don't Care")

This cycle divides the 4K-bit (512 words × 8 bits) data of the random access port into the lower and upper columns and transfers them to the serial access port.

In this cycle, the serial read/write can be performed in the columns to which data is not transfer.

**(a) Setting of Split Read Data Transfer Cycle**

To set this cycle, input a high level to $\overline{CAS}$, $\overline{WB}/\overline{WE}$ and DSF, and low level to $\overline{DT}/\overline{OE}$ at the falling edge of $\overline{RAS}$.

The memory cells (512 words × 8 bits) of the transfer source of the random access port are selected using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$. And the address data input to A0 to A7 at the falling edge of $\overline{CAS}$ is latched as the TAP register data of serial access port. Refer to **3.4 TAP Register**. There is no need to control address data input to A8 (See the table below.).

**Table 3-3. Relationships among Data Register Transfer Destination, A8 Inputs, and QSF Outputs (Split Read Data Transfer Cycle in the MSB Don't Care Mode)**

| A8 data input to the TAP register | QSF output | Data register transfer destination |
|---|---|---|
| A8 = × | QSF = 1 | Lower column (addresses 0 to 255) |
| | QSF = 0 | Upper column (addresses 256 to 511) |

**Remark** 1 = high level; 0 = low level; × = high or low level

**(b) Execution of Split Read Data Transfer Cycle**

To execute this cycle, set the split read data transfer cycle and then input the high level to $\overline{RAS}$. Data will be transferred at the rising edge of $\overline{RAS}$. Data is transferred from the random access port to the serial access port automatically at the column side (Column not pointed to by the serial address pointer) where serial access port is inactive. To confirm the transferred column side, check the output state of the QSF pin. Refer to **3.3.3 QSF Pin Output.**

When the serial address pointer comes to the jump source address specified by the STOP register, the serial address pointer jumps to the start column (TAP) of the serial read/write cycle at the inactive column side, and the TAP register will be set the empty state.

### 3.2.2 Split Read Data Transfer Cycle (Versions E: For "MSB Care")

This cycle divides the 4K-bit (512 words × 8 bits) data of the random access port into the lower and upper columns and transfers them to the serial access port.

In this cycle, the serial read/write can be performed in the columns to which data is not transfer.

**(a) Setting of Split Read Data Transfer Cycle**

To set this cycle, input a high level to $\overline{CAS}$, $\overline{WB}/\overline{WE}$ and DSF, and low level to $\overline{DT}/\overline{OE}$ at the falling edge of $\overline{RAS}$.

The memory cells (512 words × 8 bits) of the transfer source of the random access port are selected using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$. And the address data input to A0 to A8 at the falling edge of $\overline{CAS}$ is latched as the TAP register data of serial access port. Refer to **3.4 TAP Register.**

If the address data input to A8 is 0, data in lower columns is trasferred from the random access port to the serial access port. If it is 1, data in upper columns is transferred from the random access port to the serial access port. To perform the data transfer in columns for which no serial read is not being made (columns for which the serial register is inactive), it is necessary to supply an inverted QSF output to A8 (See the table below.).

**Table 3-4. Relationships among Data Register Transfer Destination, A8 Inputs, and QSF Outputs (Split Read Data Transfer Cycle in the MSB Care Mode)**

| A8 data input to the TAP register | QSF output | Data register transfer destination |
|---|---|---|
| A8 = 0 | QSF = 1 | Lower column (addresses 0 to 255) |
| A8 = 1 | QSF = 0 | Upper column (addresses 256 to 511) |

**Remark** 1 = high level; 0 = low level

**(b) Execution of Split Read Data Transfer Cycle**

To execute this cycle, set the split read data transfer cycle and then input the high level to $\overline{RAS}$. Data will be transferred at the rising edge of $\overline{RAS}$. Data in the upper or lower column is transferred from the random access port to the serial access port, depending on the data input to A8. To confirm the transferred column side, check the output state of the QSF pin. Refer to **3.3.3 QSF Pin Output.**

When the serial address pointer comes to the jump source address specified by the STOP register, the serial address pointer jumps to the start column (TAP) of the serial read/write cycle at the inactive column side, and the TAP register will be set the empty state.

**Caution When data is tranferred in a split read data transfer cycle, if the related serial register is already involved in serial read/write operation, the serial address pointer changes during data transfer, and the TAP register becomes empty. The timing for this operation is the same as for the read data transfer cycle (SC inactive).**

### 3.2.3 Split Mask Write Data Transfer Cycle (Versions A and F: For "MSB Don't Care")

This cycle divides the 4K-bit (512 words × 8 bits) data of the serial access port into the lower and upper columns and transfers them to the random access port.

In this cycle, serial read/write can be performed for columns to which data is not transferred.

Because $\overline{WB}/\overline{WE}$ is low level at the falling edge of $\overline{RAS}$, the write-per-bit function always functions in this transfer cycle. Refer to **2.8 Mask Write Cycle.**

**(a) Setting of Split Mask Write Data Transfer Cycle**

To set this data transfer cycle, input a high level to $\overline{CAS}$ and DSF and low level to $\overline{DT}/\overline{OE}$, $\overline{WB}/\overline{WE}$ at the falling edge of $\overline{RAS}$. Because the write-per-bit function functions in this transfer operation, for the new mask data method, the mask data must be supplied to W0 to W7 at the falling edge of $\overline{RAS}$, and for the old mask data method, there is no need to control the mask data.

The memory cells (512 words × 8 bits) of the transfer destination of the random access port are selected using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$. The address data input to A0 to A7 at the falling edge of $\overline{CAS}$ is input as the TAP register data. Refer to **3.4 TAP Register.** There is no need to control address data input to A8 (See the table below.).

**Table 3-5. Relationships among Data Register Transfer Source, A8 Inputs, and QSF Outputs (Split Write Data Transfer Cycle in the MSB Don't Care Mode)**

| A8 data input to the TAP register | QSF output | Data register transfer destination |
|---|---|---|
| A8 = × | QSF = 1 | Lower column (addresses 0 to 255) |
| | QSF = 0 | Upper column (addresses 256 to 511) |

**Remark** 1 = high level; 0 = low level; × = high or low level

**(b) Execution of Split Mask Write Data Transfer Cycle**

To execute this cycle, set the split write data transfer cycle and then input high level to $\overline{RAS}$. Data will be transferred at the rising edge of $\overline{RAS}$. Data is transferred from the serial access port to the random access port automatically at the column side (Column not pointed to by the serial address pointer) where the serial access port is inactive. To confirm the transferred column side, check the output state of the QSF pin. Refer to **3.3.3 QSF Pin Output.**

When the serial address pointer comes to the jump source address specified by the STOP register, the serial address pointer jumps to the start column (TAP) of the serial read/write cycle at the inactive column side, and the TAP register will be set the empty state.

### 3.2.4 Split Mask Write Data Transfer Cycle (Versions E: For "MSB Care")

This cycle divides the 4K-bit (512 words × 8 bits) data of the serial access port into the lower and upper columns and transfers them to the random access port.

In this cycle, serial read/write can be performed for columns to which data is not transferred.

Because $\overline{WB}/\overline{WE}$ is low level at the falling edge of $\overline{RAS}$, the write-per-bit function always functions in this transfer cycle. Refer to **2.8 Mask Write Cycle.**

**(a) Setting of Split Mask Write Data Transfer Cycle**

To set this data transfer cycle, input a high level to $\overline{CAS}$ and DSF and low level to $\overline{DT}/\overline{OE}$, $\overline{WB}/\overline{WE}$ at the falling edge of $\overline{RAS}$. Because the write-per-bit function functions in this transfer operation, for the new mask data method, the mask data must be supplied to W0 to W7 at the falling edge of $\overline{RAS}$, and for the old mask data method, there is no need to control the mask data.

The memory cells (512 words × 8 bits) of the transfer destination of the random access port are selected using the row address input to A0 to A8 at the falling edge of $\overline{RAS}$. The address data input to A0 to A8 at the falling edge of $\overline{CAS}$ is input as the TAP register data. Refer to **3.4 TAP Register.**

If the address data input to A8 is 0, data in lower columns is transferred from the serial access port to the random access port. If it is 1, data in upper columns is transferred from the serial access port to the random access port. To perform the data transfer in columns for which no serial write is not being made (columns for which the serial register is inactive), it is necessary to supply an inverted QSF output to A8 (See the table below.).

**Table 3-6. Relationships among Data Register Transfer Source, A8 Inputs, and QSF Outputs (Split Write Data Transfer Cycle in the MSB Care Mode)**

| A8 data input to the TAP register | QSF output | Data register transfer destination |
|---|---|---|
| A8 = 0 | QSF = 1 | Lower column (addresses 0 to 255) |
| A8 = 1 | QSF = 0 | Upper column (addresses 256 to 511) |

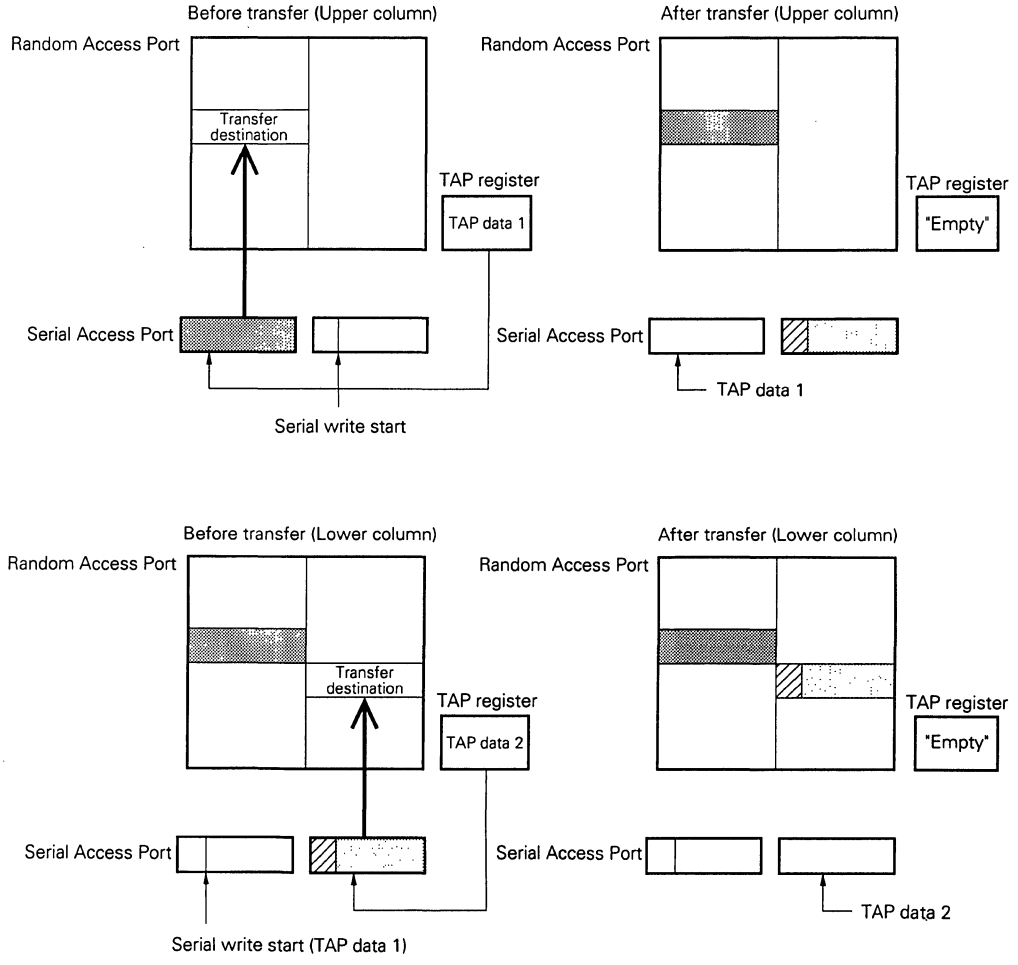
**Remark** 1 = high level; 0 = low level

**(b) Execution of Split Mask Write Data Transfer Cycle**

To execute this cycle, set the split write data transfer cycle and then input high level to $\overline{RAS}$. Data will be transferred at the rising edge of $\overline{RAS}$. Data in the upper or lower column is transferred from the serial access port to the random access port, depending on the data input to A8. To confirm the transferred column side, check the output state of the QSF pin. Refer to **3.3.3 QSF Pin Output.**

When the serial address pointer comes to the jump source address specified by the STOP register, the serial address pointer jumps to the start column (TAP) of the serial read/write cycle at the inactive column side, and the TAP register will be set the empty state.

**Figure 3-2. Split Mask Write Data Transfer and TAP Operations**



### 3.3 Serial Read/Write

The serial access port (512K × 8 bits) is independent from the random access port and can perform read and write operations. The serial access port performing single data transfer and split data transfer can not perform read and write operations independently.

**Caution When the power is turned on, the serial access port sets into the input (write) mode and the SIO pin is the high impedance state.**

#### 3.3.1 Serial Read Cycle

To set the serial read cycle, perform the single read data transfer cycle (The mode will not change in the split read data transfer cycle.).

Execute the single read data transfer cycle and latch the data and TAP data. By inputting a clock signal to the SC pin and inputting a low level to the $\overline{SE}$ pin, data will be output from the serial address pointer specified by TAP register. The data synchronizes with the rising edge of the SC clock and is output from the SIO0 to SIO7 pin, and the data is kept until the next rising edge of the SC clock.

**(a) Reading-Jump**

The $\overline{SE}$ pin controls the SIO pin output buffer independently from the SC clock. By setting the $\overline{SE}$ pin to high level even while inputting the SC clock, SIO0 to SIO7 pins become high impedance. But the operations of serial address pointer will be continued while the SC clock is being input even though reading has been prohibited from $\overline{SE}$ pin. Reading-jump of the column can be performed using this function.

#### 3.3.2 Serial Write Cycle

To set the serial write cycle, perform the single write data transfer cycle (The mode will not change in the split write data transfer cycle.). To prevent the transfer data from being written in the memory cell of the random access port, set all bits of the mask data to "0" and control the mask data.

Execute the single write data transfer cycle and set the serial write cycle. By inputting the clock signal to the SC pin and inputting a low level to the $\overline{SE}$ pin, data can be latched from the serial address pointer specified by TAP register. The data synchronizes with the rising edge of the SC clock and is input from SIO0 to SIO7 pins. Be sure to follow the specifications for the setup time ($t_{SES}$) and hold time ($t_{SEH}$) of $\overline{SE}$ pin for the SC clock.

**(a) Writing-Jumps (Intermittent Writing)**

The $\overline{SE}$ pin controls writing operations independently from the SC clock. By setting the $\overline{SE}$ pin to high level even while inputting the SC clock, writing will not be executed. But the operations of serial address pointer will be continued while the SC clock is being input even though writing has been prohibited from $\overline{SE}$ pin. These functions enable writing-jumps (intermittent writing) to be performed. The masked data is kept as the old data.

#### 3.3.3 QSF Pin Output

QSF pin determines whether the serial address pointer is at the upper column side (addresses 256 to 511) or the lower column side (addresses 0 to 255) at the rising edge of the following SC clock during serial read or write. In other words, it outputs the uppermost bit (A8) of the column address of the serial address pointer.

The following table shows the QSF pin output state and the access pointer of following SC clocks.

| Access Address of Following SC clock | QSF Output |
|---|---|
| Addresses 0 to 255 | Low level |
| Addresses 256 to 511 | High level |

### 3.4 TAP (Top Access Point) Register

The TAP register is a data register which specifies the start address (first serial address point = TAP) of the serial read or serial write.

Set data to this register each time a transfer cycle is executed.

#### 3.4.1 Setting of TAP Register

The data input to A0 to A8 at the falling edge of $\overline{CAS}$ during the setting of a transfer cycle is set as the TAP register data. By executing the transfer cycle, the start address of the following serial read (or write) operations is specified by the data of the TAP register and the TAP register will be kept in the empty state until the TAP register is set again.

In the split data transfer cycle (Versions A and F: For "MSB Don't Care"), because the inactive serial access port column addresses are specified by the data of the TAP register automatically, there is no need to control the A8 data. However in the split data transfer cycle (Version E: For "MSB Care"), the data in the TAP register, which is input to A8, specifies the column on the side that performs the transfer (A8 = 0: Lower column, A8 = 1: Upper column).

**Caution When the TAP register is empty, the address following the 511 serial address point will be 0. In addition, because the serial address pointer will not jump to the column specified by the STOP register, the binary boundary jump function cannot be used. Refer to 3.6 Binary Boundary Jump Function.**

### 3.5 STOP Register

The STOP register is a data register which determines the column of the jump source when jumping to a different column side (lower column or upper column) in the split data transfer cycle. Five types of columns can be selected for starting jump (jumping is possible at 2, 4, 8, 16, and 32 points). The following table shows the correspondence between the column at the jump source and data of the STOP register.

Once set, the STOP register data is kept until it is set again.

#### 3.5.1 Setting of STOP Register

To set the STOP register, set $\overline{WB}/\overline{WE}$ to low level at the falling edge of $\overline{RAS}$ in the $\overline{CAS}$ before $\overline{RAS}$ refresh cycle. The data input to A0 to A7 will be input as the STOP register data.

**Table 3-7. STOP Register Data and Jump Source Column**

| STOP Register Data | | | | | Division | Bit Width | Jump Source Bit Column (Decimal Number) |
|---|---|---|---|---|---|---|---|
| A7 | A6 | A5 | A4 | A3 to A0 | | | |
| 1 | 1 | 1 | 1 | 1 | 1/2 | 256 | 255 |
| | | | | | | | 511 |
| 0 | 1 | 1 | 1 | 1 | 1/4 | 128 | 127, 255 |
| | | | | | | | 383, 511 |
| 0 | 0 | 1 | 1 | 1 | 1/8 | 64 | 63, 127, 191, 255 |
| | | | | | | | 319, 383, 447, 511 |
| 0 | 0 | 0 | 1 | 1 | 1/16 | 32 | 31, 63, 95, 127, 159, 191, 223, 255 |
| | | | | | | | 287, 319, 351, 383, 415, 447, 479, 511 |
| 0 | 0 | 0 | 0 | 1 | 1/32 | 16 | 15, 31, 47, 63, 79, 95, 111, 127, 143, 159, 175, 191, 207, 223, 239, 255 |
| | | | | | | | 271, 287, 303, 319, 335, 351, 367, 383, 399, 415, 431, 447, 463, 479, 495, 511 |

**Remark** A8: Don't care.

**Caution When the power is supplied, all STOP register data will be set to all "1".**

### 3.6 Binary Boundary Jump Function

This function causes the serial address pointer jump to the TAP specified by the TAP register when the pointer moves to a column specified by the STOP register (split data transfer).

This function cannot be used when the jump destination address is not set (TAP register is empty).

This function facilitates tile map application which divides the screen into tiles and manages data for each tile.

#### 3.6.1 Usage of Binary Boundary Jump Function

After setting the STOP register, execute the single read (or write) data transfer and initialize the serial access port. The initialization process will switch the serial access port read (or write) operations, set TAP, set the serial access port data, and set the TAP register to empty. By inputting the serial clock in this state, the serial access port will read (or write) operations from TAP in ascending order of address. Because the TAP register is in the empty state, the address at the jump source set by the STOP register will be ignored, and the serial address pointer will move on.

When the column to be jumped approaches, execute split data transfer and set new TAP data in the TAP register. The serial pointer will jump at the desired jump source address. Jump can be controlled freely by repeating these operations.

### 3.7 Special Operations

#### 3.7.1 Serial Address Set Operations

Because the serial address counter is undefined when the power up, the serial access port operations when the SC clock is input are not guaranteed. Execute single read (or write) transfer after turning on the power. The serial access port will be initialized, enabling serial access port operations to be performed.

#### 3.7.2 Lap Around Operations

If all the data of the register is read (write) during data transfer while the serial read (write) cycle is being executed, the serial pointer will repeat 0 to 511.

#### 3.7.3 Cycle After Power On

After supplying power, initialize the internal circuitry by waiting for at least 100 μs after $V_{CC} \geq 4.5$ V, then supplying at least 8 $\overline{RAS}$ clock cycles. The $\overline{RAS}$ clock only requires that $t_{RC}$, $t_{RAS}$, and $t_{RP}$ are satisfied; there is no problem if other signals are in any state. Note however that if the signal supplied to $\overline{RAS}$, $\overline{CAS}$, $\overline{DT}/\overline{OE}$, and $\overline{WB}/\overline{WE}$ is high at power-on, the serial access port and each register have the following values.

- Serial access port ..... Input mode, SIO: High impedance
- Color register ..... Undefined
- Mask register ..... All "1"
- TAP register ..... Undefined
- STOP register ..... All "1"

## 4. Electrical Characteristics

### Absolute Maximum Ratings

| Parameter | Symbol | Rating | Unit |
|---|---|---|---|
| Pin voltage | $V_T$ | −1.0 to +7.0 | V |
| Supply voltage | $V_{CC}$ | −1.0 to +7.0 | V |
| Output current | $I_O$ | 50 | mA |
| Power dissipation | $P_D$ | 1.5 | W |
| Operating ambient temperature | $T_A$ | 0 to 70 | °C |
| Storage temperature | $T_{stg}$ | −55 to +125 | °C |

**Caution Exposing the device to stress above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits in the operational sections of this characteristics. Exposure to Absolute Maximum rating conditions for extended periods may affect device reliability.**

### Recommended Operating Conditions

| Parameter | Symbol | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|
| Supply voltage | $V_{CC}$ | 4.5 | 5.0 | 5.5 | V |
| High level input voltage | $V_{IH}$ | 2.4 | | 5.5 | V |
| Low level input voltage | $V_{IL}$ | −1.0 | | +0.8 | V |
| Operating ambient temperature | $T_A$ | 0 | | 70 | °C |

**DC Characteristics 1 (Recommended operating conditions unless otherwise noted)**

| Parameter | Symbol | Test conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input leakage current | $I_{IL}$ | $V_{IN}$ = 0 V to 5.5 V, Other inputs are 0 V | −10 | | +10 | μA |
| Output leakage current | $I_{OL}$ | W/IO, SIO, QSF are inactive, $V_{OUT}$ = 0 V to 5.5 V | −10 | | +10 | μA |
| Random access port high level output voltage | $V_{OH}$ (R) | $I_{OH}$ (R) = −1.0mA | 2.4 | | | V |
| Random access port low level output voltage | $V_{OL}$ (R) | $I_{OL}$ (R) = 2.1mA | | | 0.4 | V |
| Serial access port high level output voltage | $V_{OH}$ (S) | $I_{OH}$ (S) = −1.0mA | 2.4 | | | V |
| Serial access port low level output voltage | $V_{OL}$ (S) | $I_{OL}$ (S) = 2.1mA | | | 0.4 | V |

**Capacitance ($T_A$ = 25 °C, f = 1MHz)**

| Parameter | Symbol | Test conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_{I1}$ | $\overline{RAS}$, $\overline{CAS}$, $\overline{WB}/\overline{WE}$, $\overline{DT}/\overline{OE}$, DSF, $\overline{SE}$, SC | | | 8 | pF |
| | $C_{I2}$ | A0 to A8 | | | 5 | |
| Input/Output Capacitance | $C_{IO}$ | W/IO (0 to 7), SIO (0 to 7) | | | 7 | pF |
| Output Capacitance | $C_O$ | QSF | | | 7 | pF |

**DC Characteristics 2 (Recommended operating conditions unless otherwise noted)[Note 1]**
**(μPD482234)**

| Random Access Port | Serial Access Port | | Symbol | μPD482234-60 | | μPD482234-70 | | μPD482234-80 | | Unit | Conditions |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Standby | Active | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Random Read/Write Cycle<br>$\overline{RAS}$, $\overline{CAS}$ cycle,<br>$t_{RC} = t_{RC}$ (MIN.), $I_O$ = 0mA | ○ | | $I_{CC1}$ | | 110 | | 130 | | 130 | mA | Note 2 |
| | | ○ | $I_{CC7}$ | | 155 | | 195 | | 190 | | |
| Standby<br>$\overline{RAS} = \overline{CAS} = V_{IH}$,<br>$D_{OUT}$ = high impedance | ○ | | $I_{CC2}$ | | 10 | | 10 | | 10 | mA | |
| | | | | | 1 | | 1 | | 1 | mA | Note 3 |
| | | ○ | $I_{CC8}$ | | 55 | | 70 | | 65 | mA | |
| $\overline{RAS}$ only refresh cycle<br>$\overline{RAS}$ cycle, $\overline{CAS} = V_{IH}$,<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC3}$ | | 110 | | 115 | | 115 | mA | Note 4 |
| | | ○ | $I_{CC9}$ | | 155 | | 180 | | 175 | | |
| Fast page mode cycle<br>$\overline{RAS} = V_{IL}$, $\overline{CAS}$ cycle,<br>$t_{PC} = t_{PC}$ (MIN.) | ○ | | $I_{CC4}$ | | 100 | | 100 | | 90 | mA | Note 5 |
| | | ○ | $I_{CC10}$ | | 145 | | 165 | | 150 | | |
| $\overline{CAS}$ before $\overline{RAS}$ refresh cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC5}$ | | 110 | | 90 | | 90 | mA | |
| | | ○ | $I_{CC11}$ | | 155 | | 155 | | 150 | | |
| Data transfer cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC6}$ | | 120 | | 140 | | 140 | mA | |
| | | ○ | $I_{CC12}$ | | 165 | | 205 | | 200 | | |
| Color/Mask write register set cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC13}$ | | 100 | | 120 | | 120 | mA | |
| | | ○ | $I_{CC14}$ | | 145 | | 185 | | 180 | | |
| Flash write cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC15}$ | | 100 | | 120 | | 120 | mA | |
| | | ○ | $I_{CC16}$ | | 145 | | 185 | | 180 | | |
| Block write cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC17}$ | | 120 | | 130 | | 130 | mA | |
| | | ○ | $I_{CC18}$ | | 165 | | 195 | | 190 | | |
| Fast page mode block write cycle<br>$t_{PC} = t_{PC}$ (MIN.) | ○ | | $I_{CC19}$ | | 100 | | 110 | | 100 | mA | |
| | | ○ | $I_{CC20}$ | | 130 | | 175 | | 160 | | Note 5 |

**Notes** 1. No load on W/IO, SIO, QSF. The current consumption actually used depends on the output load and operating frequency of each pin.

2. A change in row addresses must not occur more than once in $t_{RC} = t_{RC}$ (MIN.).

3. $\overline{RAS}$, $\overline{CAS}$, and $\overline{SE}$ remain at $V_{IH} \geq V_{CC} - 0.2$ V, and A0 to A8, $\overline{WB}/\overline{WE}$, $\overline{DT}/\overline{OE}$, DSF, SC remain at $V_{IH} \geq V_{CC} - 0.2$ V or $V_{IL} \leq$ GND + 0.2 V.

4. When the address input is set to $V_{IH}$ or $V_{IL}$ during the $t_{RAS}$ period.

5. Value when the address in $t_{PC}$ one cycle is changed once when $t_{PC} = t_{PC}$ (MIN.).

**DC Characteristics 2 (Recommended operating conditions unless otherwise noted)[Note 1]**
**(μPD482235)**

| Random Access Port | Serial Access Port | | Symbol | μPD482235-60 | | μPD482235-70 | | μPD482235-80 | | Unit | Conditions |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Standby | Active | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Random Read/Write Cycle<br>$\overline{RAS}$, $\overline{CAS}$ cycle,<br>$t_{RC} = t_{RC}$ (MIN.), $I_O$ = 0mA | ○ | | $I_{CC1}$ | | 110 | | 130 | | 130 | mA | Note 2 |
| | | ○ | $I_{CC7}$ | | 155 | | 195 | | 190 | | |
| Standby<br>$\overline{RAS} = \overline{CAS} = V_{IH}$,<br>$D_{OUT}$ = high impedance | ○ | | $I_{CC2}$ | | 10 | | 10 | | 10 | mA | |
| | | | | | 1 | | 1 | | 1 | mA | Note 3 |
| | | ○ | $I_{CC8}$ | | 55 | | 70 | | 65 | mA | |
| $\overline{RAS}$ only refresh cycle<br>$\overline{RAS}$ cycle, $\overline{CAS} = V_{IH}$,<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC3}$ | | 110 | | 115 | | 115 | mA | Note 4 |
| | | ○ | $I_{CC9}$ | | 155 | | 180 | | 175 | | |
| Hyper page mode cycle<br>$\overline{RAS} = V_{IL}$, $\overline{CAS}$ cycle,<br>$t_{HPC} = t_{HPC}$ (MIN.) | ○ | | $I_{CC4}$ | | 120 | | 130 | | 120 | mA | Note 5 |
| | | ○ | $I_{CC10}$ | | 155 | | 195 | | 180 | | |
| $\overline{CAS}$ before $\overline{RAS}$ refresh cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC5}$ | | 110 | | 90 | | 90 | mA | |
| | | ○ | $I_{CC11}$ | | 155 | | 155 | | 150 | | |
| Data transfer cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC6}$ | | 120 | | 140 | | 140 | mA | |
| | | ○ | $I_{CC12}$ | | 165 | | 205 | | 200 | | |
| Color/Mask write register set cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC13}$ | | 100 | | 120 | | 120 | mA | |
| | | ○ | $I_{CC14}$ | | 145 | | 185 | | 180 | | |
| Flash write cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC15}$ | | 100 | | 120 | | 120 | mA | |
| | | ○ | $I_{CC16}$ | | 145 | | 185 | | 180 | | |
| Block write cycle<br>$t_{RC} = t_{RC}$ (MIN.) | ○ | | $I_{CC17}$ | | 120 | | 130 | | 130 | mA | |
| | | ○ | $I_{CC18}$ | | 165 | | 195 | | 190 | | |
| Hyper page mode block write cycle<br>$t_{HPC} = t_{HPC}$ (MIN.) | ○ | | $I_{CC19}$ | | 140 | | 135 | | 125 | mA | |
| | | ○ | $I_{CC20}$ | | 190 | | 200 | | 185 | | Note 5 |

**Notes** 1. No load on W/IO, SIO, QSF. The current consumption actually used depends on the output load and operating frequency of each pin.

2. A change in row addresses must not occur more than once in $t_{RC} = t_{RC}$ (MIN.).

3. $\overline{RAS}$, $\overline{CAS}$, and $\overline{SE}$ remain at $V_{IH} \geq V_{CC} - 0.2$ V, and A0 to A8, $\overline{WB}/\overline{WE}$, $\overline{DT}/\overline{OE}$, DSF, SC remain at $V_{IH} \geq V_{CC} - 0.2$ V or $V_{IL} \leq$ GND + 0.2 V.

4. When the address input is set to $V_{IH}$ or $V_{IL}$ during the $t_{RAS}$ period.

5. Value when the address in $t_{HPC}$ one cycle is changed once when $t_{HPC} = t_{HPC}$ (MIN.).

**AC Characteristics ($T_A$ = 0 to 70 °C, $V_{CC}$ = 5.0 V ± 10 %, GND = 0 V)[Notes 1, 2, 3, 4]**

**(Common for μPD482234, 482235)** (1/4)

| Parameter | Symbol | μPD482234-60 μPD482235-60 | | μPD482234-70 μPD482235-70 | | μPD482234-80 μPD482235-80 | | Unit | Condition |
|---|---|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Access time from $\overline{RAS}$ | $t_{RAC}$ | | 60 | | 70 | | 80 | ns | Note 5 |
| Access time from $\overline{CAS}$ | $t_{CAC}$ | | 15 | | 20 | | 25 | ns | Note 5 |
| Access time from column address | $t_{AA}$ | | 30 | | 35 | | 40 | ns | Note 5 |
| Access time from $\overline{CAS}$ trailing edge | $t_{ACP}$ | | 35 | | 40 | | 45 | ns | |
| Access time from $\overline{OE}$ | $t_{OEA}$ | | 15 | | 20 | | 25 | ns | |
| Serial output access time from SC | $t_{SCA}$ | | 15 | | 17 | | 20 | ns | |
| Serial output access time from $\overline{SE}$ | $t_{SEA}$ | | 15 | | 17 | | 20 | ns | |
| Output disable time from $\overline{SE}$ high | $t_{SEZ}$ | 0 | 15 | 0 | 15 | 0 | 20 | ns | Note 6 |
| Random read or write cycle time | $t_{RC}$ | 120 | | 140 | | 150 | | ns | |
| Read modify-write cycle time | $t_{RWC}$ | 165 | | 185 | | 205 | | ns | |
| Transition time (Rise/Fall) | $t_T$ | 3 | 35 | 3 | 35 | 3 | 35 | ns | |
| $\overline{RAS}$ precharge time | $t_{RP}$ | 50 | | 60 | | 60 | | ns | |
| $\overline{RAS}$ pulse width (Non page mode) | $t_{RAS}$ | 60 | 10,000 | 70 | 10,000 | 80 | 10,000 | ns | |
| $\overline{RAS}$ pulse width (Fast page/Hyper page mode) | $t_{RASP}$ | 60 | 100,000 | 70 | 100,000 | 80 | 100,000 | ns | |
| $\overline{RAS}$ hold time | $t_{RSH}$ | 15 | | 20 | | 25 | | ns | |
| $\overline{CAS}$ precharge time (Non page mode) | $t_{CPN}$ | 10 | | 10 | | 10 | | ns | |
| $\overline{CAS}$ precharge time (Fast page/Hyper page mode) | $t_{CP}$ | 10 | | 10 | | 10 | | ns | |
| $\overline{CAS}$ hold time | $t_{CSH}$ | 60 | | 70 | | 80 | | ns | |
| $\overline{RAS}$ to $\overline{CAS}$ delay time | $t_{RCD}$ | 20 | 40 | 20 | 50 | 22 | 55 | ns | Note 5 |
| $\overline{CAS}$ high to $\overline{RAS}$ low precharge time | $t_{CRP}$ | 5 | | 10 | | 10 | | ns | |
| $\overline{RAS}$ high to $\overline{CAS}$ low precharge time | $t_{RPC}$ | 10 | | 10 | | 10 | | ns | |
| Row address setup time | $t_{ASR}$ | 0 | | 0 | | 0 | | ns | |
| Row address hold time | $t_{RAH}$ | 10 | | 10 | | 12 | | ns | |
| Column address setup time | $t_{ASC}$ | 0 | | 0 | | 0 | | ns | |
| Column address hold time | $t_{CAH}$ | 10 | | 10 | | 12 | | ns | |
| $\overline{RAS}$ to column address delay time | $t_{RAD}$ | 15 | 30 | 15 | 35 | 17 | 40 | ns | Note 5 |
| Column address to $\overline{RAS}$ lead time | $t_{RAL}$ | 30 | | 35 | | 40 | | ns | |
| Read command setup time | $t_{RCS}$ | 0 | | 0 | | 0 | | ns | |

**(Common for μPD482234, 482235)** (2/4)

| Parameter | Symbol | μPD482234-60 μPD482235-60 | | μPD482234-70 μPD482235-70 | | μPD482234-80 μPD482235-80 | | Unit | Condition |
|---|---|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Read command hold time after $\overline{RAS}$ high | $t_{RRH}$ | 0 | | 0 | | 0 | | ns | **Note 7** |
| Read command hold time after $\overline{CAS}$ high | $t_{RCH}$ | 0 | | 0 | | 0 | | ns | **Note 7** |
| $\overline{OE}$ hold time after $\overline{RAS}$ high | $t_{ORH}$ | 10 | | 10 | | 10 | | ns | **Note 8** |
| $\overline{OE}$ hold time after $\overline{CAS}$ high | $t_{OCH}$ | 10 | | 10 | | 10 | | ns | **Note 8** |
| Write command setup time | $t_{WCS}$ | 0 | | 0 | | 0 | | ns | **Note 10** |
| Write command hold time | $t_{WCH}$ | 10 | | 12 | | 15 | | ns | |
| Write command pulse width | $t_{WP}$ | 10 | | 12 | | 15 | | ns | |
| Write command to $\overline{RAS}$ lead time | $t_{RWL}$ | 20 | | 20 | | 20 | | ns | |
| Write command to $\overline{CAS}$ lead time | $t_{CWL}$ | 15 | | 15 | | 20 | | ns | |
| Data in setup time | $t_{DS}$ | 0 | | 0 | | 0 | | ns | **Note 9** |
| Data in hold time | $t_{DH}$ | 12 | | 12 | | 15 | | ns | **Note 9** |
| Refresh period | $t_{REF}$ | | 8 | | 8 | | 8 | ms | |
| $\overline{CAS}$ to $\overline{WB}/\overline{WE}$ delay time | $t_{CWD}$ | 40 | | 40 | | 50 | | ns | **Note 10** |
| $\overline{RAS}$ to $\overline{WB}/\overline{WE}$ delay time | $t_{RWD}$ | 85 | | 90 | | 105 | | ns | **Note 10** |
| Column address to $\overline{WB}/\overline{WE}$ delay time | $t_{AWD}$ | 55 | | 55 | | 65 | | ns | **Note 10** |
| $\overline{CAS}$ setup time (for $\overline{CAS}$ before $\overline{RAS}$ refresh cycle) | $t_{CSR}$ | 0 | | 0 | | 0 | | ns | |
| $\overline{CAS}$ hold time (for $\overline{CAS}$ before $\overline{RAS}$ refresh cycle) | $t_{CHR}$ | 10 | | 10 | | 12 | | ns | |
| $\overline{DT}$ low setup time | $t_{DLS}$ | 0 | | 0 | | 0 | | ns | |
| $\overline{DT}$ low hold time after $\overline{RAS}$ low | $t_{RDH}$ | 55 | | 65 | | 70 | | ns | **Note 11** |
| $\overline{DT}$ low hold time after $\overline{RAS}$ low | $t_{RDHS}$ | 10 | | 25 | | 30 | | ns | **Note 11** |
| $\overline{DT}$ low hold time after $\overline{CAS}$ low | $t_{CDH}$ | 15 | | 20 | | 25 | | ns | **Note 11** |
| $\overline{DT}$ low hold time after address | $t_{ADD}$ | 20 | | 25 | | 30 | | ns | **Note 11** |
| SC high to $\overline{DT}$ high | $t_{SDD}$ | 0 | | 0 | | 0 | | ns | **Note 11** |
| SC high to $\overline{CAS}$ low | $t_{SSC}$ | 10 | | 10 | | 10 | | ns | **Note 11, 15, 16** |
| SC low hold time after $\overline{DT}$ high | $t_{SDH}$ | 40 | | 40 | | 50 | | ns | **Note 11** |

**(Common for μPD482234, 482235)** (3/4)

| Parameter | Symbol | μPD482234-60 μPD482235-60 | | μPD482234-70 μPD482235-70 | | μPD482234-80 μPD482235-80 | | Unit | Condition |
|---|---|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| SC low hold time after $\overline{DT}$ high | $t_{SDHR}$ | 40 | | 45 | | 55 | | ns | **Note 11, 15** |
| $\overline{OE}$ high to data in setup delay time | $t_{OED}$ | 15 | | 15 | | 20 | | ns | |
| $\overline{OE}$ high hold time after $\overline{WB/WE}$ low | $t_{OEH}$ | 0 | | 0 | | 0 | | ns | |
| Serial clock cycle time | $t_{SCC}$ | 18 | | 22 | | 25 | | ns | |
| SC pulse width | $t_{SCH}$ | 5 | | 5 | | 7 | | ns | |
| SC precharge time | $t_{SCL}$ | 5 | | 5 | | 7 | | ns | |
| $\overline{SE}$ low to serial output setup delay time | $t_{SOO}$ | 3 | | 5 | | 5 | | ns | |
| Serial output hold time after SC high | $t_{SOH}$ | 3 | | 5 | | 5 | | ns | |
| $\overline{DT}$ high setup time | $t_{DHS}$ | 0 | | 0 | | 0 | | ns | |
| $\overline{DT}$ high hold time | $t_{DHH}$ | 10 | | 10 | | 12 | | ns | |
| $\overline{DT}$ high to $\overline{RAS}$ high delay time | $t_{DTR}$ | 0 | | 0 | | 0 | | ns | **Note 11** |
| $\overline{DT}$ high pulse width | $t_{DTP}$ | 20 | | 20 | | 25 | | ns | |
| $\overline{OE}$ to $\overline{RAS}$ inactive setup time | $t_{OES}$ | 0 | | 0 | | 0 | | ns | |
| Write-per-bit setup time | $t_{WBS}$ | 0 | | 0 | | 0 | | ns | |
| Write-per-bit hold time | $t_{WBH}$ | 10 | | 10 | | 12 | | ns | |
| DSF setup time from $\overline{RAS}$ | $t_{FRS}$ | 0 | | 0 | | 0 | | ns | |
| DSF hold time from $\overline{RAS}$ | $t_{FRH}$ | 10 | | 10 | | 12 | | ns | |
| DSF setup time from $\overline{CAS}$ | $t_{FCS}$ | 0 | | 0 | | 0 | | ns | |
| DSF hold time from $\overline{CAS}$ | $t_{FCH}$ | 10 | | 12 | | 15 | | ns | |
| Write-per-bit selection setup time | $t_{WS}$ | 0 | | 0 | | 0 | | ns | |
| Write-per-bit selection hold time | $t_{WH}$ | 10 | | 10 | | 12 | | ns | |
| $\overline{SE}$ pulse width | $t_{SEE}$ | 5 | | 5 | | 7 | | ns | |
| $\overline{SE}$ precharge time | $t_{SEP}$ | 5 | | 5 | | 7 | | ns | |
| $\overline{SE}$ setup time | $t_{SES}$ | 0 | | 0 | | 0 | | ns | |
| $\overline{SE}$ hold time from SC | $t_{SEH}$ | 10 | | 10 | | 12 | | ns | |
| Serial data in setup time | $t_{SIS}$ | 0 | | 0 | | 0 | | ns | |
| Serial data in hold time | $t_{SIH}$ | 10 | | 10 | | 12 | | ns | |
| Serial input disable time from SC | $t_{SIZ}$ | 0 | | 0 | | 0 | | ns | |
| Serial output disable time from $\overline{RAS}$ | $t_{SRZ}$ | 0 | | 0 | | 0 | | ns | |

**(Common for μPD482234, 482235)** (4/4)

| Parameter | Symbol | μPD482234-60 μPD482235-60 | | μPD482234-70 μPD482235-70 | | μPD482234-80 μPD482235-80 | | Unit | Condition |
|---|---|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Serial input enable time from $\overline{RAS}$ | $t_{SZH}$ | 20 | | 20 | | 25 | | ns | |
| SC setup time from $\overline{RAS}$ | $t_{SRS}$ | 10 | | 10 | | 10 | | ns | **Note 14, 15, 16** |
| SC hold time from $\overline{RAS}$ | $t_{SRH}$ | 10 | | 10 | | 10 | | ns | **Note 14** |
| Propagation delay time from SC to QSF | $t_{PD}$ | | 20 | | 20 | | 25 | ns | |
| Propagation delay time from $\overline{RAS}$ to QSF | $t_{RQD}$ | | 80 | | 95 | | 105 | ns | |
| Propagation delay time from $\overline{CAS}$ to QSF | $t_{CQD}$ | | 60 | | 65 | | 75 | ns | |
| Propagation delay time from $\overline{DT}/\overline{OE}$ to QSF | $t_{DQD}$ | | 30 | | 30 | | 35 | ns | |
| Propagation delay time from $\overline{RAS}$ high to QSF | $t_{DQR}$ | | 40 | | 40 | | 45 | ns | |

(μPD482234 Only)

| Parameter | Symbol | μPD482234-60 | | μPD482234-70 | | μPD482234-80 | | Unit | Condition |
|---|---|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Output disable time from $\overline{CAS}$ high | $t_{OFF}$ | 0 | 15 | 0 | 15 | 0 | 20 | ns | **Note 6, 12** |
| Output disable time from $\overline{OE}$ high | $t_{OEZ}$ | 0 | 15 | 0 | 15 | 0 | 20 | ns | **Note 6, 12** |
| Output disable time from $\overline{WB}/\overline{WE}$ low | $t_{WEZ}$ | 0 | 15 | 0 | 15 | 0 | 20 | ns | **Note 6, 12** |
| Write command pulse width | $t_{WPZ}$ | 10 | | 12 | | 15 | | ns | **Note 12** |
| Fast page mode cycle time | $t_{PC}$ | 40 | | 45 | | 50 | | ns | |
| Fast page mode read modify write cycle time | $t_{PRWC}$ | 90 | | 90 | | 105 | | ns | |
| $\overline{CAS}$ pulse width | $t_{CAS}$ | 15 | 10,000 | 20 | 10,000 | 20 | 10,000 | ns | |

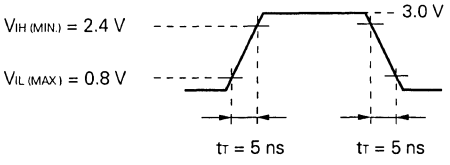
(μPD482235 Only)

| Parameter | Symbol | μPD482235-60 | | μPD482235-70 | | μPD482235-80 | | Unit | Condition |
|---|---|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Access time from previous $\overline{CAS}$ | $t_{ACE}$ | | 60 | | 65 | | 75 | ns | **Note 17** |
| Access time from previous $\overline{WE}$ | $t_{AWE}$ | | 55 | | 60 | | 70 | ns | **Note 17** |
| Output hold time from $\overline{CAS}$ | $t_{DHC}$ | 3 | | 5 | | 5 | | ns | |
| Output disable time from $\overline{RAS}$ high | $t_{OFR}$ | 0 | 15 | 0 | 15 | 0 | 20 | ns | **Note 6, 13** |
| Output disable time from $\overline{CAS}$ high | $t_{OFC}$ | 0 | 15 | 0 | 15 | 0 | 20 | ns | **Note 6, 13** |
| Output disable time from $\overline{OE}$ high | $t_{OFZ}$ | 0 | 15 | 0 | 15 | 0 | 20 | ns | **Note 6, 13** |
| Output disable time from $\overline{WB}/\overline{WE}$ low | $t_{WEZ}$ | 0 | 15 | 0 | 15 | 0 | 20 | ns | **Note 6, 13** |
| Write command pulse width | $t_{WPZ}$ | 10 | | 12 | | 15 | | ns | **Note 13** |
| Hyper page mode cycle time | $t_{HPC}$ | 30 | | 35 | | 40 | | ns | |
| Hyper page mode read modify write cycle time | $t_{HPRWC}$ | 80 | | 90 | | 105 | | ns | |
| $\overline{CAS}$ pulse width | $t_{HCAS}$ | 10 | 10,000 | 10 | 10,000 | 12 | 10,000 | ns | |

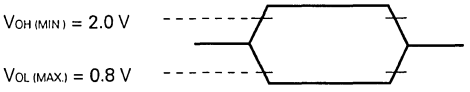
**Notes** 1. All applied voltages are referenced to GND.

2. After supplying power, initialize the internal circuitry by waiting for at least 100 μs after $V_{CC} \geq 4.5$ V, then supplying at least 8 $\overline{RAS}$ clock cycles. The $\overline{RAS}$ clock only requires $t_{RC}$, $t_{RAS}$, and $t_{RP}$ are satisfied; there is no problem if other signals are in any state.
3. Measure at $t_T = 5$ ns
4. AC characteristic measuring conditions

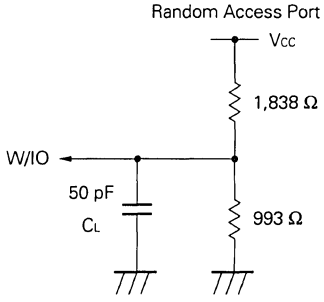
**(1) Input voltage, timing**

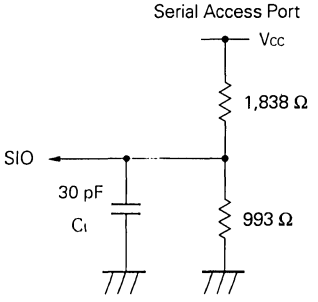



**(2) Output voltage determined**




**(3) Output load conditions**




5. For read cycle, access time is defined as follows:

| Input conditions | Access time | Access time from $\overline{RAS}$ |
|---|---|---|
| $t_{RAD} \leq t_{RAD}$ (MAX.) and $t_{RCD} \leq t_{RCD}$ (MAX.) | $t_{RAC}$ (MAX.) | $t_{RAC}$ (MAX.) |
| $t_{RAD} > t_{RAD}$ (MAX.) and $t_{RCD} \leq t_{RCD}$ (MAX.) | $t_{AA}$ (MAX.) | $t_{RAD} + t_{AA}$ (MAX.) |
| $t_{RCD} > t_{RCD}$ (MAX.) | $t_{CAC}$ (MAX.) | $t_{RCD} + t_{CAC}$ (MAX.) |

$t_{RAD}$ (MAX.) and $t_{RCD}$ (MAX.) are specified as reference points only; they are not restrictive operating parameters. They are used to determine which access time ($t_{RAC}$, $t_{AA}$, $t_{CAC}$) is to be used for finding out data will be available. Therefore, the input conditions $t_{RAD} \geq t_{RAD}$ (MAX.) and $t_{RCD} \geq t_{RCD}$ (MAX.) will not cause any operation problems.

6. $t_{SEZ}$, $t_{OEZ}$, $t_{WEZ}$, $t_{OFF}$, $t_{OFR}$, and $t_{OFC}$ define the time when the output achieves the condition of high impedance and is not referenced to $V_{OH}$ or $V_{OL}$.
7. Either $t_{RCH}$ (MIN.) or $t_{RRH}$ (MIN.) should be met in read cycles.
8. Because $t_{ORH}$ and $t_{OCH}$ are used during the mask register set cycle (new mask selection) only, these ratings are applied only to versions F and E.
9. These parameters are referenced to the following points.
   (1) Early write cycle : The falling edge of $\overline{CAS}$
   (2) Late write cycle : The falling edge of $\overline{WB}/\overline{WE}$
   (3) Read modify write cycle : The falling edge of $\overline{WB}/\overline{WE}$

**Notes 10.** $t_{WCS} \geq t_{WCS}$ (MIN.) is the condition for early write cycle to be set. $D_{OUT}$ becomes high impedance during the cycle.

$t_{RWD} \geq t_{RWD}$ (MIN.), $t_{CWD} \geq t_{CWD}$ (MIN.), $t_{AWD} \geq t_{AWD}$ (MIN.), are conditions for read modify write cycle to be set. The data of the selected address is output to $D_{OUT}$.

If any of the above conditions are not met, pin W/IO will become undefined.

**11.** One of the following specifications will be valid depending on the type of read data transfer method used.

(1) $\overline{DT}/\overline{OE}$ edge control: Satisfy the following specifications.

- For $\overline{DT}/\overline{OE}$ edge inputs : $t_{RDH}$, $t_{CDH}$, $t_{ADD}$, $t_{DTR}$
- For SC inputs : $t_{SDD}$, $t_{SDH}$

(2) Self control: Satisfy the following specification.

- For $\overline{DT}/\overline{OE}$ edge inputs : $t_{RDHS}$
- For SC inputs : $t_{SSC}$, $t_{SDHR}$

**12.** Control pins $\overline{CAS}$, $\overline{DT}/\overline{OE}$, $\overline{WB}/\overline{WE}$ to set pin W/IO to high impedance. Because the timings at which $\overline{CAS}$ and $\overline{DT}/\overline{OE}$ are set to high level and $\overline{WB}/\overline{WE}$ is set to low level affect the high impedance state, the specifications will change as follows.

(1) When $\overline{CAS}$ is set to high level at $\overline{DT}/\overline{OE}$ (low level) and $\overline{WB}/\overline{WE}$ (high level) at the end of the read cycle: $t_{OFF}$ is valid

(2) When $\overline{WB}/\overline{WE}$ is set to low level at $\overline{CAS}$ (low level) and $\overline{DT}/\overline{OE}$ (low level) at the end of the read cycle: $t_{WEZ}$ and $t_{WPZ}$ are valid

(3) When $\overline{DT}/\overline{OE}$ is set to high level at $\overline{CAS}$ (low level) and $\overline{WB}/\overline{WE}$ (high level) at the end of the read cycle: $t_{OEZ}$ is valid

**13.** Control pins $\overline{RAS}$, $\overline{CAS}$, $\overline{DT}/\overline{OE}$, $\overline{WB}/\overline{WE}$ to set pin W/IO to high impedance. Because the timings at which $\overline{RAS}$, $\overline{CAS}$, and $\overline{DT}/\overline{OE}$ are set to high level and $\overline{WB}/\overline{WE}$ is set to low level affect the high impedance state, the specifications will change as follows.

When controlling $\overline{RAS}$ and $\overline{CAS}$, the output cannot be made high impedance unless both pins are set to high. There is difference between $t_{OFC}$ and $t_{OFR}$, because $\overline{RAS}$ and $\overline{CAS}$ control is specified from the rising edge of the slower one.

(1) When $\overline{RAS}$ is set to high level after $\overline{CAS}$ is set to high level at $\overline{DT}/\overline{OE}$ (low level) and $\overline{WB}/\overline{WE}$ (high level) at the end of the read cycle: $t_{OFR}$ is valid

(2) When $\overline{CAS}$ is set to high level after $\overline{RAS}$ is set to high level at $\overline{DT}/\overline{OE}$ (low level) and $\overline{WB}/\overline{WE}$ (high level) at the end of the read cycle: $t_{OFC}$ is valid

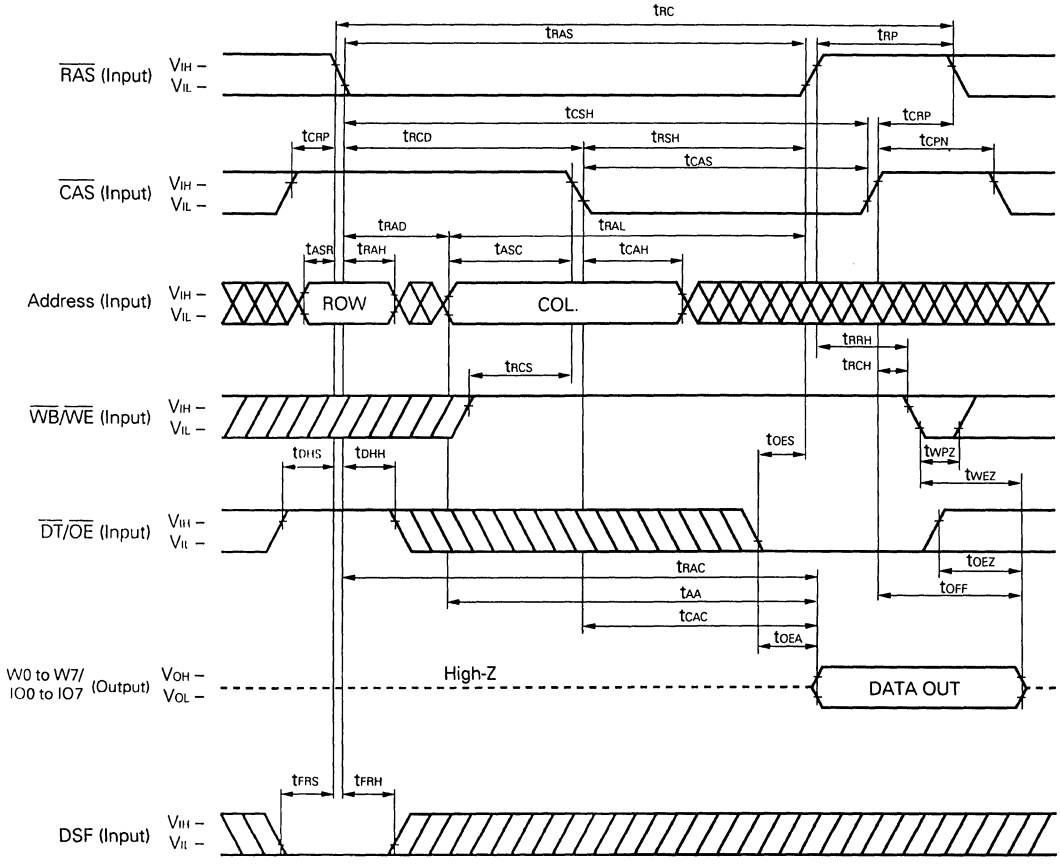
(3) When $\overline{WB}/\overline{WE}$ is set to low level at $\overline{RAS}$, $\overline{CAS}$ (low level) and $\overline{DT}/\overline{OE}$ (low level) at the end of the read cycle: $t_{WEZ}$ and $t_{WPZ}$ are valid

(4) When $\overline{DT}/\overline{OE}$ is set to high level at $\overline{RAS}$, $\overline{CAS}$ (low level) and $\overline{WB}/\overline{WE}$ (high level) at the end of the read cycle: $t_{OEZ}$ is valid

**14.** The $t_{SRS}$ and $t_{SRH}$ in the hidden refresh cycle, $\overline{CAS}$ before $\overline{RAS}$ refresh cycle (STOP register set cycle and optional reset cycle) are specified to guarantee the serial port operations until the transfer cycle is executed after the STOP register value is changed. When the STOP register value is not to be changed, or when the binary boundary jump function is not used (when the TAP register is empty), $t_{SRS}$ and $t_{SRH}$ will not be specified.

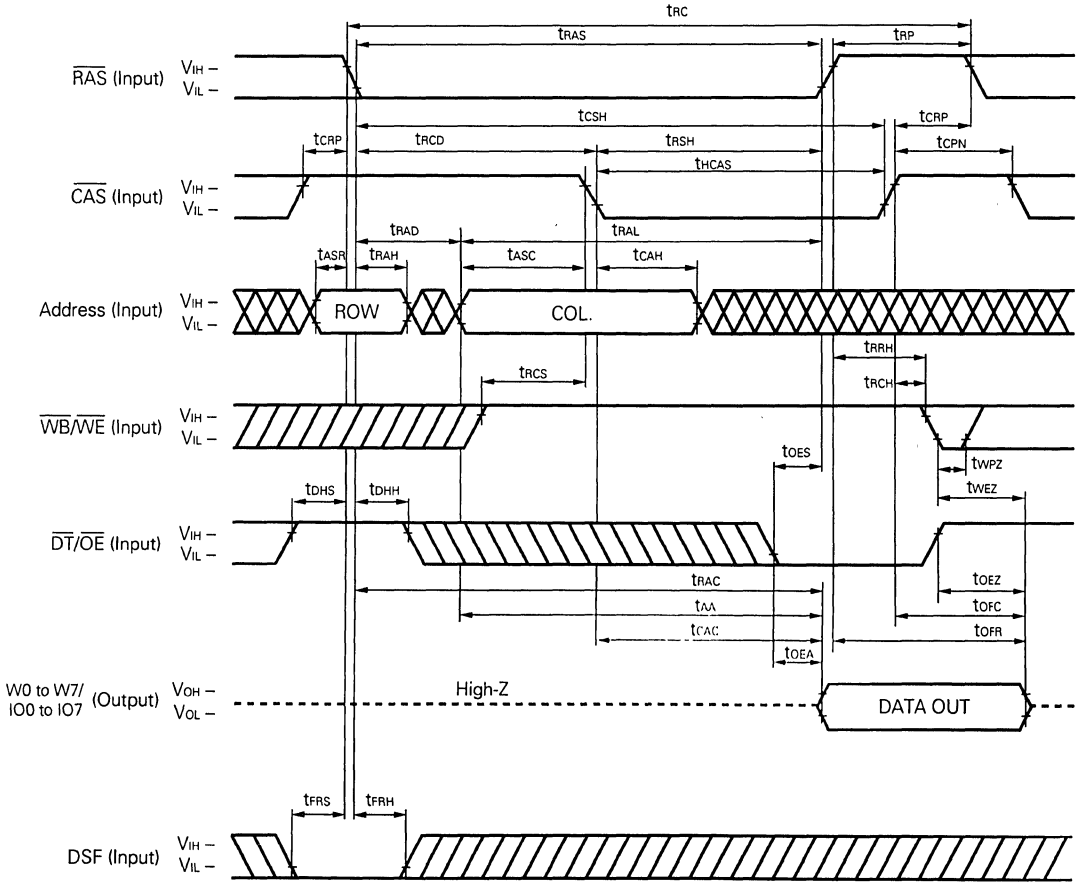
15. $t_{SSC}$ (split read data transfer cycle) and $t_{SRS}$ (split write data transfer cycle) are specified at the rising edge of SC which reads/writes the address of the jump source in the binary boundary jump function. $t_{SDHR}$ (split read data transfer cycle and split write data transfer cycle) is specified at the rising edge of SC which reads/writes the address of the jump destination in the binary boundary jump function. The rising edge of these SCs cannot be input in periods (1) and (2).
    (1) Split read data transfer cycle: Period from the rising edge of the SC specifying $t_{SSC}$ to that of the SC specifying $t_{SDHR}$ (Refer to **Note 2 Split Read/Write Data Transfer Cycle Timing Chart.**)
    (2) Split write data transfer cycle: Period from the rising edge of the SC specifying $t_{SRS}$ to that of the SC specifying $t_{SDHR}$ (Refer to **Note 2 Split Read/Write Data Transfer Cycle Timing Chart.**)
16. Restrictions to the split read data transfer cycle during serial write operation
    (1) If split read data transfer is attempted for an address which is already involved in serial write, normal operation is not guaranteed, except for a period in which no serial write has been performed, that is from when $\overline{SE}$ goes low at the rising edge of SC to just before the serial write begins.
    (2) If split read data transfer is attempted when an address involved in serial write is the boundary address specified by the STOP register, normal operation is not guaranteed, except for a period in which no serial write has been performed, that is from just after the mask write or mask split write transfer cycle is executed to just before the serial write is started by setting $\overline{SE}$ to a low level at the rising edge of SC.
17. In the hyper page mode, the hyper page mode read modify write cycle, the hyper page mode read modify block write cycle, this parameter is valid when the read cycle changes to the write cycle.

**Read Cycle (μPD482234)**



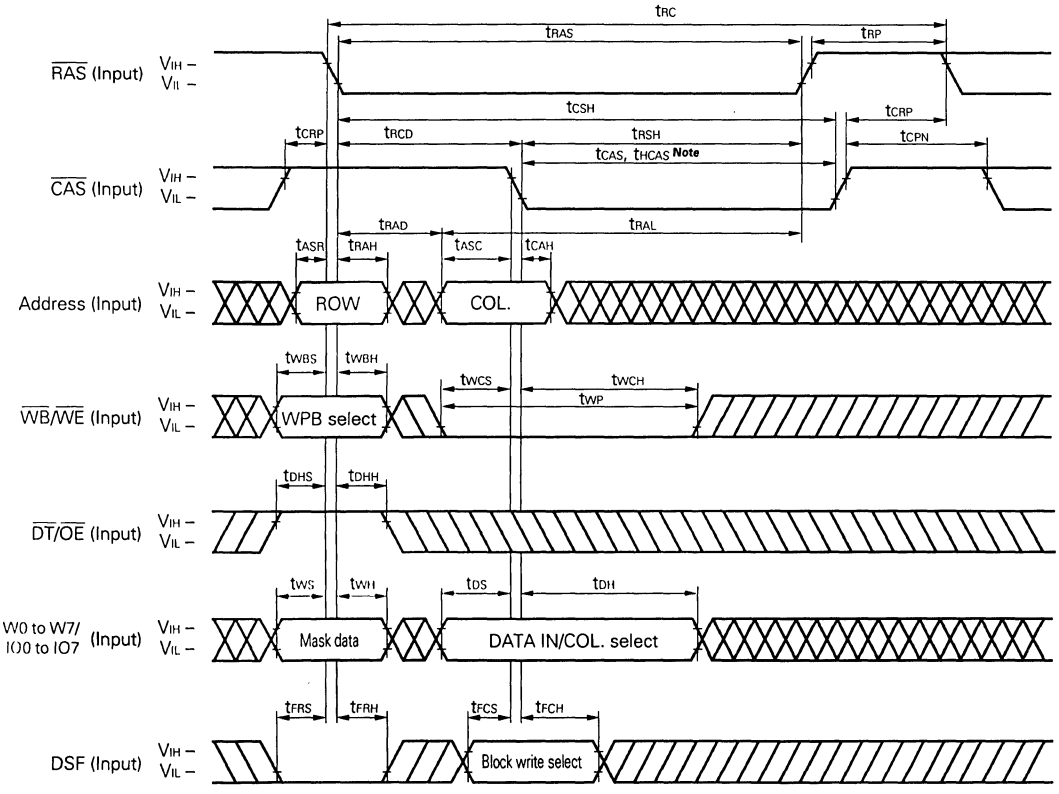

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Read Cycle (Extended data output: μPD482235)**




**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

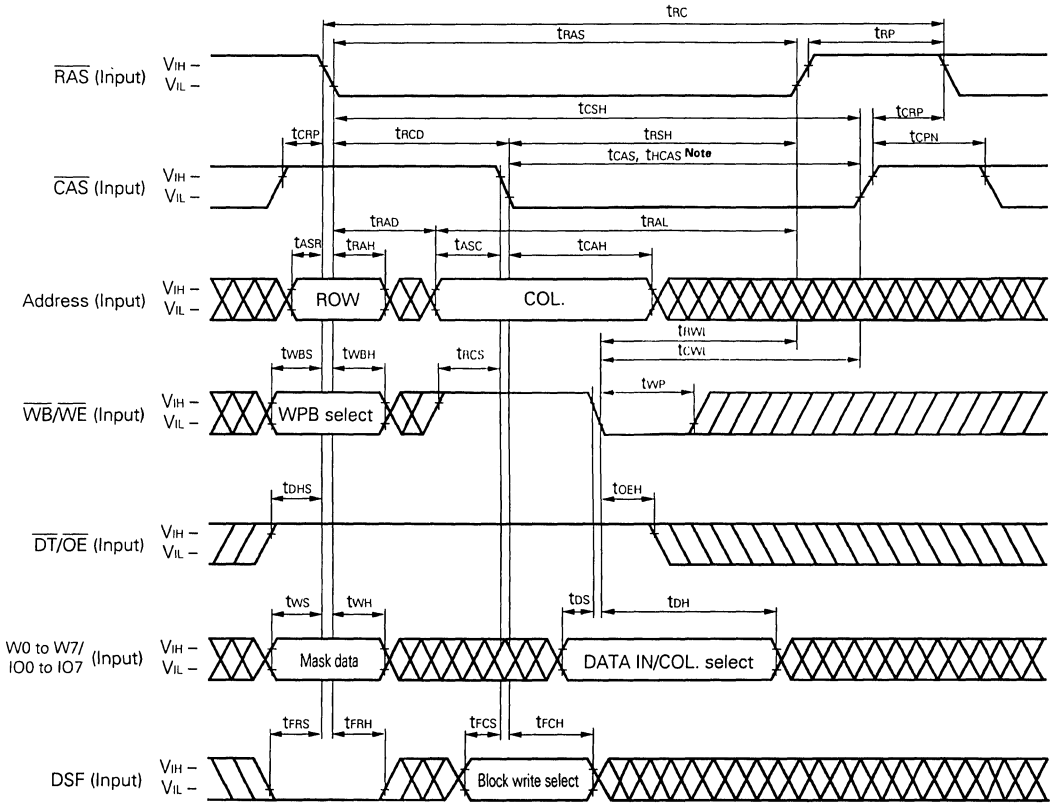
**Early Write Cycle/Early Block Write Cycle**




**Note** tCAS for the μPD482234
tHCAS for the μPD482235

**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

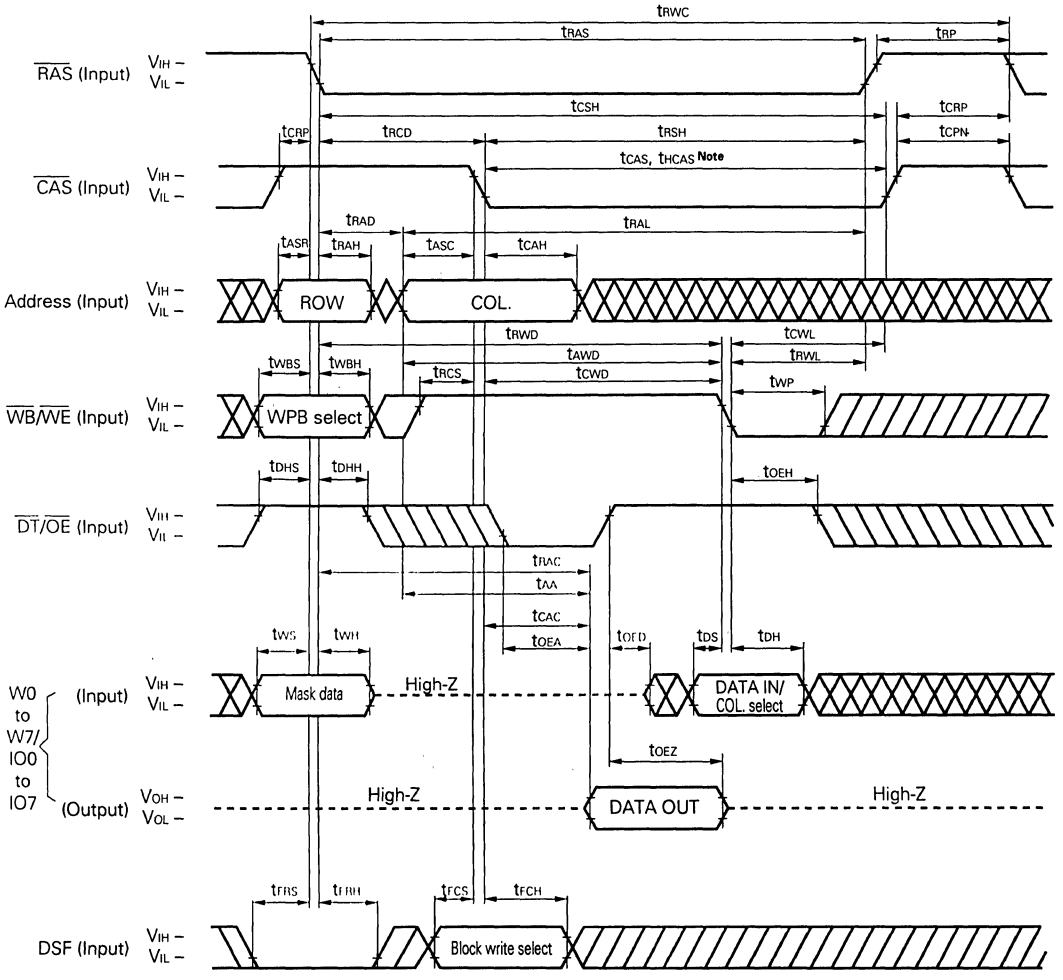
**Late Write Cycle/Late Block Write Cycle**




**Note** tCAS for the μPD482234
tHCAS for the μPD482235

**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

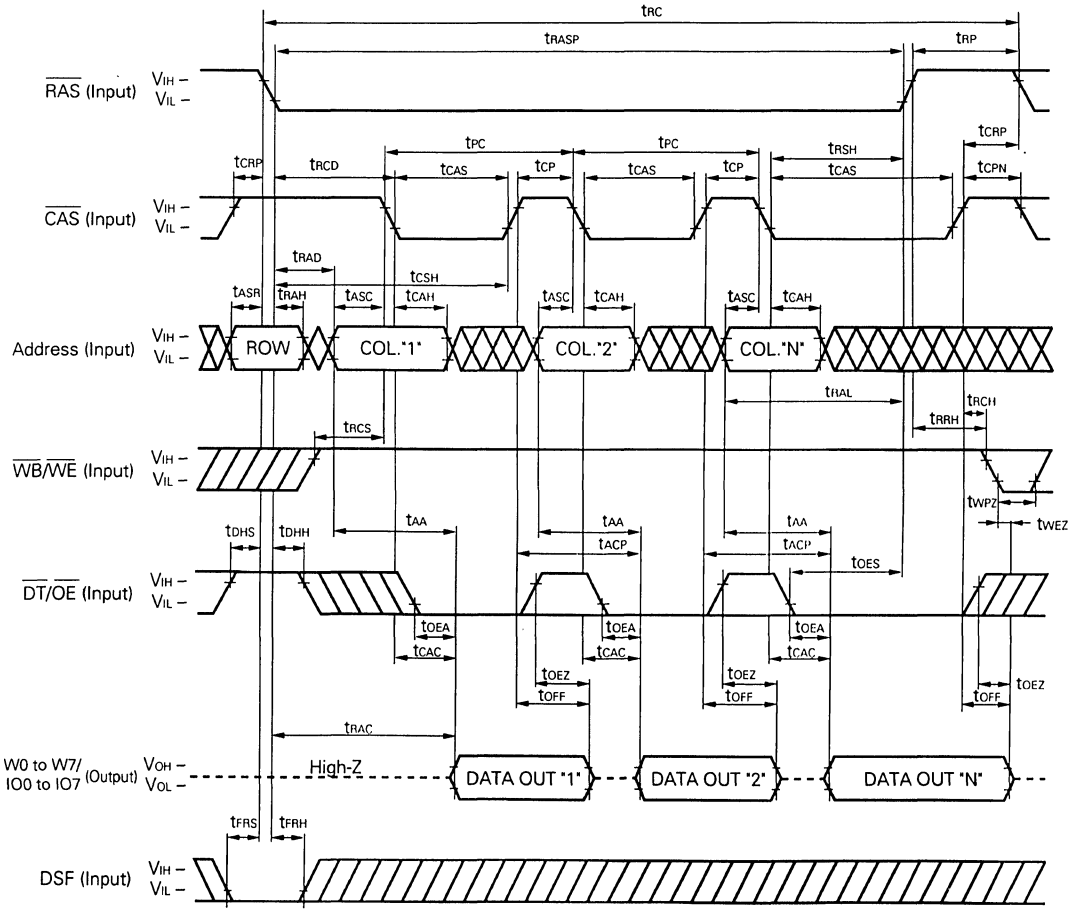
**Read Modify Write Cycle/Read Modify Block Write Cycle**




**Note** $t_{CAS}$ for the μPD482234
$t_{HCAS}$ for the μPD482235

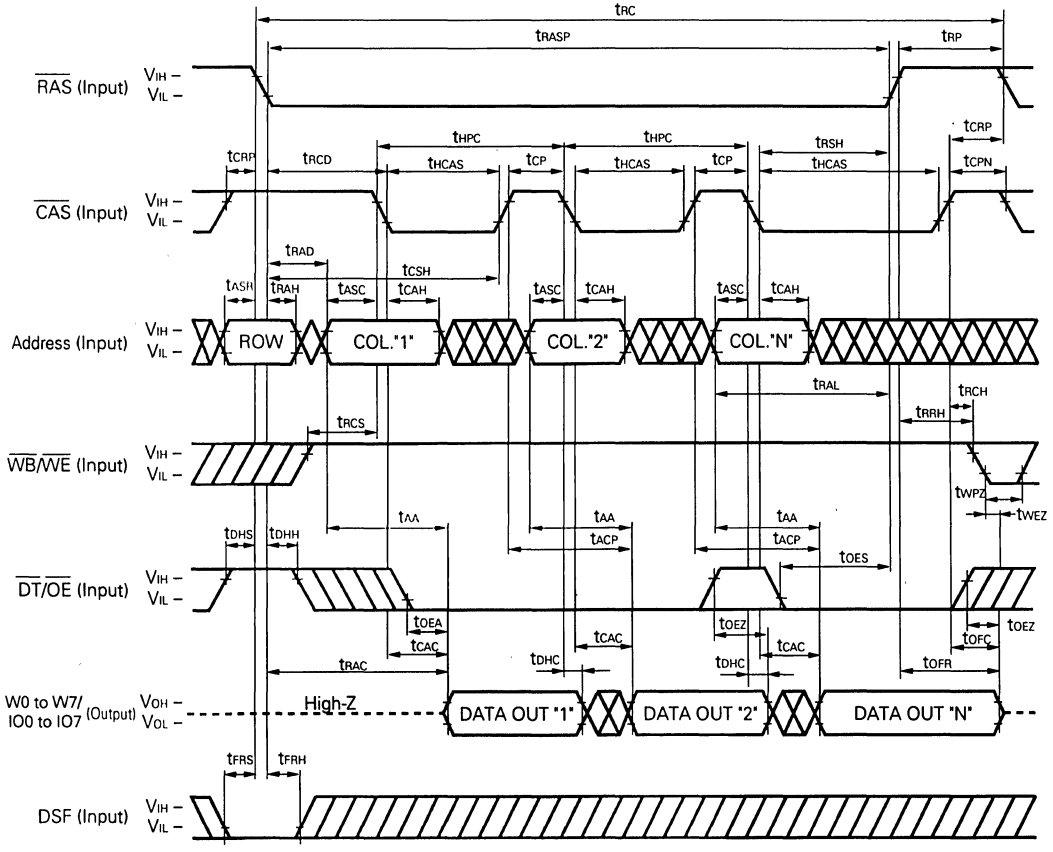
**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Fast Page Mode Read Cycle (μPD482234)**



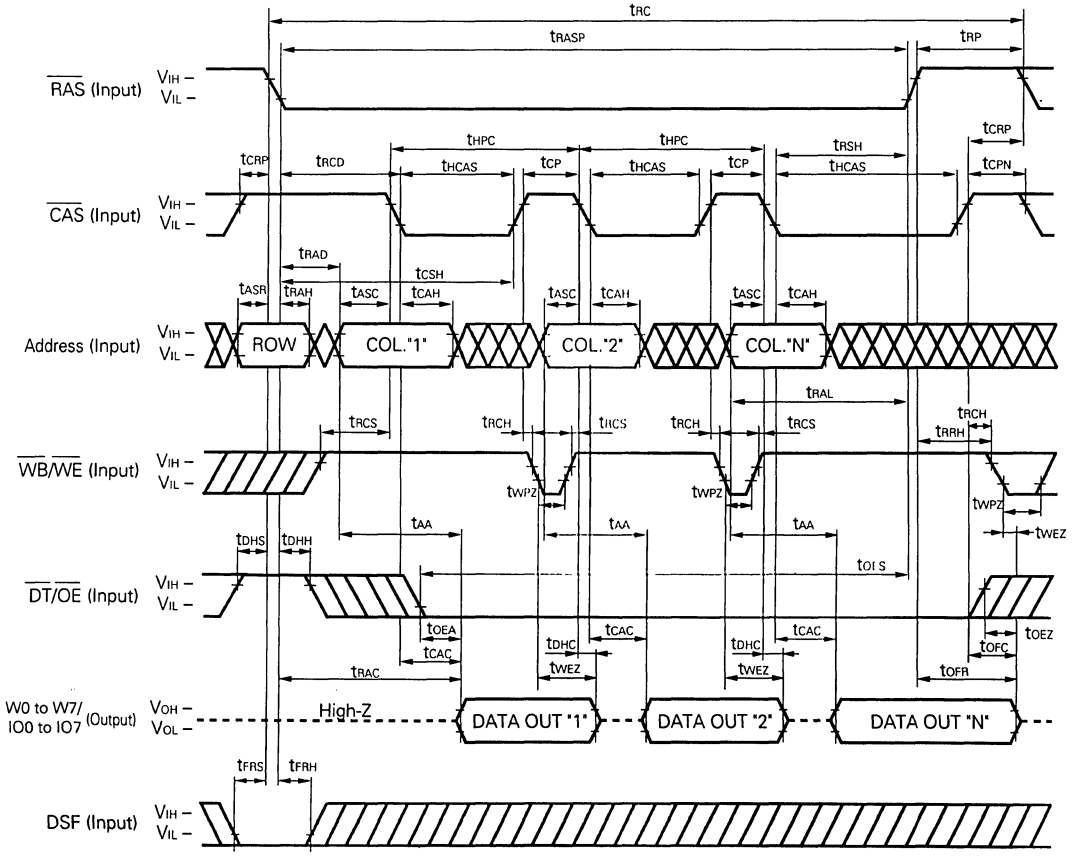

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Hyper Page Mode Read Cycle (Extended data output: μPD482235)**



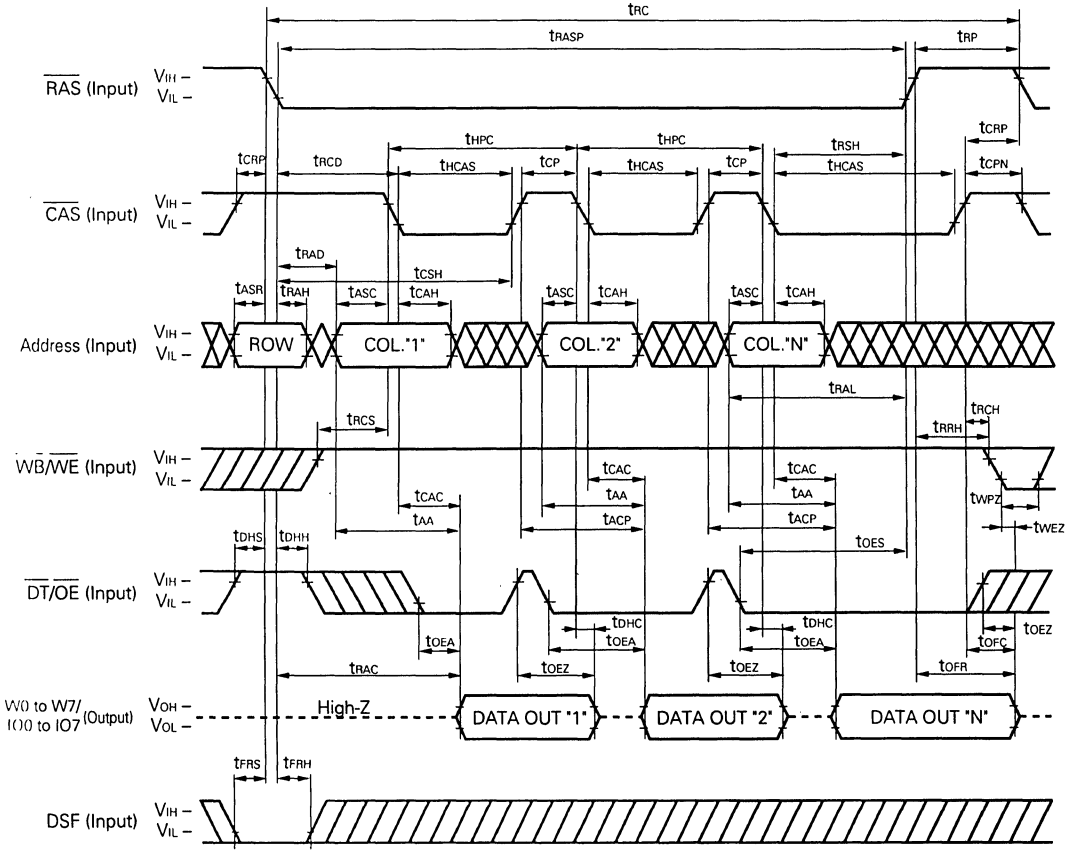

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Hyper Page Mode Read Cycle ($\overline{WE}$ controlled)**
**(Extended data output: μPD482235 (Versions A, F and E))**



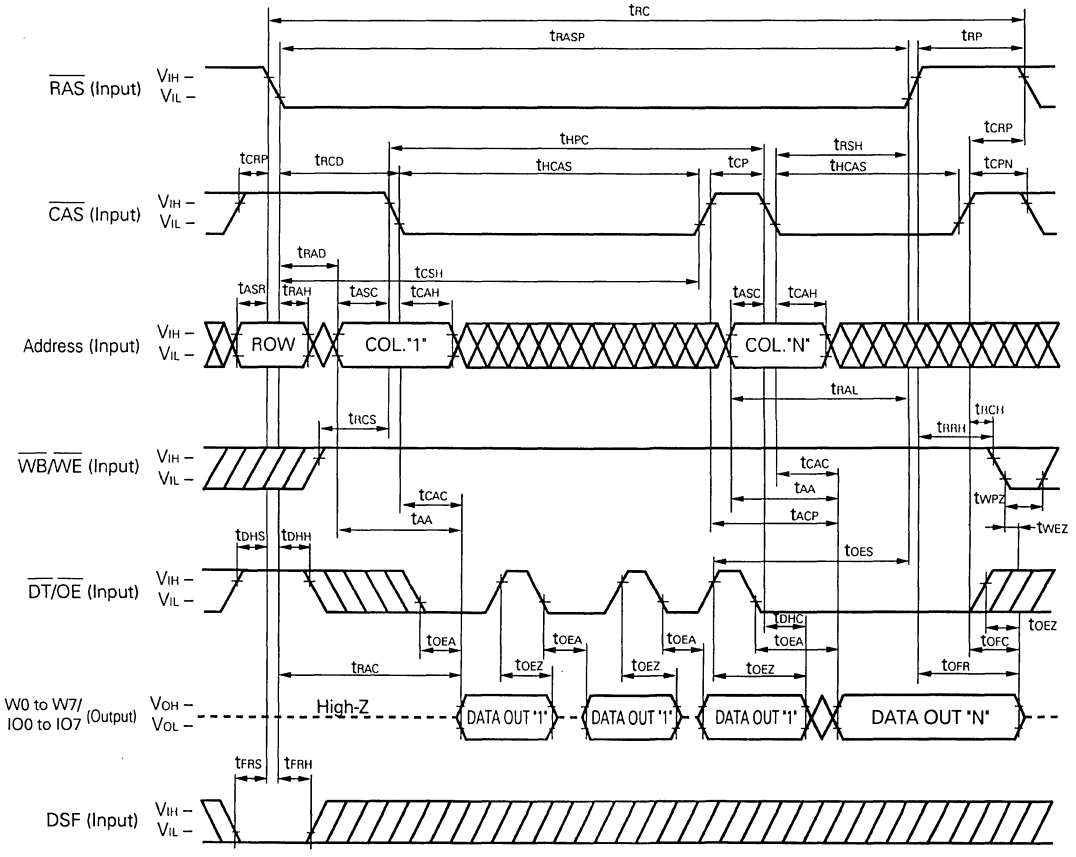

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Hyper Page Mode Read Cycle ($\overline{OE}$ controlled: Latched control)**
**(Extended data output: μPD482235 (Version A))**



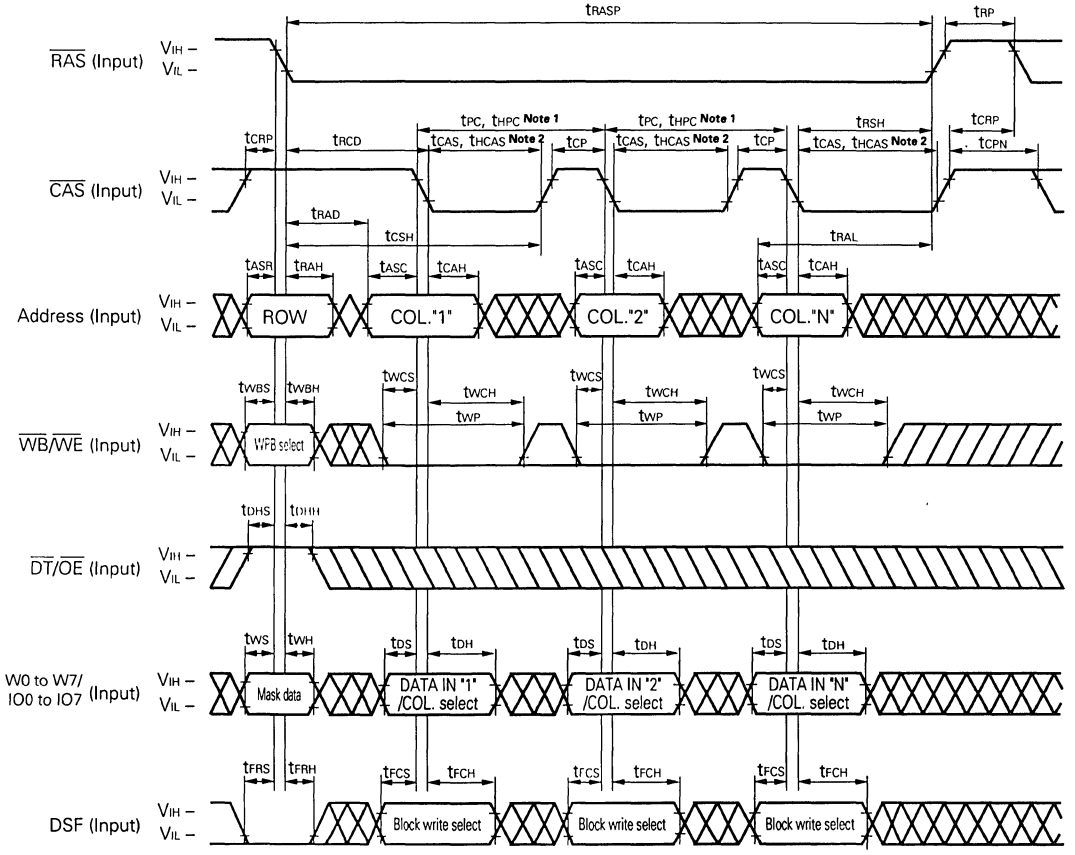

**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Hyper Page Mode Read Cycle ($\overline{OE}$ controlled: Non-Latched control)**
**(Extended data output: μPD482235 (Versions F, E))**




**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Fast Page, Hyper Page Mode Early Write Cycle/Fast Page, Hyper Page Mode Early Block Write Cycle**




**Notes 1.** tPC for the μPD482234
tHPC for the μPD482235

**2.** tCAS for the μPD482234
tHCAS for the μPD482235

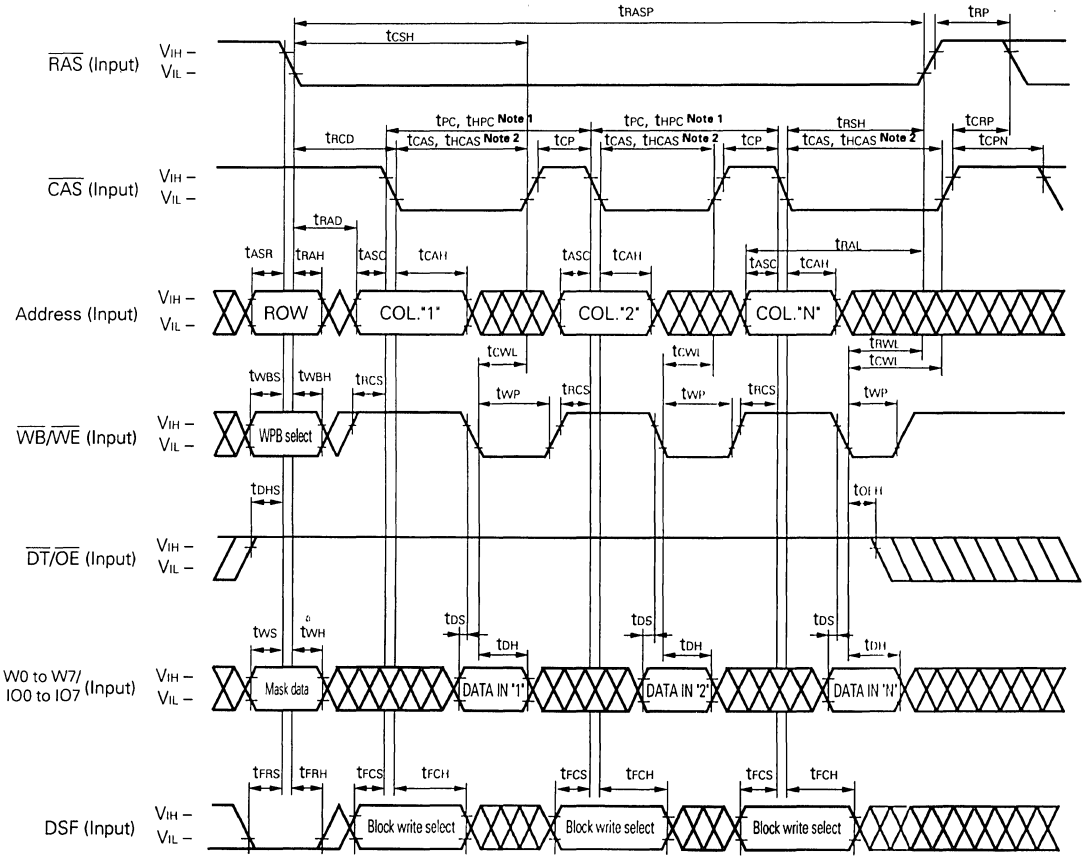
**Remarks 1.** When DSF is high level : Block write cycle
When DSF is low level : Write cycle

**2.** WPB : Write-per-bit

**3.** When block write cycle is selected, input the column selection data to DATA IN.

**4.** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Fast Page, Hyper Page Mode Late Write Cycle/Fast Page, Hyper Page Mode Late Block Write Cycle**




**Notes 1.** $t_{PC}$ for the μPD482234
$t_{HPC}$ for the μPD482235

**2.** $t_{CAS}$ for the μPD482234
$t_{HCAS}$ for the μPD482235

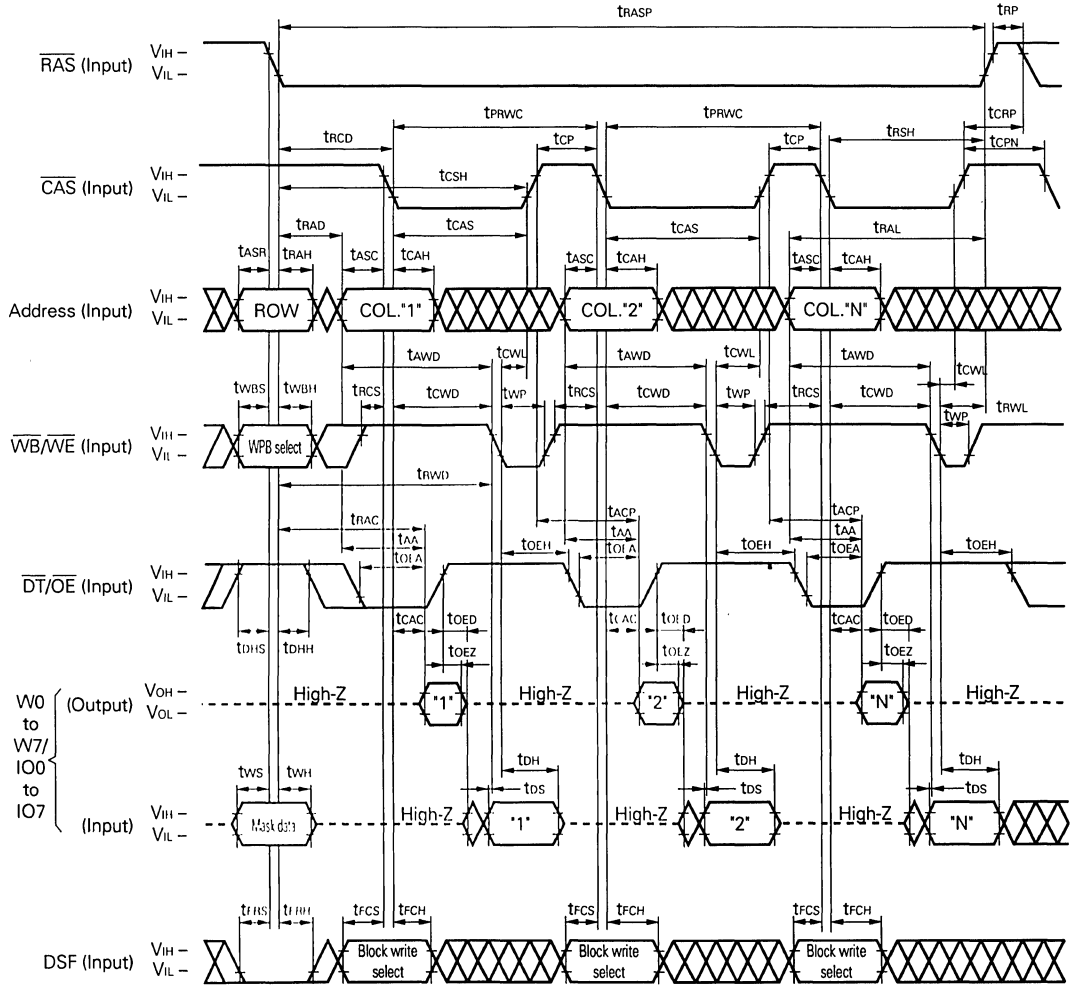
**Remarks 1.** When DSF is high level : Block write cycle
When DSF is low level : Write cycle

**2.** WPB : Write-per-bit

**3.** When block write cycle is selected, input the column selection data to DATA IN.

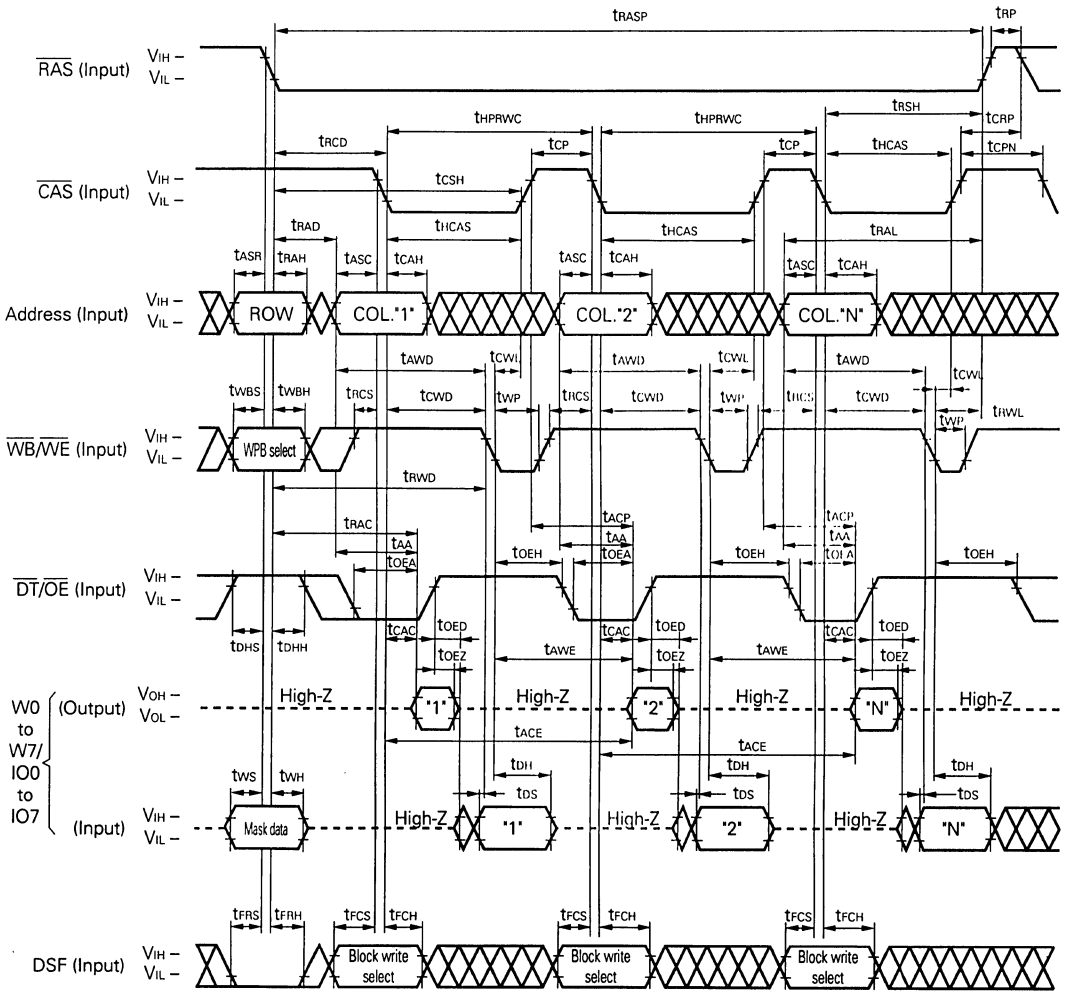
**4.** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Fast Page Mode Read Modify Write Cycle (μPD482234)/**
**Fast Page Mode Read Modify Block Write Cycle (μPD482234)**



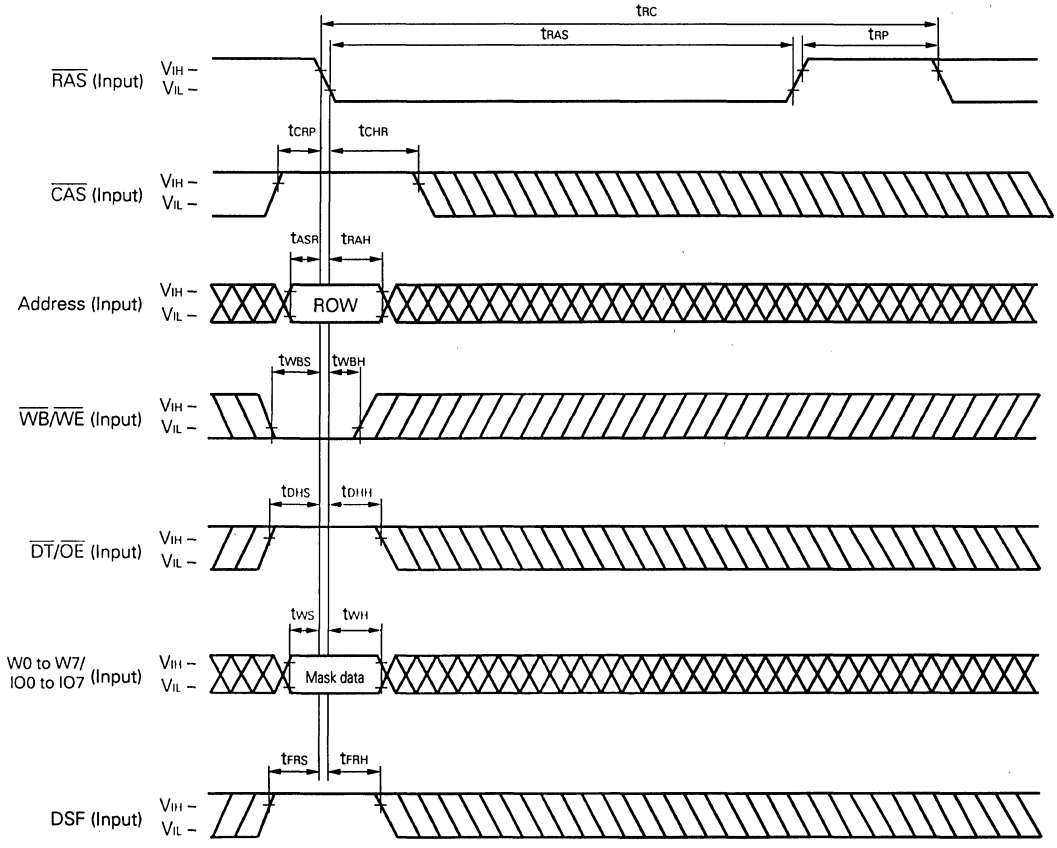

**Remarks**
1. When DSF is high level : Block write cycle
   When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Hyper Page Mode Read Modify Write Cycle (Extended data output: μPD482235)/**
**Hyper Page Mode Read Modify Block Write Cycle (Extended data output: μPD482235)**



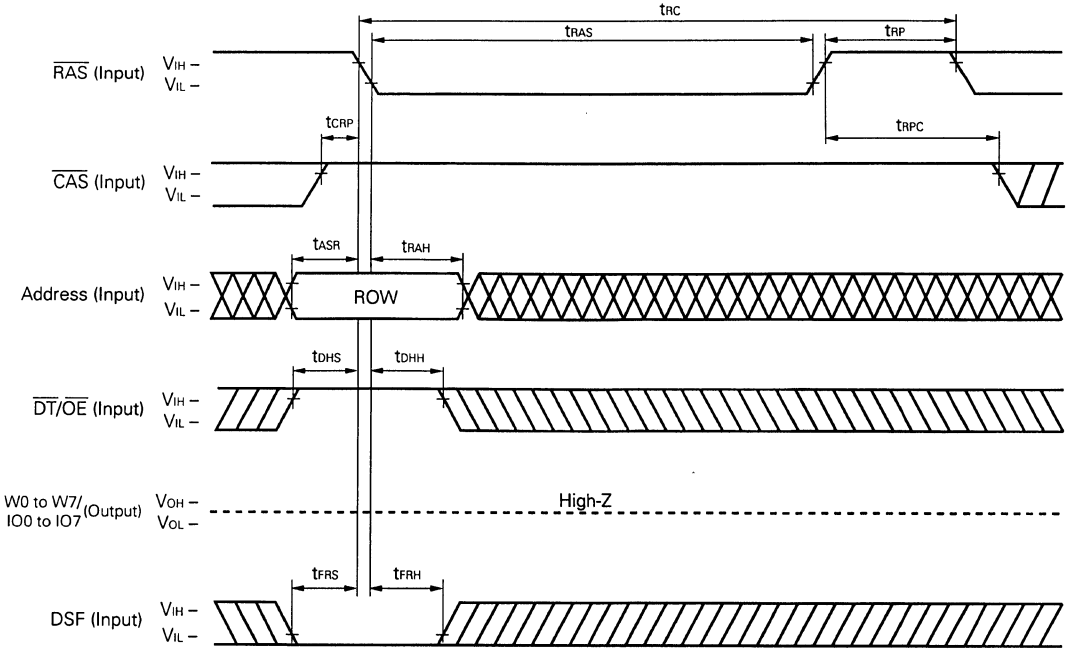

**Remarks** 1. When DSF is high level : Block write cycle
When DSF is low level : Write cycle
2. WPB : Write-per-bit
3. When block write cycle is selected, input the column selection data to DATA IN.
4. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Flash Write Cycle**




**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

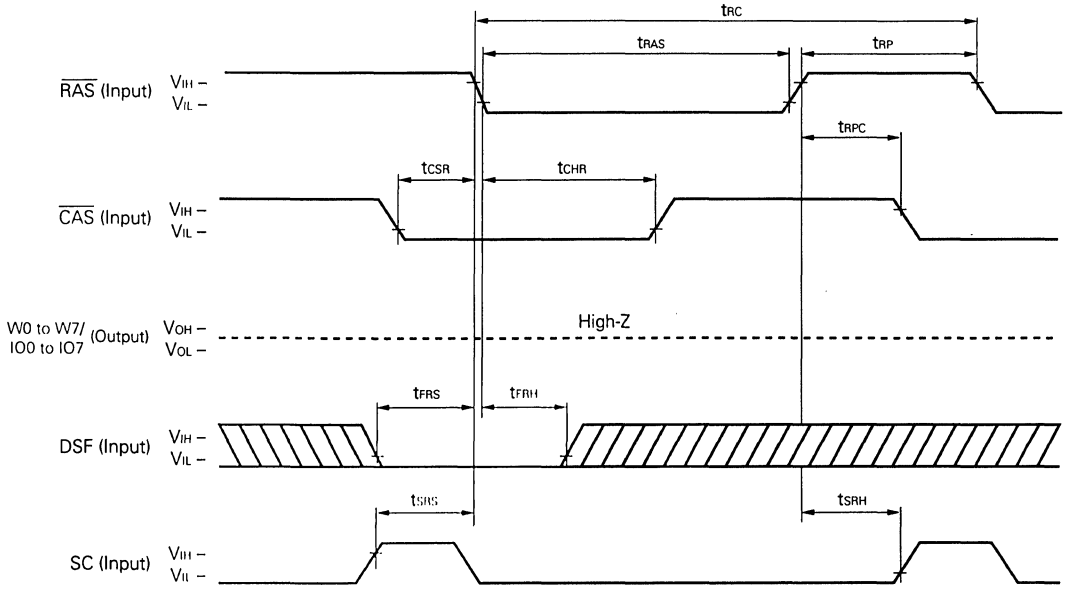
**$\overline{RAS}$ Only Refresh Cycle**




**Remarks** 1. $\overline{WB}/\overline{WE}$ : Don't care

2. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

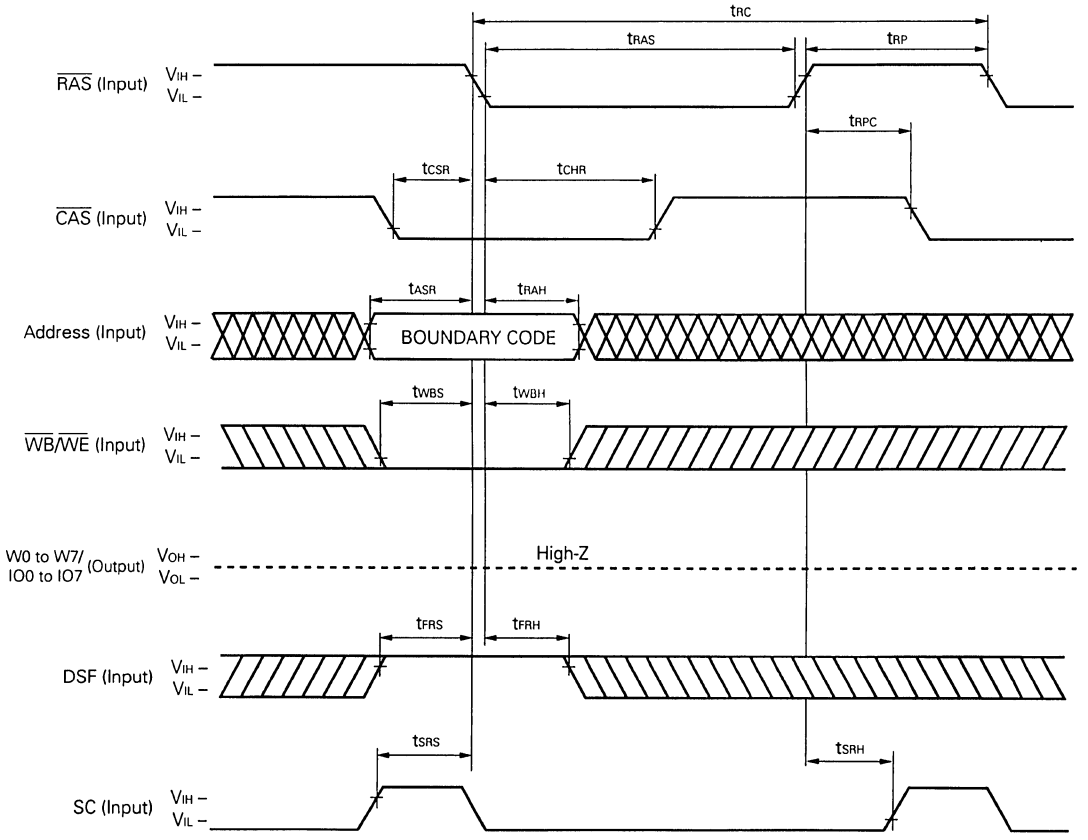
**$\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (Optional Reset)**




**Remarks** 1. Address, $\overline{WB}/\overline{WE}$, $\overline{DT}/\overline{OE}$ : Don't care

2. Because the serial access port operates independently of the random access port, there is no need to control the $\overline{SE}$, SIO pins in this cycle.

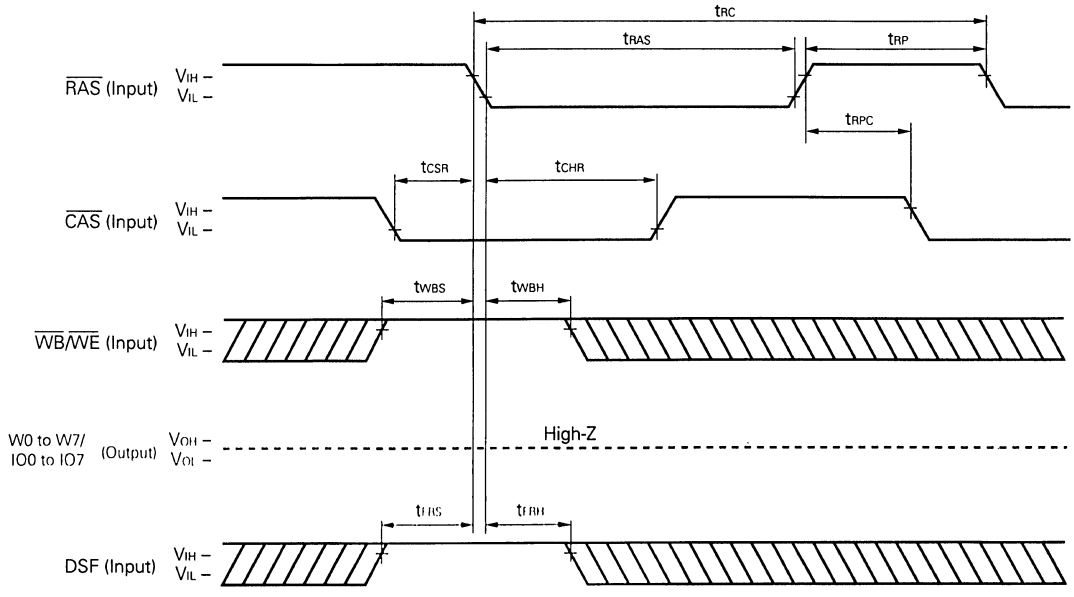
**$\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (STOP Register Set)**




**Remarks** 1. $\overline{DT}/\overline{OE}$ : Don't care

2. Because the serial access port operates independently of the random access port, there is no need to control the $\overline{SE}$, SIO pins in this cycle.

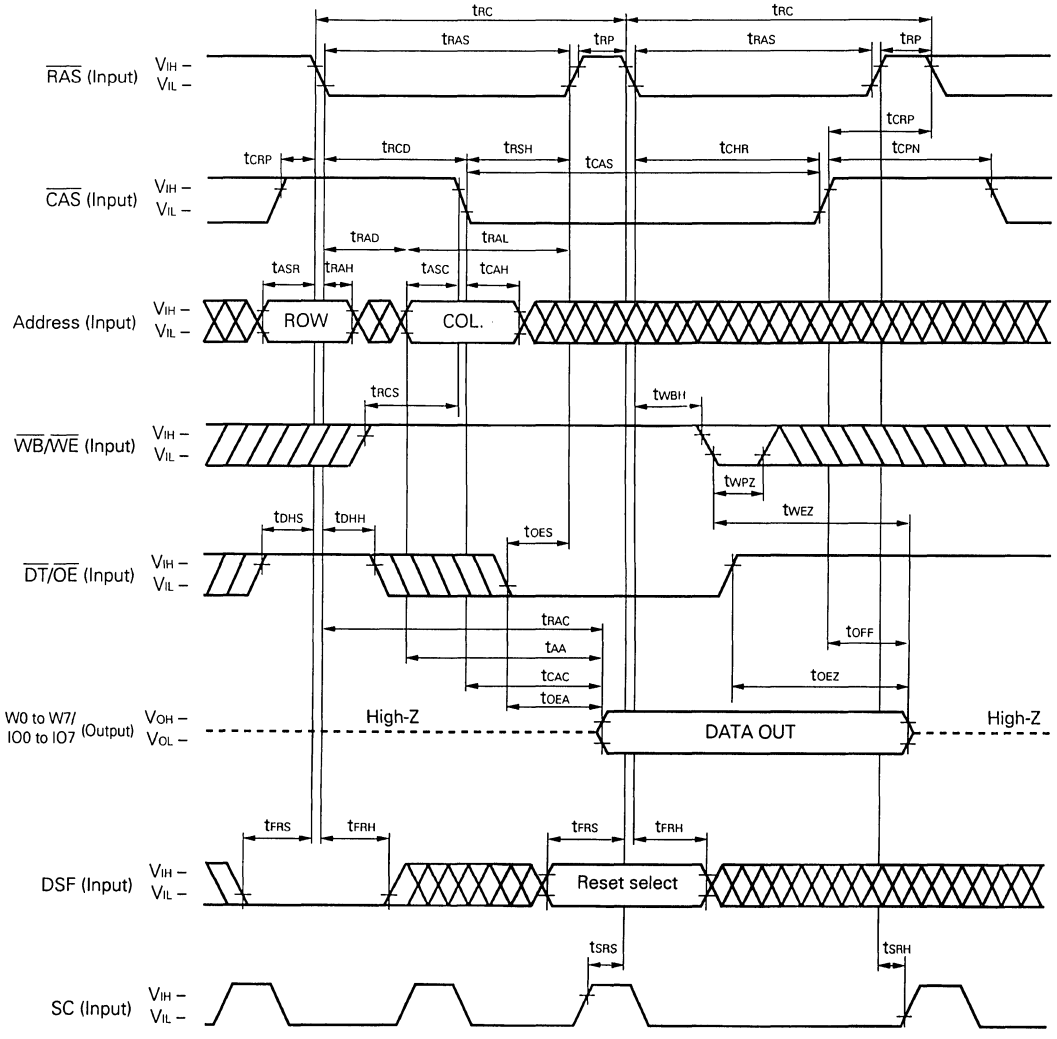
**$\overline{CAS}$ Before $\overline{RAS}$ Refresh Cycle (No Reset)**




**Remarks** 1. Address, $\overline{DT}/\overline{OE}$ : Don't care

2. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

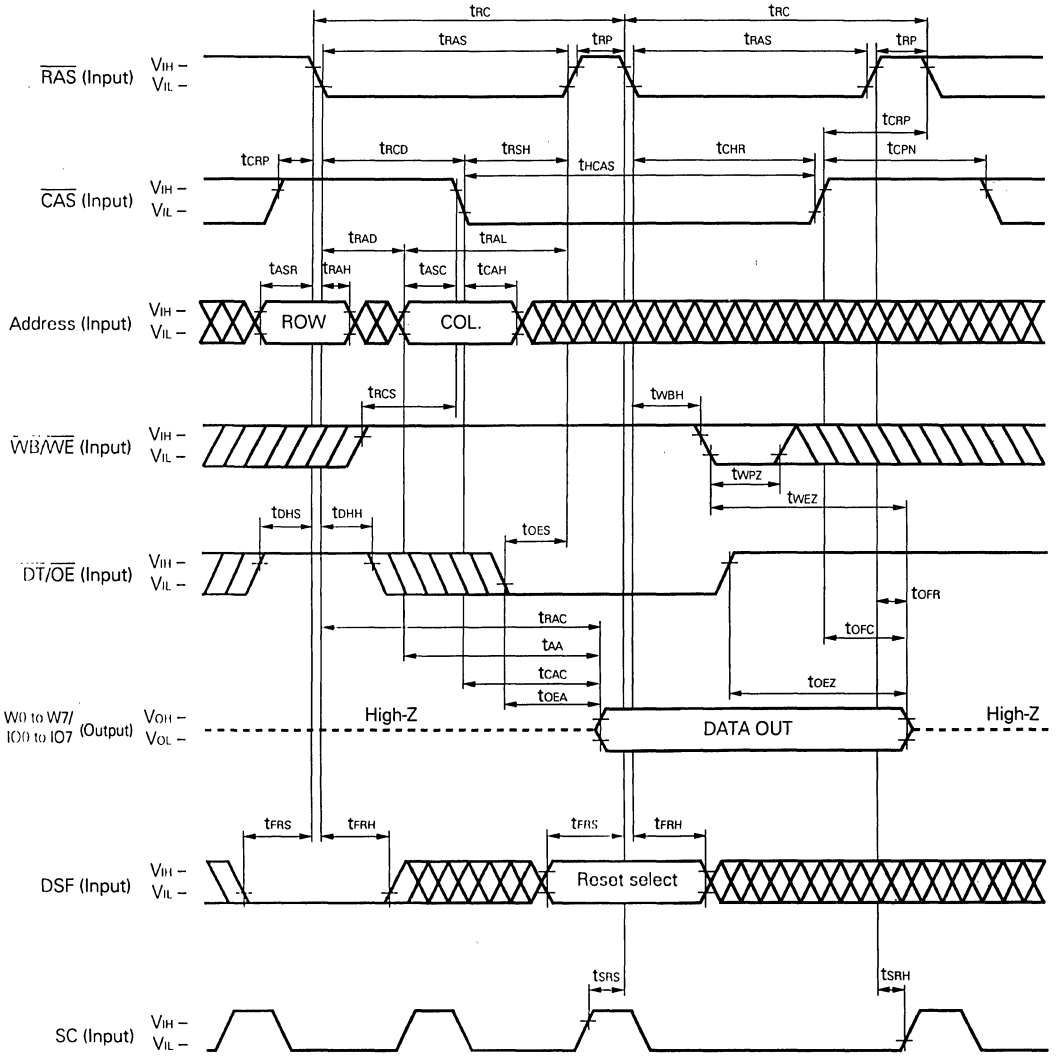
**Hidden Refresh Cycle (μPD482234)**




**Remarks** 1. When DSF is high level : Reset select = No Reset
When DSF is low level : Reset select = Optional Reset

2. Because the serial access port operates independently of the random access port, there is no need to control the $\overline{SE}$, SIO pins in this cycle.

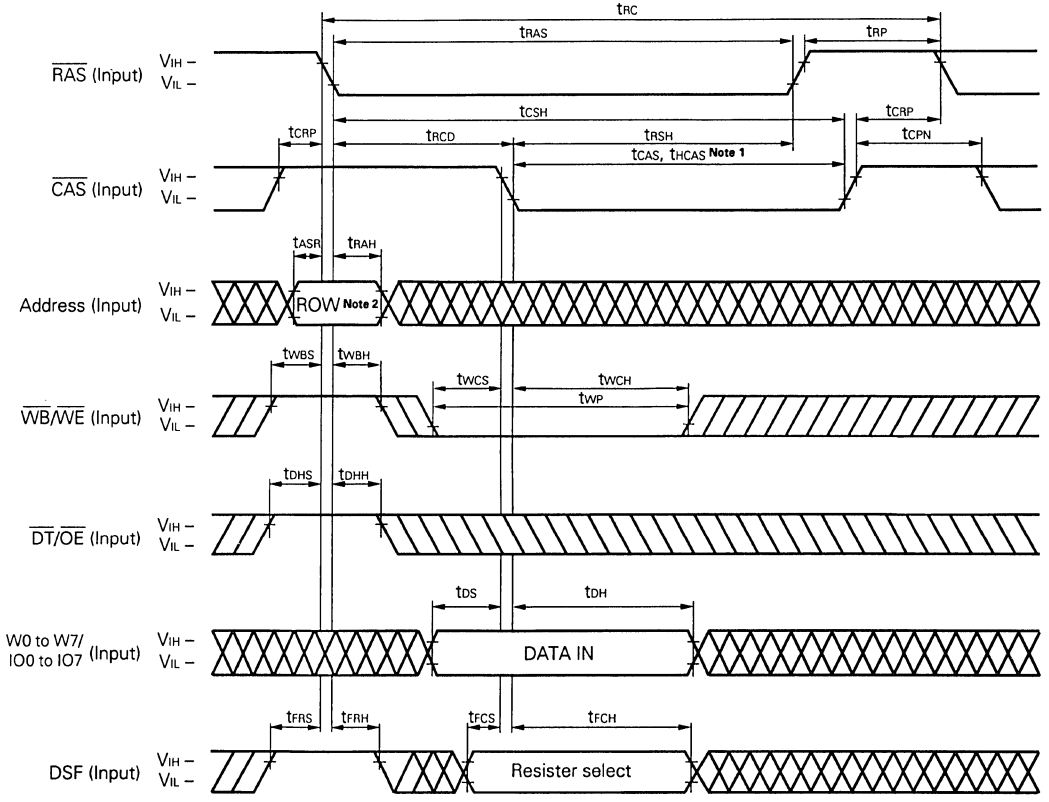
**Hidden Refresh Cycle (Extended data output: μPD482235)**




**Remarks 1.** When DSF is high level : Reset select = No Reset
When DSF is low level : Reset select = Optional Reset

**2.** Because the serial access port operates independently of the random access port, there is no need to control the $\overline{SE}$, SIO pins in this cycle.

**Register Set Cycle (Early Write)**



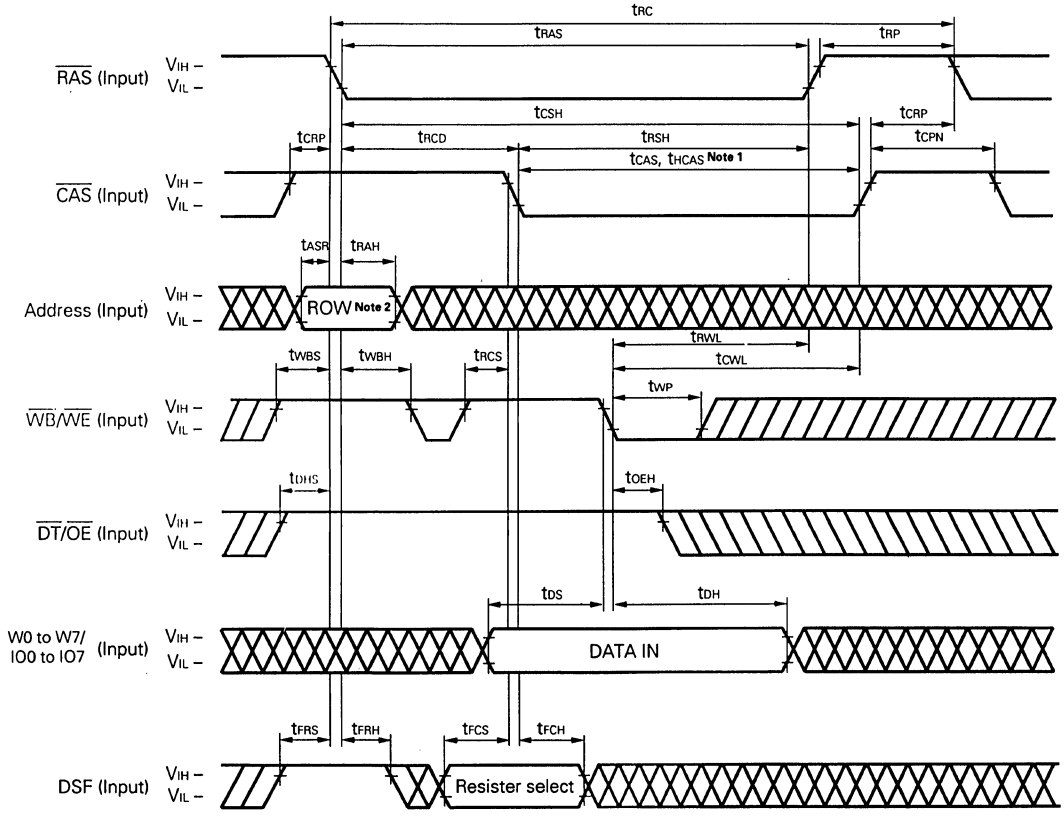

**Notes 1.** $t_{CAS}$ for the μPD482234
$t_{HCAS}$ for the μPD482235

**2.** Refresh address ($\overline{RAS}$ only refresh)

**Remarks 1.** When DSF is high level : Register select = Color Register Select
When DSF is low level : Register select = Write Mask Register Select

**2.** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

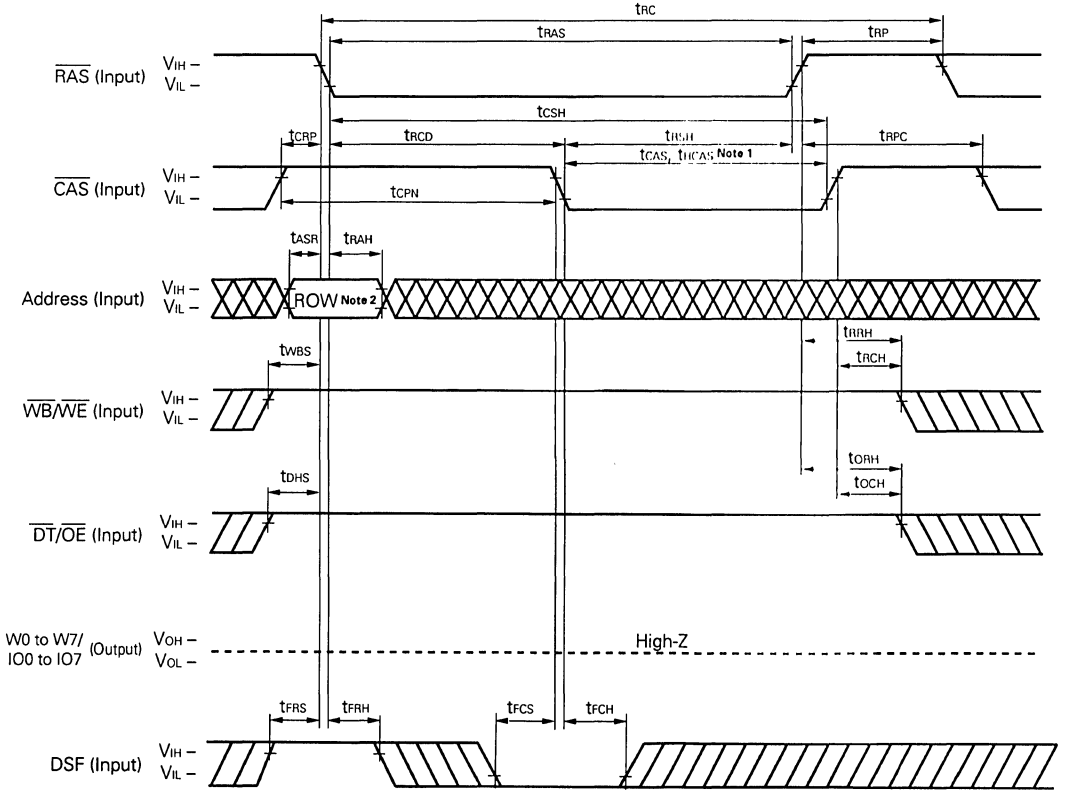
**Register Set Cycle (Late Write)**




**Notes** 1. $t_{CAS}$ for the μPD482234
$t_{HCAS}$ for the μPD482235
2. Refresh address ($\overline{RAS}$ only refresh)

**Remarks** 1. When DSF is high level : Register select = Color Register Select
When DSF is low level : Register select = Write Mask Register Select
2. Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Mask Register Set Cycle (New Mask Selection) (Versions F and E only)**

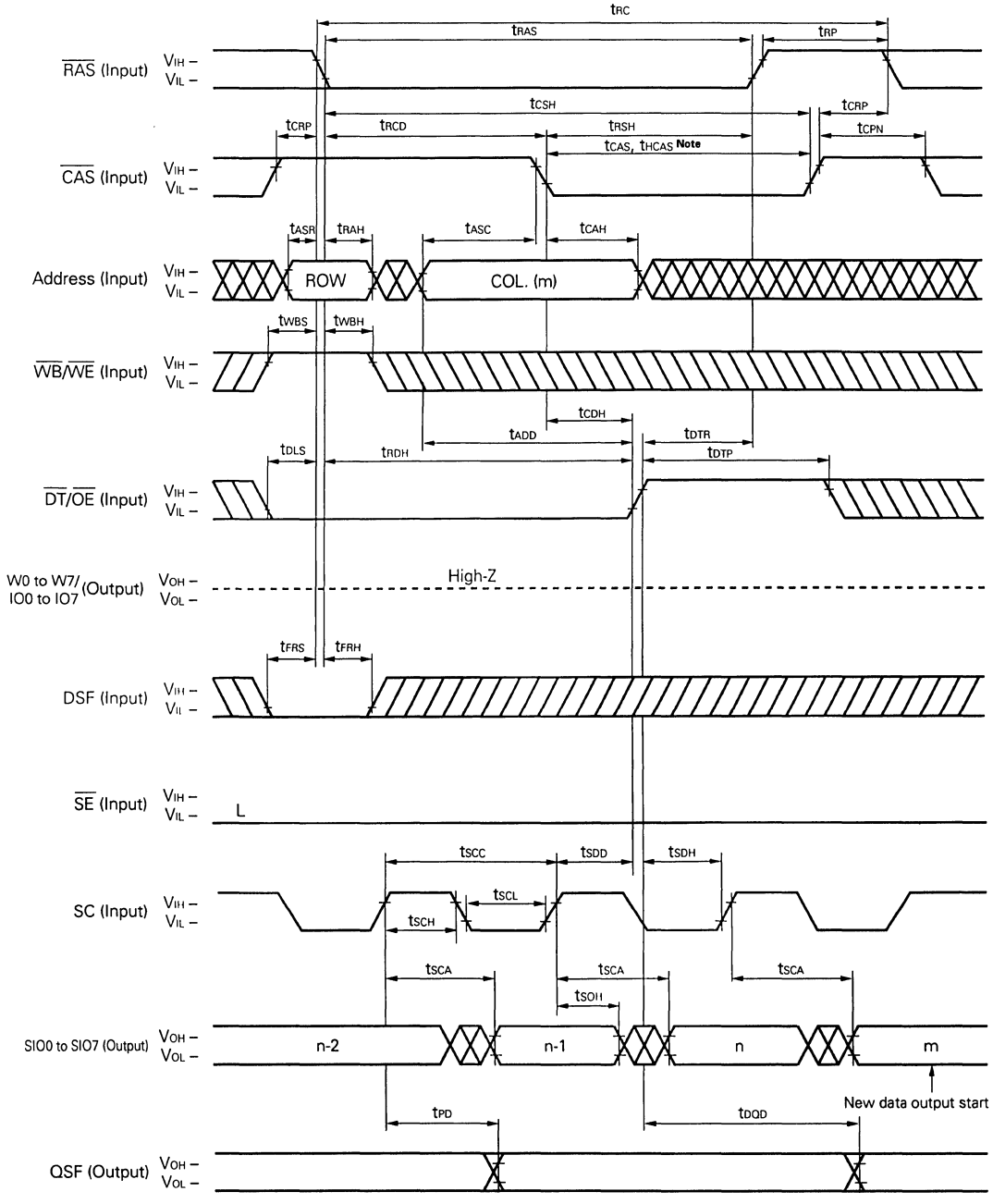



**Notes 1.** $t_{CAS}$ for the μPD482234
$t_{HCAS}$ for the μPD482235

**2.** Refresh address ($\overline{RAS}$ only refresh)

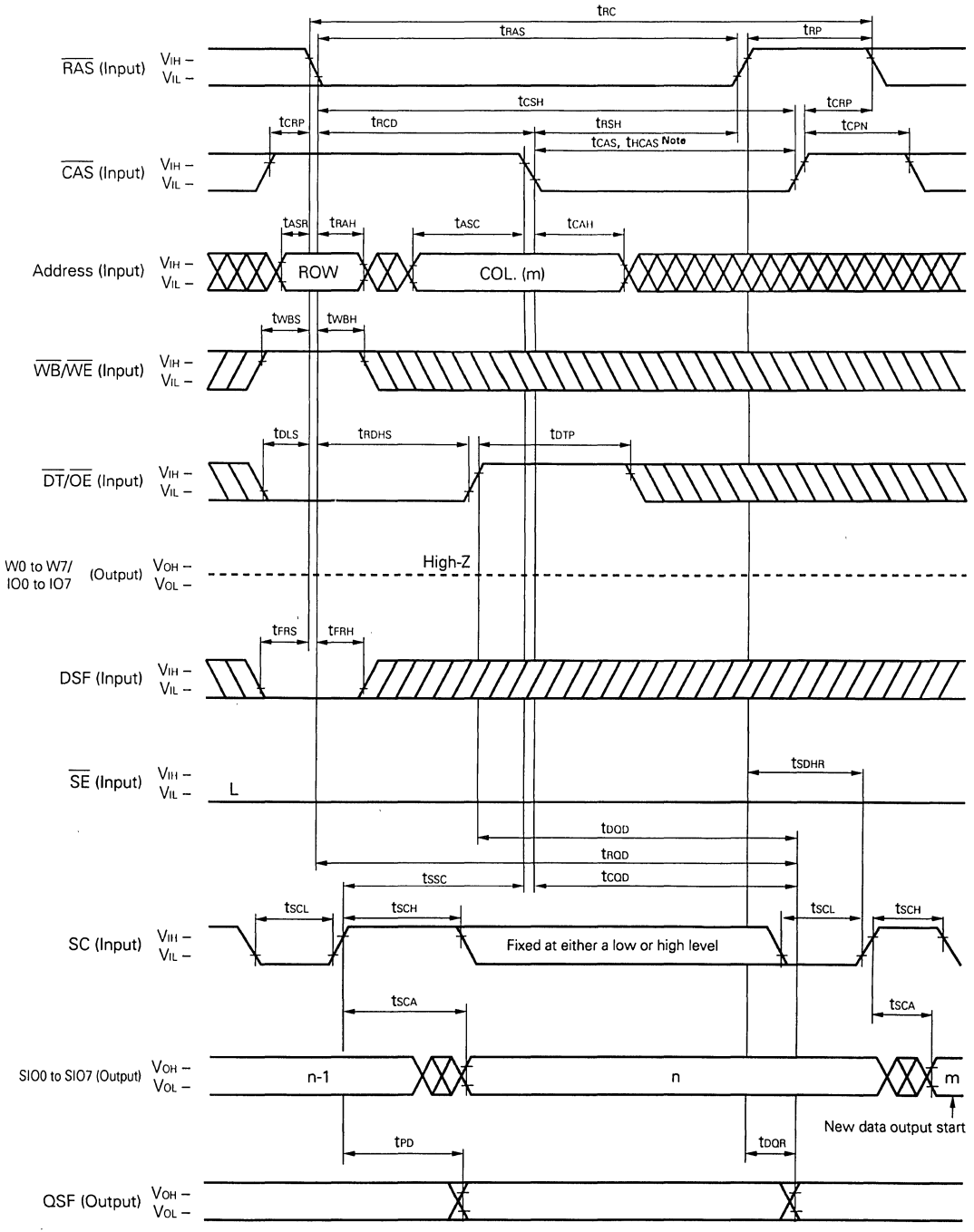
**Remark** Because the serial access port operates independently of the random access port, there is no need to control the SC, $\overline{SE}$, SIO pins in this cycle.

**Read Data Transfer Cycle (SC Active)**



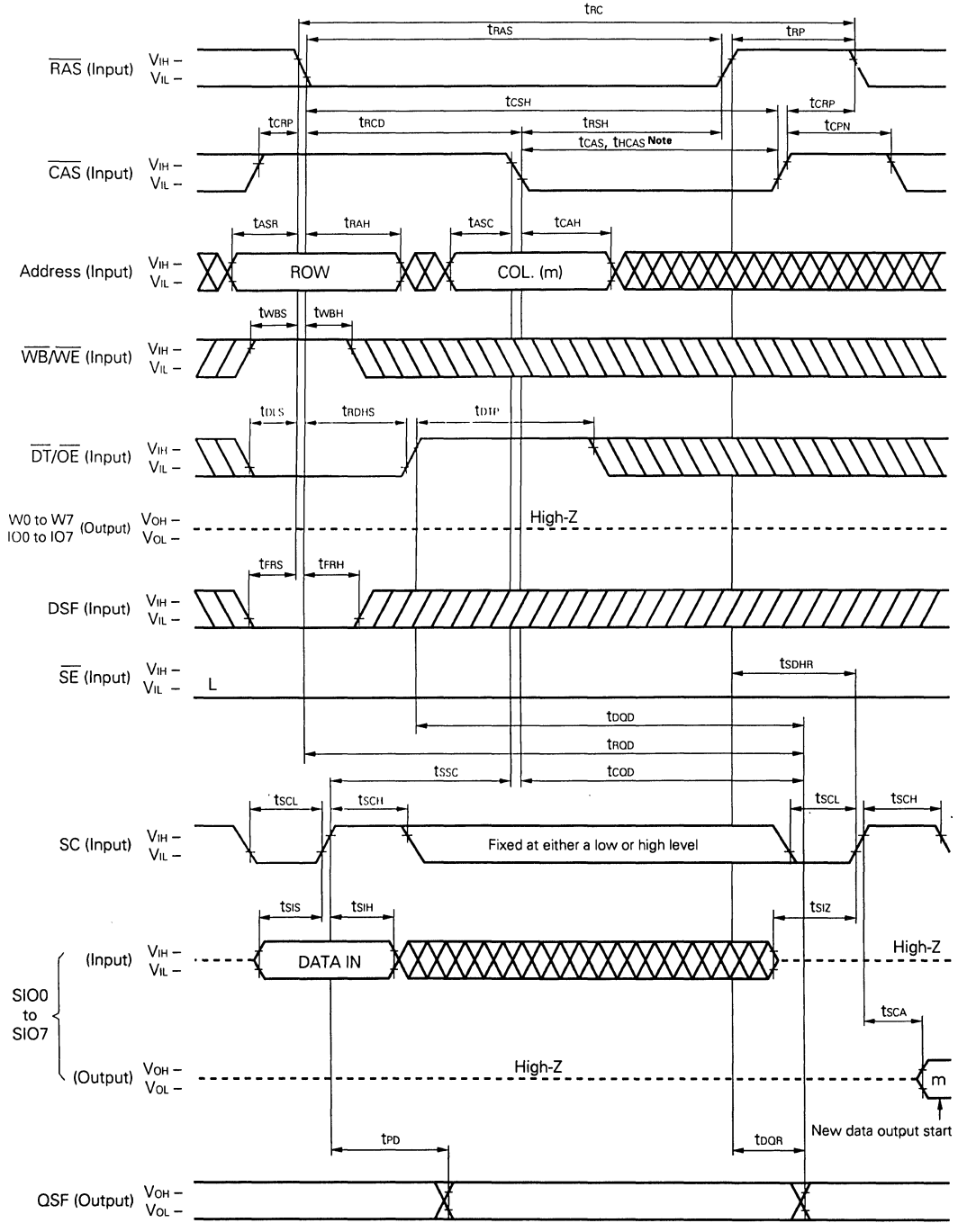

**Note** $t_{CAS}$ for the μPD482234
$t_{HCAS}$ for the μPD482235

**Read Data Transfer Cycle (SC Inactive)**



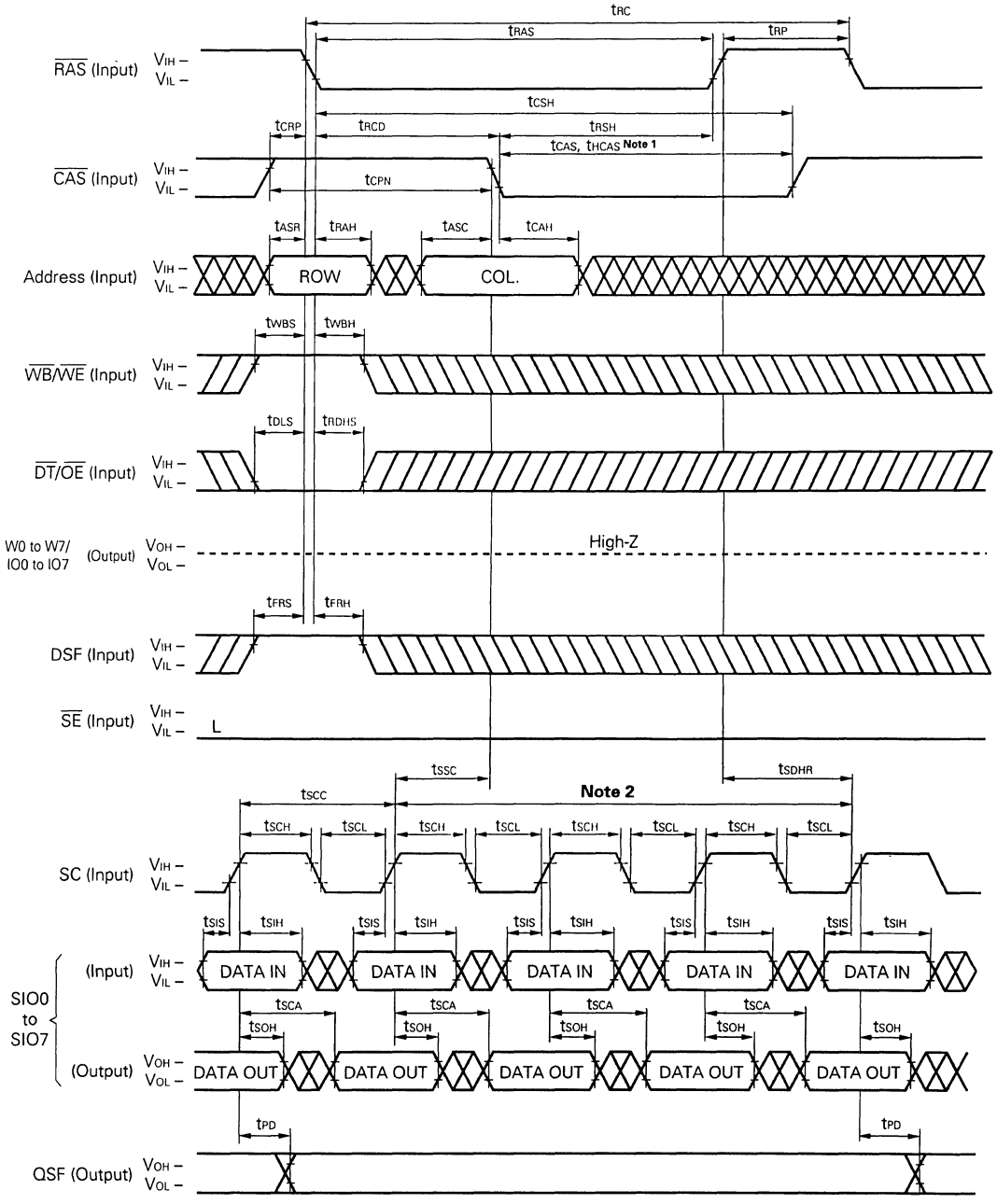

**Note** tCAS for the μPD482234
tHCAS for the μPD482235

**Read Data Transfer Cycle (Serial Write → Serial Read Switching)**




**Note** tCAS for the μPD482234
tHCAS for the μPD482235

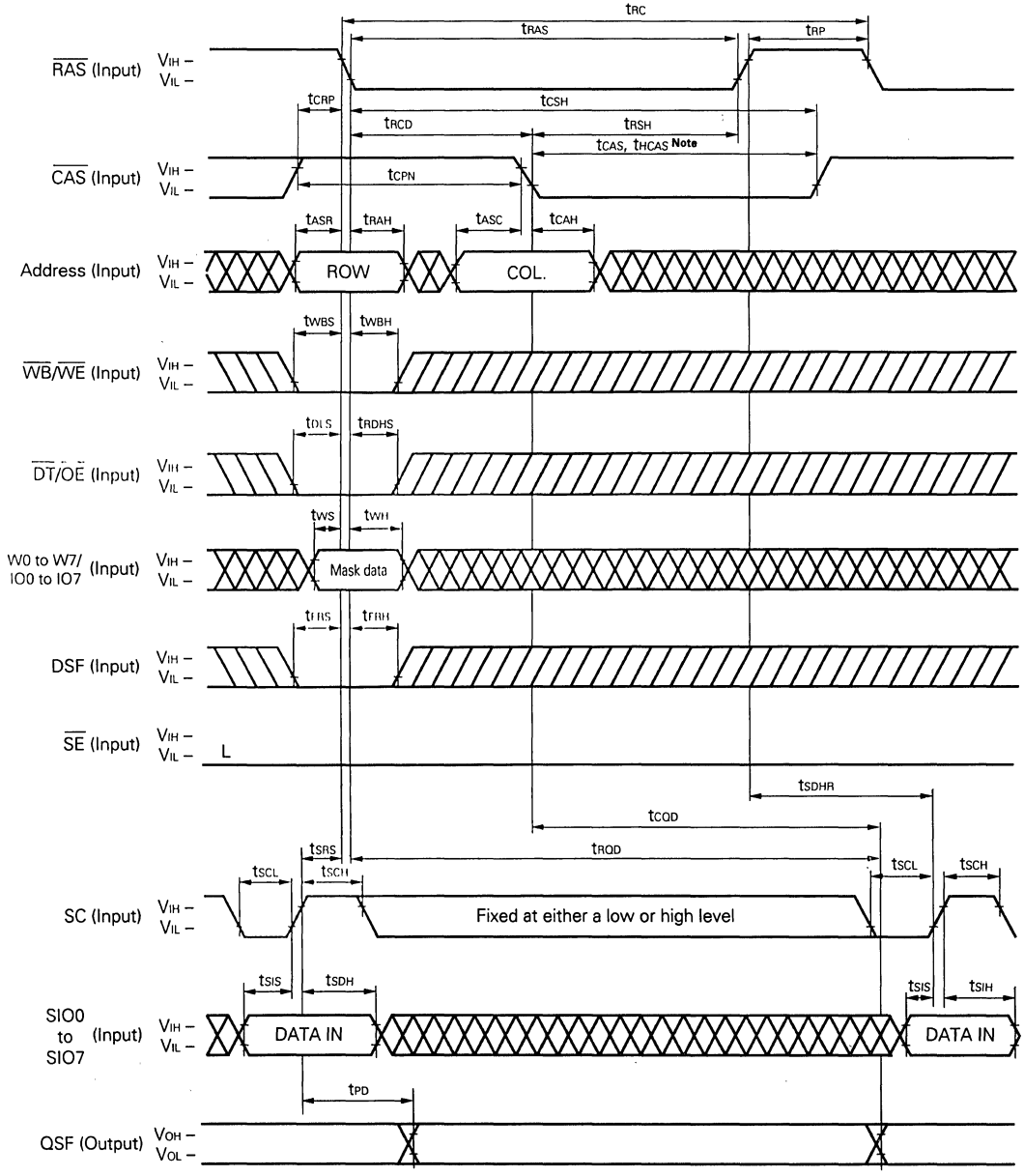
**Split Read Data Transfer Cycle**




**Notes** 1. $t_{CAS}$ for the μPD482234
$t_{HCAS}$ for the μPD482235

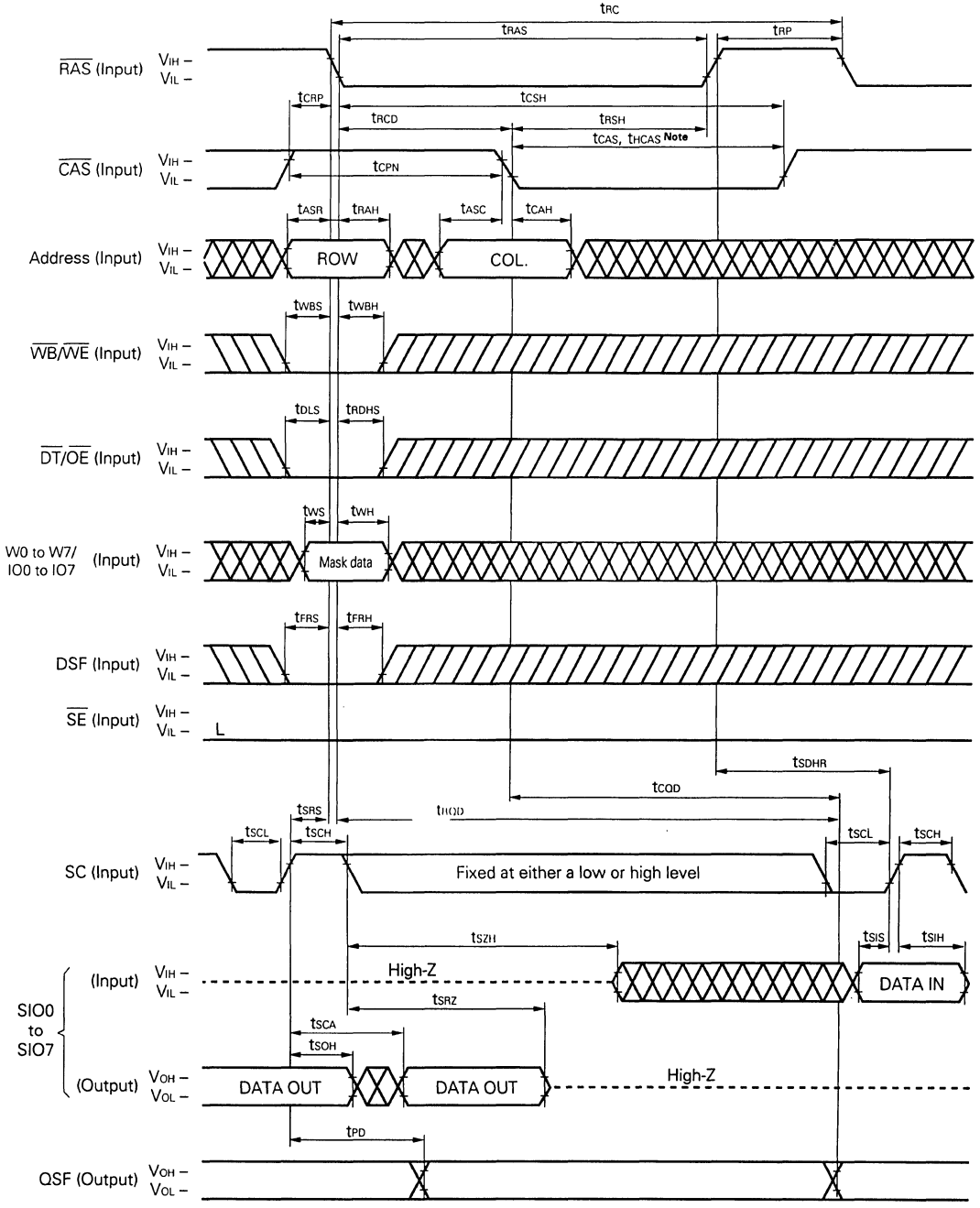
2. Do not perform the following two serial read/write during this period.
   - Serial read/write of jump source address set to the STOP register of the data register which does not perform the data transfer cycle.
   - Serial read/write of last address of data register (Address 255 or 511)
   - Data register serial read for the side to be involved in data transfer cycle (for version E only)

**Write Data Transfer Cycle**



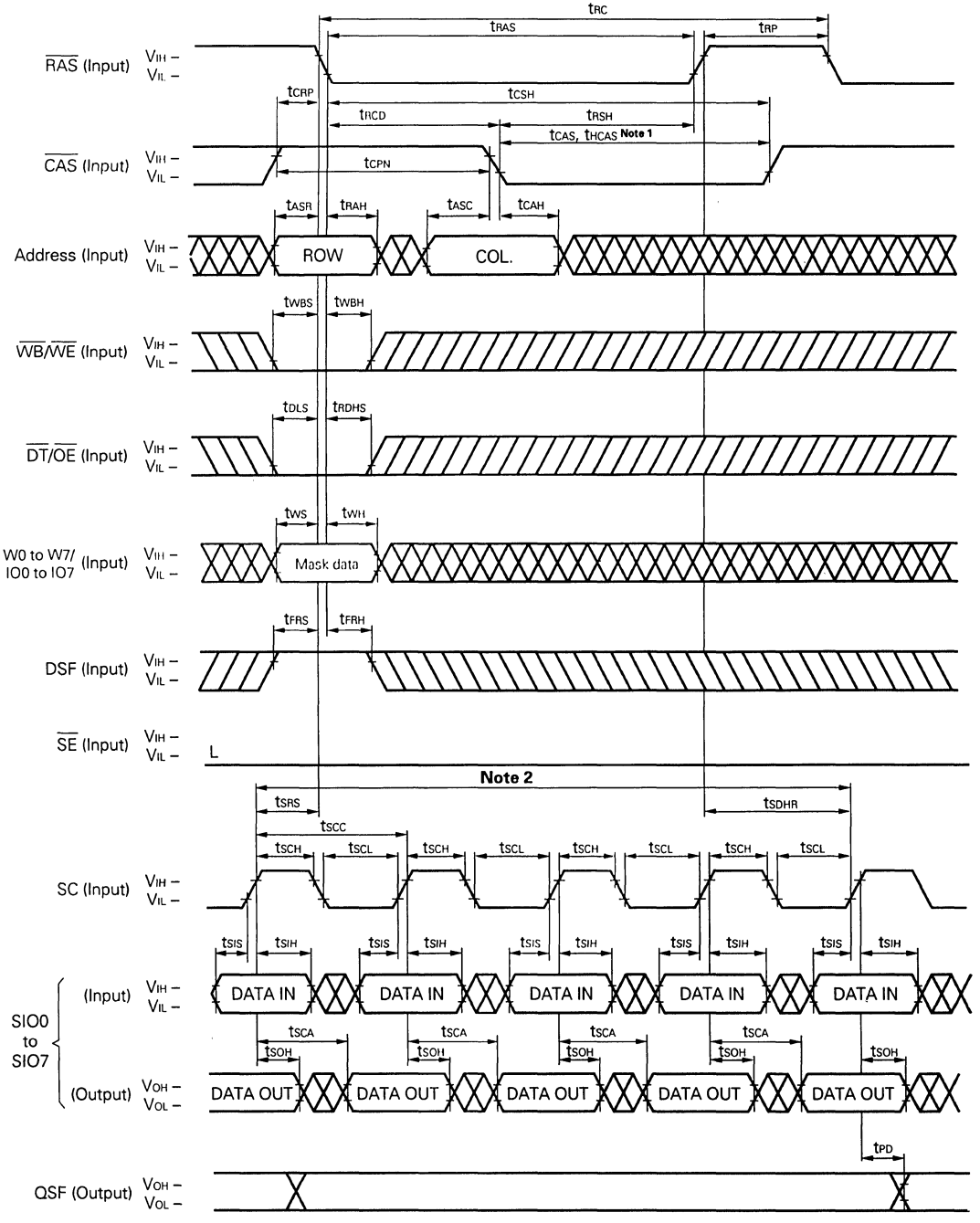

**Note** tCAS for the μPD482234
tHCAS for the μPD482235

**Write Data Transfer Cycle (Serial Read → Serial Write Switching)**




**Note** tCAS for the μPD482234
tHCAS for the μPD482235

**Split Write Data Transfer Cycle**

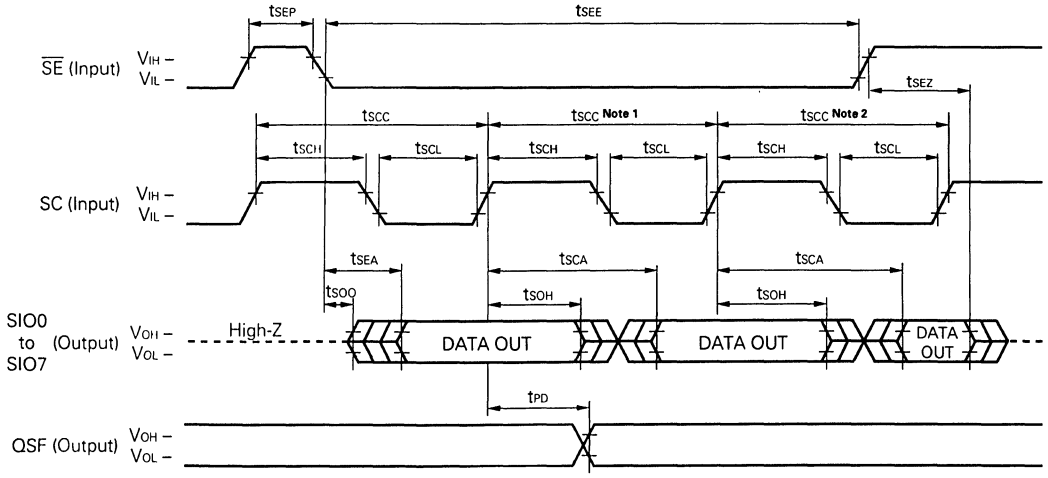


**Notes 1.** $t_{CAS}$ for the μPD482234
$t_{HCAS}$ for the μPD482235

**2.** Do not perform the following two serial read/write during this period.

- Serial read/write of jump source address set to the STOP register of the data register which does not perform the data transfer cycle.
- Serial read/write of last address of data register (Address 255 or 511)
- Data register serial read for the side to be involved in data transfer cycle (for version E only)

**Serial Read Cycle**

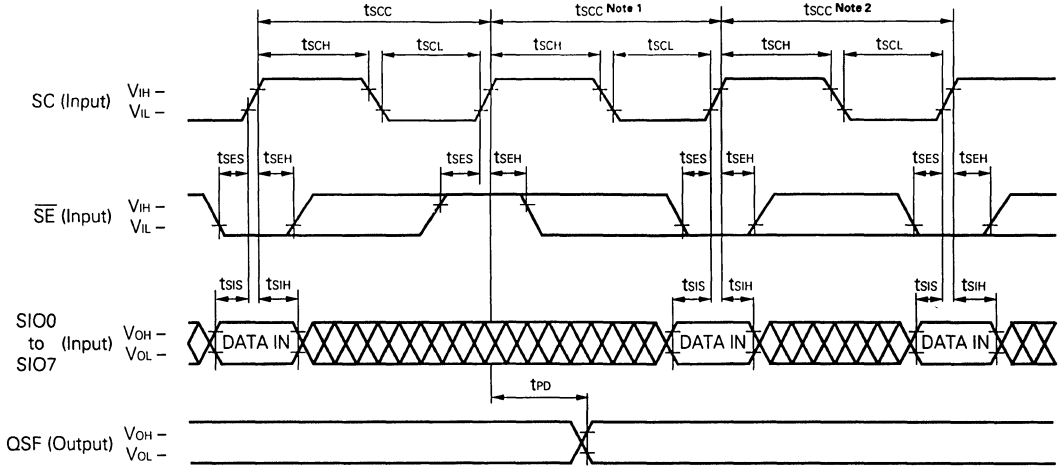



**Notes 1.** Last address of data register (Address 255 or 511)

**2.** Starting address of data register newly read (address is specified in the data transfer cycle).

**Remark** Because the random access port operates independently of the serial access port, there is no need to control the $\overline{RAS}$, $\overline{CAS}$, Address, $\overline{WB}/\overline{WE}$, $\overline{DT}/\overline{OE}$, WI/O, DSF pins in this cycle.

**Serial Write Cycle**



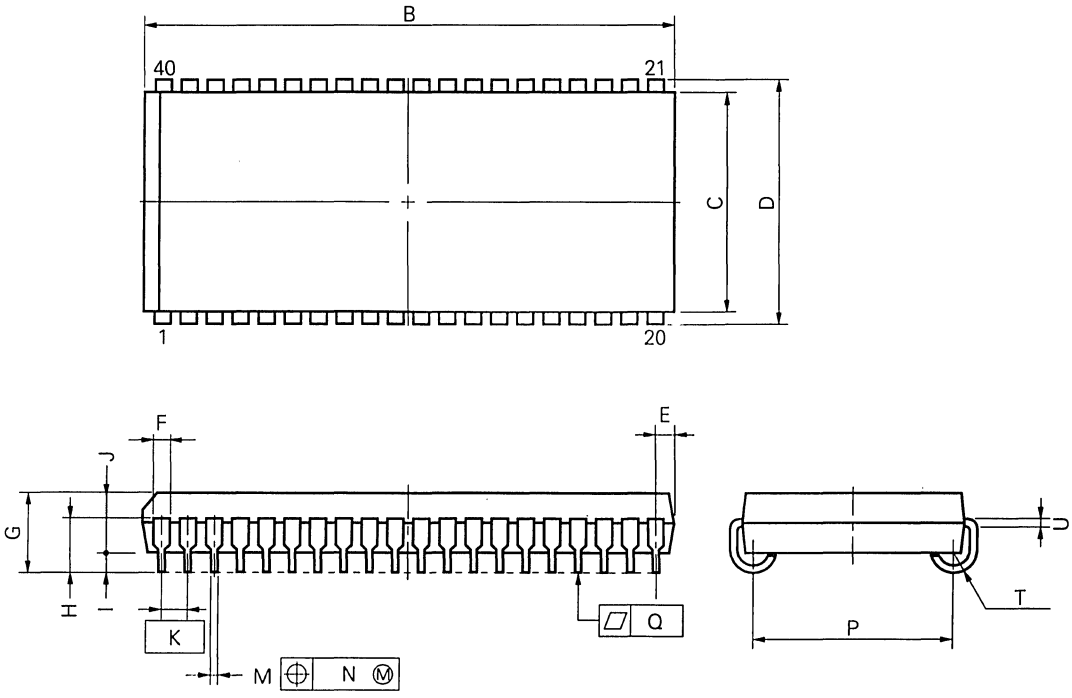

**Notes 1.** Last address of data register (Address 255 or 511)

**2.** Starting address of data register newly read (address is specified in the data transfer cycle).

**Remark** Because the random access port operates independently of the serial access port, there is no need to control the $\overline{RAS}$, $\overline{CAS}$, Address, $\overline{WB}/\overline{WE}$, $\overline{DT}/\overline{OE}$, WI/O, DSF pins in this cycle.

## 5. Package Drawings

### 40 PIN PLASTIC SOJ (400 mil)

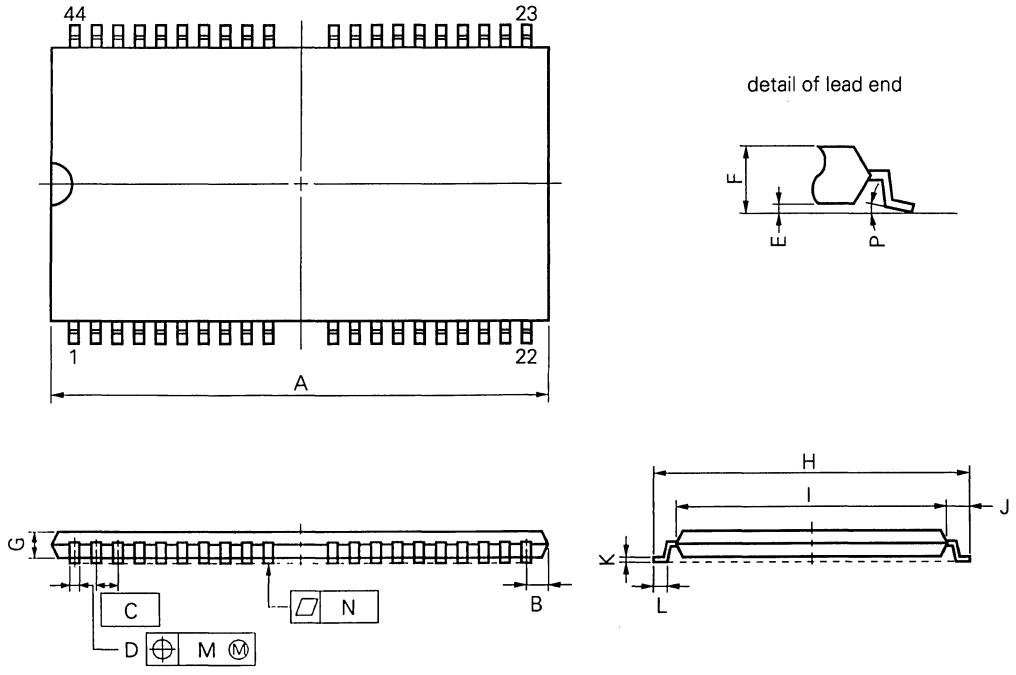



**NOTE**
Each lead centerline is located within 0.12 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| B | $26.29^{+0.2}_{-0.35}$ | $1.035^{+0.008}_{-0.014}$ |
| C | 10.16 | 0.400 |
| D | 11.18±0.2 | 0.440±0.008 |
| E | 1.08±0.15 | $0.043^{+0.006}_{-0.007}$ |
| F | 0.7 | 0.028 |
| G | 3.5±0.2 | 0.138±0.008 |
| H | 2.4±0.2 | $0.094^{+0.009}_{-0.008}$ |
| I | 0.8 MIN. | 0.031 MIN. |
| J | 2.6 | 0.102 |
| K | 1.27(T.P.) | 0.050(T.P.) |
| M | 0.40±0.10 | $0.016^{+0.004}_{-0.005}$ |
| N | 0.12 | 0.005 |
| P | 9.40±0.20 | 0.370±0.008 |
| Q | 0.15 | 0.006 |
| T | R0.85 | R0.033 |
| U | $0.20^{+0.10}_{-0.05}$ | $0.008^{+0.004}_{-0.002}$ |

**P40LE-400A-2**

## 44 PIN PLASTIC TSOP(II) (400 mil)




**NOTE**

Each lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 18.63 MAX. | 0.734 MAX. |
| B | 0.93 MAX. | 0.037 MAX. |
| C | 0.8 (T.P.) | 0.031 (T.P.) |
| D | $0.32^{+0.08}_{-0.07}$ | 0.013±0.003 |
| E | 0.1±0.05 | 0.004±0.002 |
| F | 1.2 MAX. | 0.048 MAX. |
| G | 0.97 | 0.038 |
| H | 11.76±0.2 | 0.463±0.008 |
| I | 10.16±0.1 | 0.400±0.004 |
| J | 0.8±0.2 | $0.031^{+0.009}_{-0.008}$ |
| K | $0.145^{+0.025}_{-0.015}$ | 0.006±0.001 |
| L | 0.5±0.1 | $0.020^{+0.004}_{-0.005}$ |
| M | 0.13 | 0.005 |
| N | 0.10 | 0.004 |
| P | $3°^{+7°}_{-3°}$ | $3°^{+7°}_{-3°}$ |

S44G5-80-7JF4

## 6. Recommended Soldering Conditions

Please consult with our sales offices for soldering conditions of the μPD482234, μPD482235.

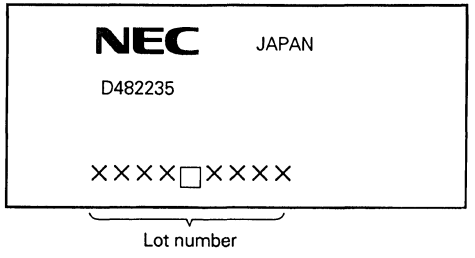
**Types of Surface Mount Device**

μPD482234LE-xx : 40-pin plastic SOJ (400 mil)
μPD482235LE-xx : 40-pin plastic SOJ (400 mil)

μPD482234G5-xx : 44-pin plastic TSOP (II) (400 mil)
μPD482235G5-xx : 44-pin plastic TSOP (II) (400 mil)

## 7. Example of Stamping

Letter A in the fifth character position in a lot number signifies version A, letter F, version F, and letter E, version E.



# Line Buffer

NEC

MOS INTEGRATED CIRCUIT

# μPD485506

## LINE BUFFER
## 5K-WORD BY 16-BIT/10K-WORD BY 8-BIT

### Description

The μPD485506 is a high speed FIFO (First in First Out) ine buffer. Word organization can be changed either 5 048 words by 16 bits or 10 096 words by 8 bits.

Its CMOS static circuitry provides high speed access and low power consumption.

The μPD485506 can be used for one line delay and time axis conversion in high speed facsimile machines and digital copiers.

Moreover, the μPD485506 can execute read and write operations independently on an asynchronous basis. Thus the μPD485506 is suitable as a buffer for data transfer between units with different transfer rates and as a buffer for the synchronization of multiple input signals.

### Features

- 5 048 words by 16 bits (Word mode) /10 096 words by 8 bits (Byte mode)
- Full static operation; data hold time = infinity
- Suitable for sampling one line of A3 size paper (16 dots/mm)
- Asynchronous read/write operations available
- Variable length delay bits; 21 to 5 048 bits or 10 096 bits (Cycle time: 25 ns)
  19 to 5 048 bits or 10 096 bits (Cycle time: 27 ns)

- Power supply voltage $V_{CC}$ = 5 V ± 10 %
- All input/output TTL compatible
- 3-state output

### Ordering Information

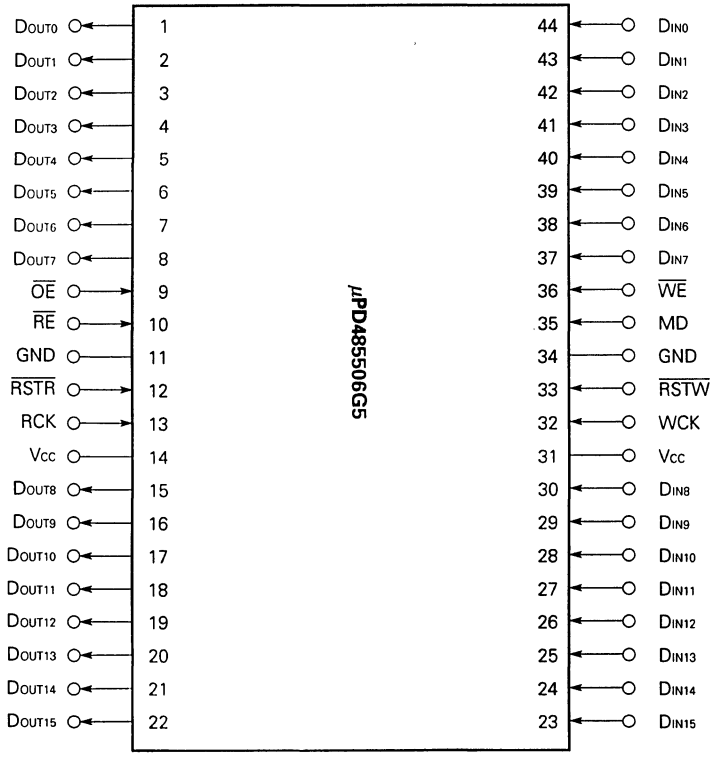
| Part Number | R/W Cycle Time | Package | Quality Grade |
|---|---|---|---|
| μPD485506G5-25 | 25 ns | 44-pin plastic TSOP (II) (400 mil) | Standard |
| μPD485506G5-27 | 27 ns[Note] | | |
| | | | |

**Note** Write cycle time is 25 ns.

Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

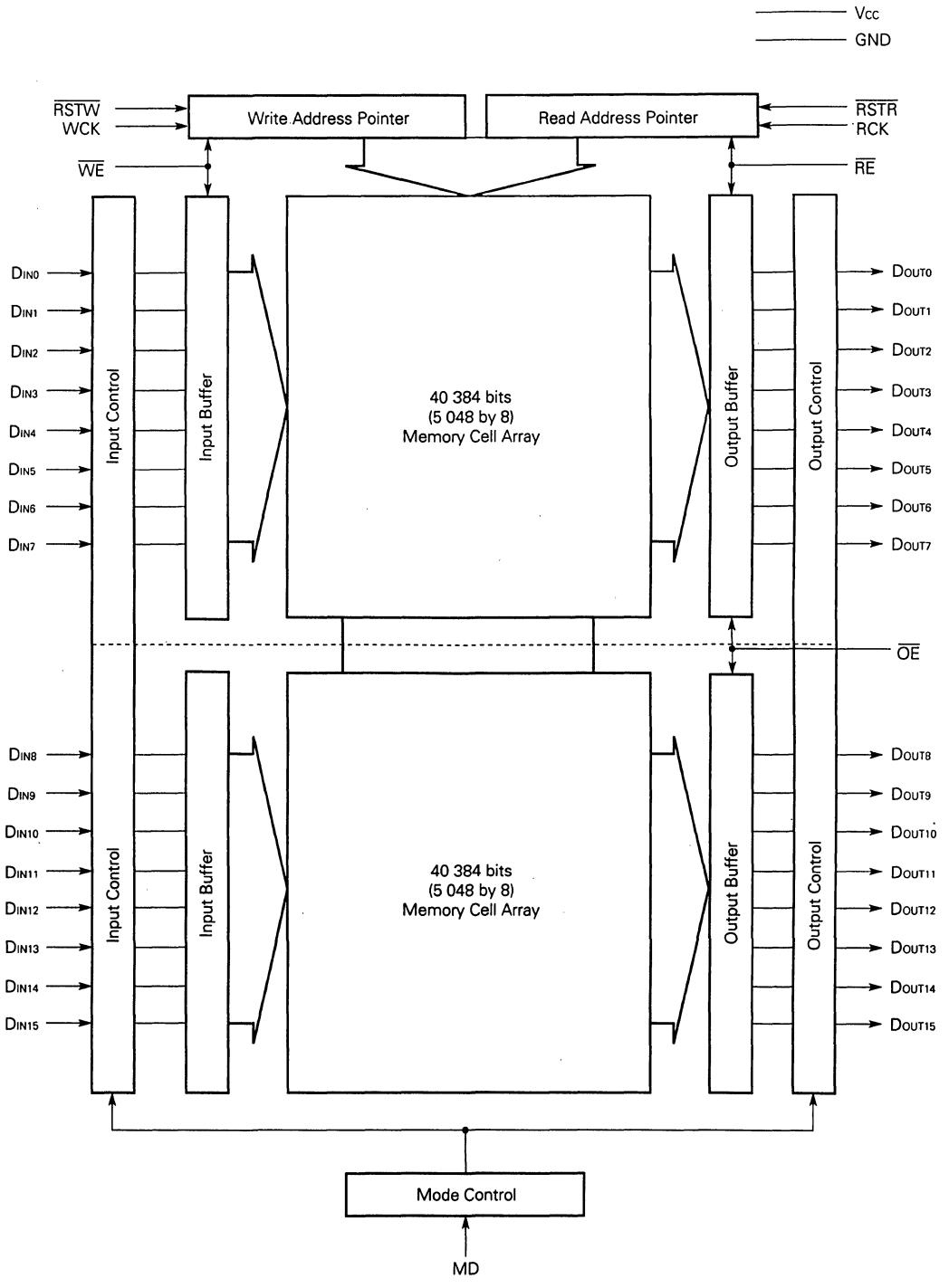
**Pin Configuration** (Marking side)

**44-pin Plastic TSOP (II) (400 mil)**




| | | |
|---|---|---|
| $D_{IN0}$ to $D_{IN15}$ | : | Data Inputs |
| $D_{OUT0}$ to $D_{OUT15}$ | : | Data Outputs |
| WCK | : | Write Clock Input |
| RCK | : | Read Clock Input |
| $\overline{WE}$ | : | Write Enable Input |
| $\overline{RE}$ | : | Read Enable Input |
| $\overline{OE}$ | : | Output Enable Input |
| $\overline{RSTW}$ | : | Reset Write Input |
| $\overline{RSTR}$ | : | Reset Read Input |
| MD | : | Mode Set Input |
| $V_{CC}$ | : | +5 V Power Supply |
| GND | : | Ground |

## Block Diagram



## Pin Function

| Pin | | | I/O | Function |
|---|---|---|---|---|
| Pin Number | Symbol | Pin Name | | |
| 23 - 30<br>37 - 44 | $D_{IN0}$<br>I<br>$D_{IN15}$ | Data Input | In | Write data input pins.<br>The data inputs are strobed by the rising edge of WCK at the end of a cycle and the setup and hold times ($t_{DS}$, $t_{DH}$) are defined at this point. |
| 1 - 8<br>15 - 22 | $D_{OUT0}$<br>I<br>$D_{OUT15}$ | Data Output | Out | Read data output pins.<br>The access time is regulated from the rising edge of RCK at the beginning of a cycle and defined by $t_{AC}$. |
| 33 | $\overline{RSTW}$ | Reset Write Input | In | Reset input pin for the initialization of the write address pointer.<br>The state of $\overline{RSTW}$ is strobed by the rising edge of WCK at the beginning of a cycle and the setup and hold times ($t_{RS}$, $t_{RH}$) are defined. |
| 12 | $\overline{RSTR}$ | Reset Read Input | In | Reset input pin for the initialization of the read address pointer.<br>The state of $\overline{RSTR}$ is strobed by the rising edge of RCK at the beginning of a cycle and the setup and hold times ($t_{RS}$, $t_{RH}$) are defined. |
| 36 | $\overline{WE}$ | Write Enable Input | In | Write operation control signal input pin.<br>When $\overline{WE}$ is in the disable mode ("H" level), the internal write operation is inhibited and the write address pointer stops at the current position. |
| 10 | $\overline{RE}$ | Read Enable Input | In | Read operation control signal input pin.<br>When $\overline{RE}$ is in the disable mode ("H" level), the internal read operation is inhibited and the read address pointer stops at the current position. The data outputs remain valid for that address. |
| 9 | $\overline{OE}$ | Output Enable Input | In | Output operation control signal input pin.<br>When $\overline{OE}$ is in the disable mode ("H" level), the data out is inhibited and the output changes to high impedance. The internal read operation is executed at that time and the read address pointer incremented in synchronization with the read clock. |
| 32 | WCK | Write Clock Input | In | Write clock input pin.<br>When $\overline{WE}$ is enabled ("L" level), the write operation is executed in synchronization with the write clock. The write address pointer is incremented simultaneously. |
| 13 | RCK | Read Clock Input | In | Read clock input pin.<br>When $\overline{RE}$ is enabled ("L" level), the read operation is executed in synchronization with the read clock. The read address pointer is incremented simultaneously. |
| 35 | MD | Mode Set Input | In | Mode set input pin.<br>The level of MD gives the operation mode. When MD is in "L" level, 5 048 words by 16 bits configuration with $D_{IN0}$ - $D_{IN15}$, $D_{OUT0}$ - $D_{OUT15}$ is enabled. When MD is in "H" level, 10 096 words by 8 bits configuration with $D_{IN0}$ - $D_{IN7}$, $D_{OUT0}$ - $D_{OUT7}$ is enabled. |

## Operation Mode

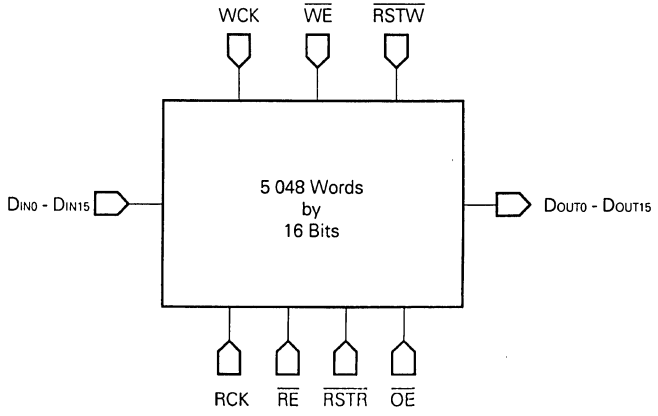
**(1) Mode Set Cycle (5 048 words by 16 bits or 10 096 words by 8 bits organization)**

μPD485506 has a capability of selecting from two operation modes by judging the MD level when $\overline{RSTW}$ or $\overline{RSTR}$ is enabled in the reset cycle.

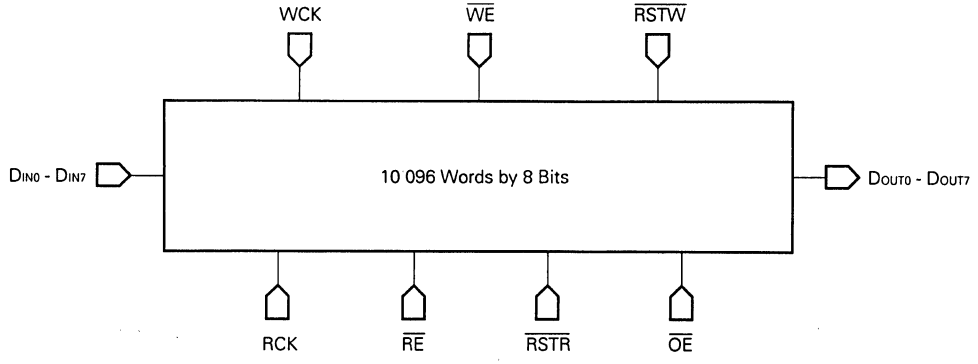
| MD Level | Bit Configuration | Data Inputs/Outputs | Control Signal |
|---|---|---|---|
| "L" | 5 048 words by 16 bits | $D_{IN0}$ - $D_{IN15}$ | WCK, $\overline{WE}$, $\overline{RSTW}$ |
| | | $D_{OUT0}$ - $D_{OUT15}$ | RCK, $\overline{RE}$, $\overline{RSTR}$ |
| "H" | 10 096 words by 8 bits | $D_{IN0}$ - $D_{IN7}$ | WCK, $\overline{WE}$, $\overline{RSTW}$ |
| | | $D_{OUT0}$ - $D_{OUT7}$ | RCK, $\overline{RE}$, $\overline{RSTR}$ |

**Caution Don't change the MD level during a reset cycle.**

**5 048 Words by 16 Bits FIFO**




**10 096 Words by 8 Bits FIFO**




**Remark** Fix $D_{IN8}$ - $D_{IN15}$ to "L" or "H" level in the 10 096 words by 8 bits mode.

**(2) Write Cycle**

When the $\overline{WE}$ input is enabled ("L" level), a write cycle is executed in synchronization with the WCK clock input.

The data inputs are strobed by the rising edge of the clock at the end of a cycle so that read data after a one-line (5 048 bits or 10 096 bits) delay and write data can be processed with the same clock.

When creating a variable length delay line by controlling $\overline{WE}$ or $\overline{RSTW}$, delay bits are as follows.

| Part Number | Cycle Time | Delay Bits |
|---|---|---|
| μPD485506-25 | 25 ns | 21 to 5 048 bits/21 to 10 096 bits |
| μPD485506-27 | | |

Unless inhibited by $\overline{WE}$, the internal write address will automatically wrap around from 5 047 to 0 and begin incrementing again.

**(3) Read Cycle**

When the $\overline{RE}$ input is enabled ("L" level), a read cycle is executed in synchronization with the RCK clock input. When the $\overline{OE}$ input is also enabled ("L" level) at that time, data is output at $t_{AC}$.

When creating a variable length delay line by controlling $\overline{RE}$ or $\overline{RSTR}$, delay bits are as follows.

| Part Number | Cycle Time | Delay Bits |
|---|---|---|
| μPD485506-25 | 25 ns | 21 to 5 048 bits/21 to 10 096 bits |
| μPD485506-27 | 27 ns | 19 to 5 048 bits/19 to 10 096 bits |

When read and write cycles contend for the same line for a time axis conversion, etc., the old data (previous line) may be output for the last 21 bits in the case of 25 ns read cycle time, the last 19 bits in the case of 27 ns read cycle time.

Unless inhibited by $\overline{RE}$, the internal read address will automatically wrap around from 5 047 to 0 and begin incrementing again.

**(4) Write Reset Cycle/Read Reset Cycle**

After power up, the μPD485506 requires the initialization of internal circuits because the read and write address pointers are not defined at that time.

It is necessary to satisfy setup requirements and hold times as measured from the rising edge of WCK and RCK, and then input the $\overline{RSTW}$ and $\overline{RSTR}$ signals to initialize the circuit.

**Remark** Write and read reset cycles can be executed at any time and do not depend on the state of $\overline{RE}$, $\overline{WE}$ or $\overline{OE}$.

**Caution Write and read reset cycles can be executed asynchronously. However, 1/2 cycle and 500 ns is required after a write cycle to read the data written in a cycle.**

## Electrical Specifications

- All voltages are referenced to GND.

### Absolute Maximum Ratings

| Parameter | Symbol | Condition | Rating | Unit |
|---|---|---|---|---|
| Voltage on any pin relative to GND | $V_T$ | | −0.5[Note] to $V_{CC}$ + 0.5 | V |
| Supply voltage | $V_{CC}$ | | −0.5 to +7.0 | V |
| Output current | $I_O$ | | 20 | mA |
| Power dissipation | $P_D$ | | 1 | W |
| Operating temperature | $T_{opt}$ | | 0 to +70 | °C |
| Storage temperature | $T_{stg}$ | | −55 to +125 | °C |

**Note** −3.0 V MIN. (Pulse width = 10 ns)

**Caution Exposing the device to stress above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational section of this specification. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.**

### Recommended Operating Conditions

| Parameter | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Supply voltage | $V_{CC}$ | | 4.5 | 5.0 | 5.5 | V |
| High level input voltage | $V_{IH}$ | | 2.4 | | $V_{CC}$ + 0.5 | V |
| Low level input voltage | $V_{IL}$ | | −0.3[Note] | | +0.8 | V |
| Ambient temperature | $T_a$ | | 0 | | 70 | °C |

**Note** −3.0 V MIN. (Pulse width = 10 ns)

**DC Characteristics** (Recommended Operating Conditions unless otherwise noted)

| Parameter | Symbol | Test Condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Operating current | $I_{CC}$ | | | | 140 | mA |
| Input leakage current | $I_I$ | $V_I$ = 0 to $V_{CC}$, Other Input 0 V | −10 | | +10 | μA |
| Output leakage current | $I_O$ | $V_O$ = 0 to $V_{CC}$, $D_{OUT}$: High Impedance | −10 | | +10 | μA |
| High level output voltage | $V_{OH}$ | $I_{OH}$ = −1 mA | 2.4 | | | V |
| Low level output voltage | $V_{OL}$ | $I_{OL}$ = 2 mA | | | 0.4 | V |

**Capacitance ($T_a$ = +25 °C, f = 1 MHz)**

| Parameter | Symbol | Test Condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input capacitance | $C_I$ | | | | 10 | pF |
| Output capacitance | $C_O$ | | | | 10 | pF |

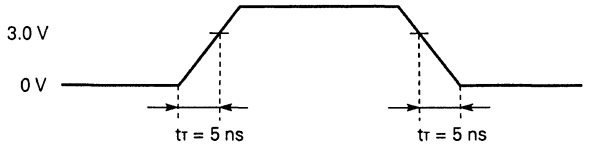
**AC Characteristics (Recommended Operating Conditions unless otherwise noted)** Notes 1, 2, 3

| Parameter | Symbol | μPD485506-25 | | μPD485506-27 | | Unit | Notes |
|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | | |
| Write clock cycle time | $t_{WCK}$ | 25 | | 25 | | ns | |
| Write clock pulse width | $t_{WCW}$ | 9 | | 9 | | ns | |
| Write clock precharge time | $t_{WCP}$ | 9 | | 9 | | ns | |
| Read clock cycle time | $t_{RCK}$ | 25 | | 27 | | ns | |
| Read clock pulse width | $t_{RCW}$ | 9 | | 9 | | ns | |
| Read clock precharge time | $t_{RCP}$ | 9 | | 9 | | ns | |
| Access time | $t_{AC}$ | | 18 | | 18 | ns | |
| Output hold time | $t_{OH}$ | 5 | | 5 | | ns | |
| Output low-impedance time | $t_{LZ}$ | 5 | 18 | 5 | 18 | ns | 4 |
| Output high-impedance time | $t_{HZ}$ | 5 | 18 | 5 | 18 | ns | 4 |
| Input data setup time | $t_{DS}$ | 7 | | 7 | | ns | |
| Input data hold time | $t_{DH}$ | 3 | | 3 | | ns | |
| MD Set setup time | $t_{MS}$ | 20 | | 20 | | ns | |
| MD Set hold time | $t_{MH}$ | 10 | | 10 | | ns | |
| MD Set time | $t_{MD}$ | 0 | | 0 | | ns | 5 |
| Output low-impedance time (Mode change) | $t_{LZM}$ | 5 | 18 | 5 | 18 | ns | 4 |
| Output high-impedance time (Mode change) | $t_{HZM}$ | 5 | 18 | 5 | 18 | ns | 4 |
| $\overline{RSTW}$/$\overline{RSTR}$ Setup time | $t_{RS}$ | 7 | | 7 | | ns | 6 |
| $\overline{RSTW}$/$\overline{RSTR}$ Hold time | $t_{RH}$ | 3 | | 3 | | ns | 6 |
| $\overline{RSTW}$/$\overline{RSTR}$ Deselected time (1) | $t_{RN1}$ | 3 | | 3 | | ns | 7 |
| $\overline{RSTW}$/$\overline{RSTR}$ Deselected time (2) | $t_{RN2}$ | 7 | | 7 | | ns | 7 |
| $\overline{WE}$ Setup time | $t_{WES}$ | 7 | | 7 | | ns | 8 |
| $\overline{WE}$ Hold time | $t_{WEH}$ | 3 | | 3 | | ns | 8 |
| $\overline{WE}$ Deselected time (1) | $t_{WEN1}$ | 3 | | 3 | | ns | 9 |
| $\overline{WE}$ Deselected time (2) | $t_{WEN2}$ | 7 | | 7 | | ns | 9 |
| $\overline{RE}$ Setup time | $t_{RES}$ | 7 | | 7 | | ns | 10 |
| $\overline{RE}$ Hold time | $t_{REH}$ | 3 | | 3 | | ns | 10 |
| $\overline{RE}$ Deselected time (1) | $t_{REN1}$ | 3 | | 3 | | ns | 11 |
| $\overline{RE}$ Deselected time (2) | $t_{REN2}$ | 7 | | 7 | | ns | 11 |
| $\overline{OE}$ Setup time | $t_{OES}$ | 7 | | 7 | | ns | 10 |
| $\overline{OE}$ Hold time | $t_{OEH}$ | 3 | | 3 | | ns | 10 |
| $\overline{OE}$ Deselected time (1) | $t_{OEN1}$ | 3 | | 3 | | ns | 11 |
| $\overline{OE}$ Deselected time (2) | $t_{OEN2}$ | 7 | | 7 | | ns | 11 |
| $\overline{WE}$ Disable time | $t_{WEW}$ | 0 | | 0 | | ms | |
| $\overline{RE}$ Disable time | $t_{REW}$ | 0 | | 0 | | ms | |
| $\overline{OE}$ Disable time | $t_{OEW}$ | 0 | | 0 | | ms | |
| Write reset time | $t_{RSTW}$ | 0 | | 0 | | ms | |
| Read reset time | $t_{RSTR}$ | 0 | | 0 | | ms | |
| Transition time | $t_T$ | 3 | 35 | 3 | 35 | ns | |

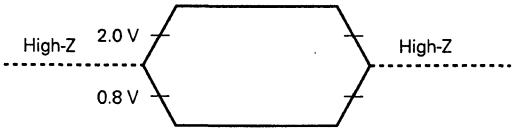
**Notes 1.** AC measurements assume $t_T$ = 5 ns.

**2.** AC Characteristics test condition

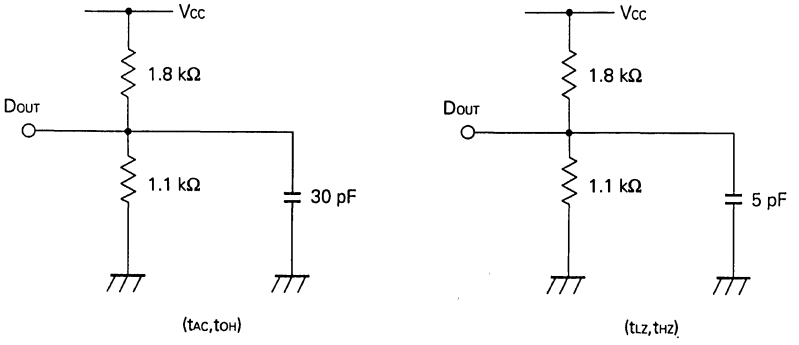
Input Timing Specification




Output Timing Specification




Output Loads for Timing




**3.** Input timing reference levels = 1.5 V.

**4.** $t_{LZ}$, $t_{HZ}$, $t_{LZM}$ and $t_{HZM}$ are measured at ±200 mV from the steady state voltage. Under any conditions, $t_{LZ} \geq t_{HZ}$ and $t_{LZM} \geq t_{HZM}$.

**5.** Mode set signal (MD) must be input synchronously with write reset signal ($t_{RSTW}$ period) or read reset signal ($t_{RSTR}$ period). Under this condition, $t_{RSTW} = t_{MD}$ ($t_{RSTR} = t_{MD}$).

**6.** If either $t_{RS}$ or $t_{RH}$ is less than the specified value, reset operations are not guaranteed.

**7.** If either $t_{RN1}$ or $t_{RN2}$ is less than the specified value, reset operations may extend to cycles preceding or following the period of reset operations.

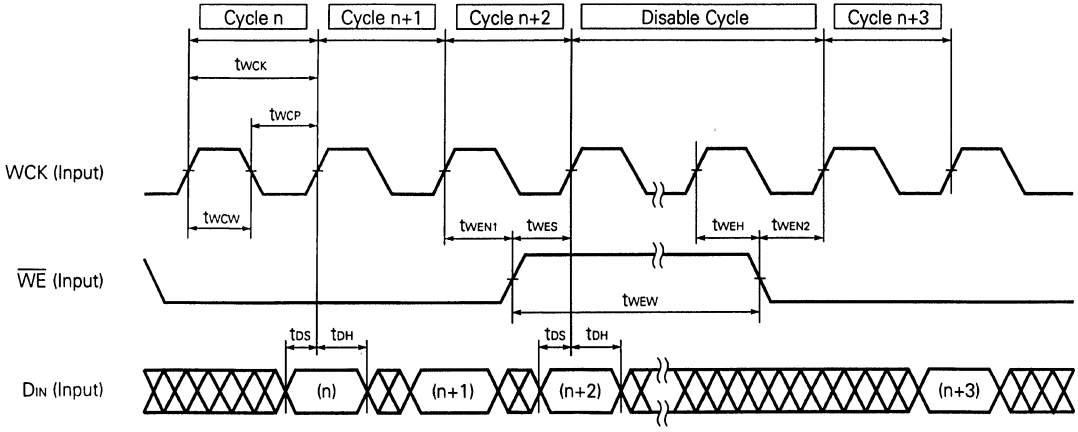
**8.** If either $t_{WES}$ or $t_{WEH}$ is less than the specified value, write disable operations are not guaranteed.

**9.** If either $t_{WEN1}$ or $t_{WEN2}$ is less than the specified value, internal write disable operations may extend to cycles preceding or following the period of write disable operations.

**10.** If either $t_{RES}$ or $t_{REH}$, $t_{OES}$ or $t_{OEH}$ is less than the specified value, read disable operations are not guaranteed.

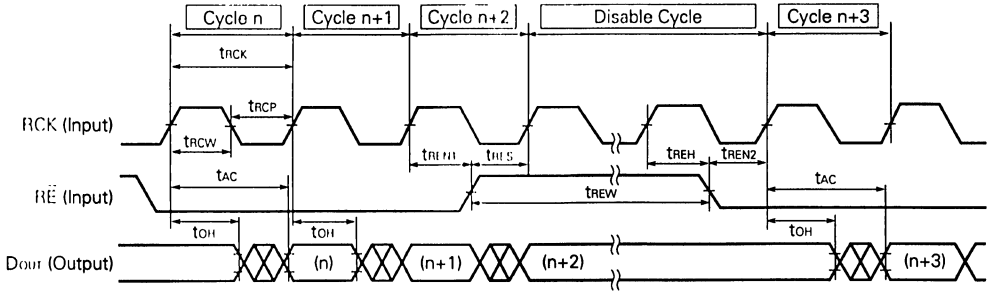
**11.** If either $t_{REN1}$ or $t_{REN2}$, $t_{OEN1}$ or $t_{OEN2}$ is less than the specified value, internal read disable operations may extend to cycles preceding or following the period of read disable operations.

**Write Cycle**



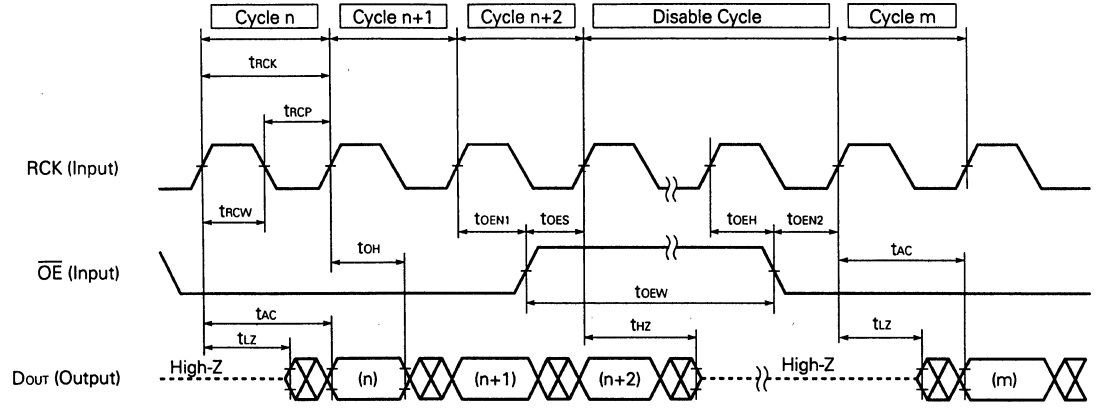

**Remark** $\overline{RSTW}$ = "H" level

**Read Cycle ($\overline{RE}$ Control)**




**Remark** $\overline{OE}$ = "L" level, $\overline{RSTR}$ = "H" level

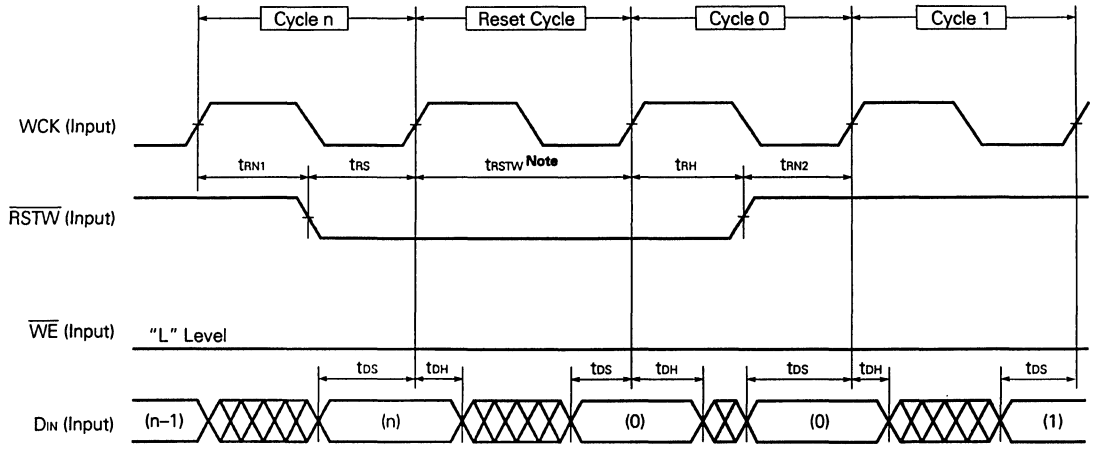
**Read Cycle ($\overline{OE}$ Control)**





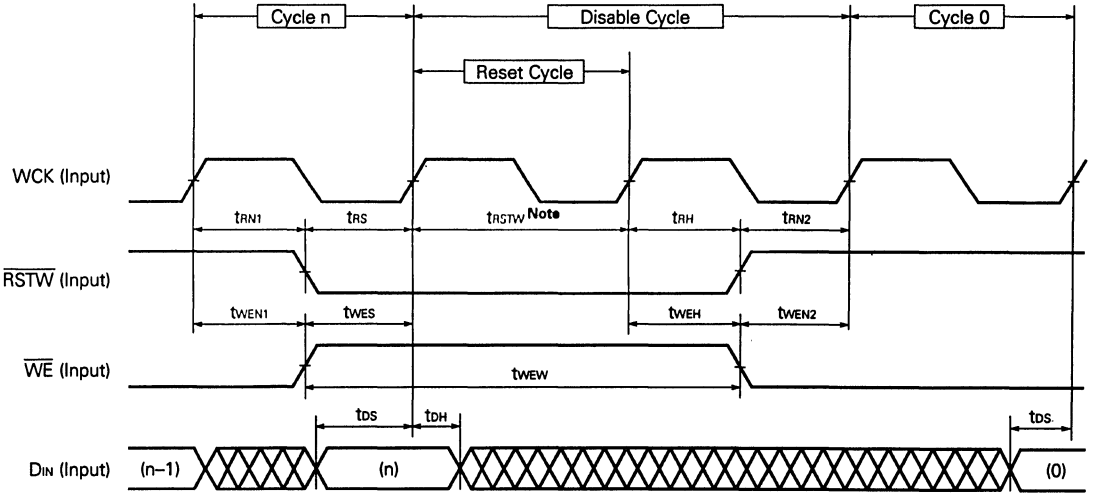

**Remark** $\overline{RE}$ = "L" level, $\overline{RSTR}$ = "H" level

**Write Reset Cycle ($\overline{WE}$ Controlled 1)**



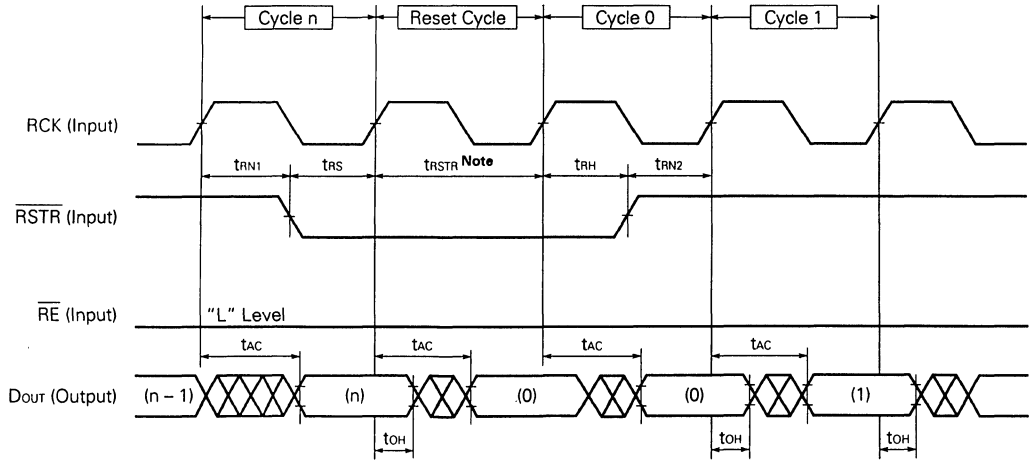

**Note** In write reset cycle, reset operation is executed even without a reset cycle ($t_{RSTW}$). WCK can be input any number of times in a reset cycle.

**Write Reset Cycle ($\overline{WE}$ Controlled 2)**




**Note** In write reset cycle, reset operation is executed even without a reset cycle ($t_{RSTW}$). WCK can be input any number of times in a reset cycle.

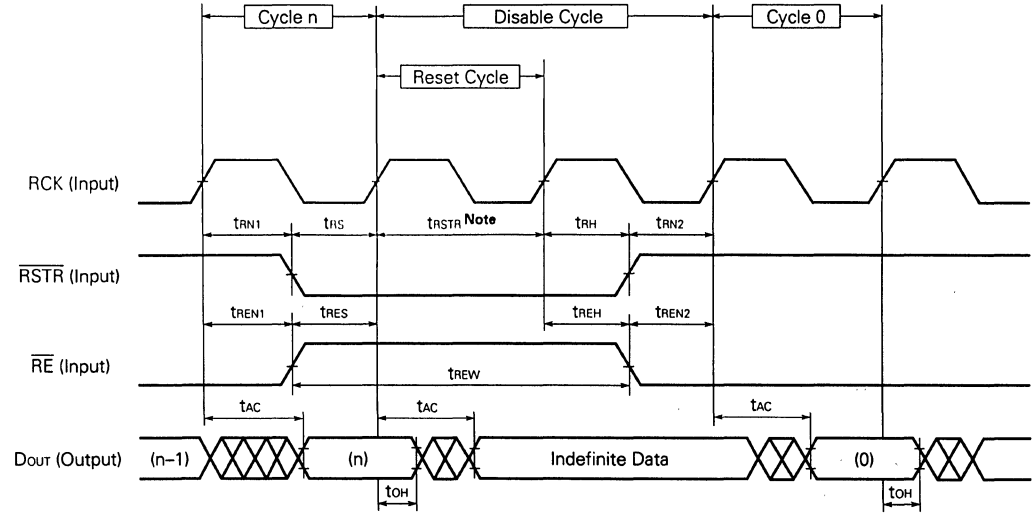
**Read Reset Cycle ($\overline{RE}$ Controlled 1)**




**Note** In read reset cycle, reset operation is executed even without a reset cycle ($t_{RSTR}$). RCK can be input any number of times in a reset cycle.

**Remark** $\overline{OE}$ = "L" level

**Read Reset Cycle ($\overline{RE}$ Controlled 2)**




**Note** In read reset cycle, reset operation is executed even without a reset cycle ($t_{RSTR}$). RCK can be input any number of times in a reset cycle.

**Remark** $\overline{OE}$ = "L" level

## Application

- **1 H Delay Line**

  μPD485506 easily allows a 1 H (5 048 bits/10 096 bits) delay line (see Figure 1, 2 below).
  It is also possible to change the number of delay bits depending on the cycle time as follows.

| Part Number | Cycle Time | Delay Bits |
|---|---|---|
| μPD485506-25 | 25 ns | 21 to 5 048 bits/21 to 10 096 bits |
| μPD485506-27 | 27 ns | 19 to 5 048 bits/19 to 10 096 bits |

To change the number of delay bits, you can choose the one of the following methods.

**Adjustments of the number of delay bits**

(1) Reset the cycle proportionate to the delay length (Figure 3).
(2) Shift the input timing of write reset ($\overline{RSTW}$) and read reset signals ($\overline{RSTR}$) according to the delay length (Figure 4).
(3) Shift the address by disabling $\overline{WE}$ or $\overline{RE}$ for the period proportionate to the delay length (Figure 5).

**Caution After power up, the μPD485506 requires the initialization of internal circuits because the read and write address pointers are not defined at that time.**

**Fig. 1 1 H Delay Line Circuit**

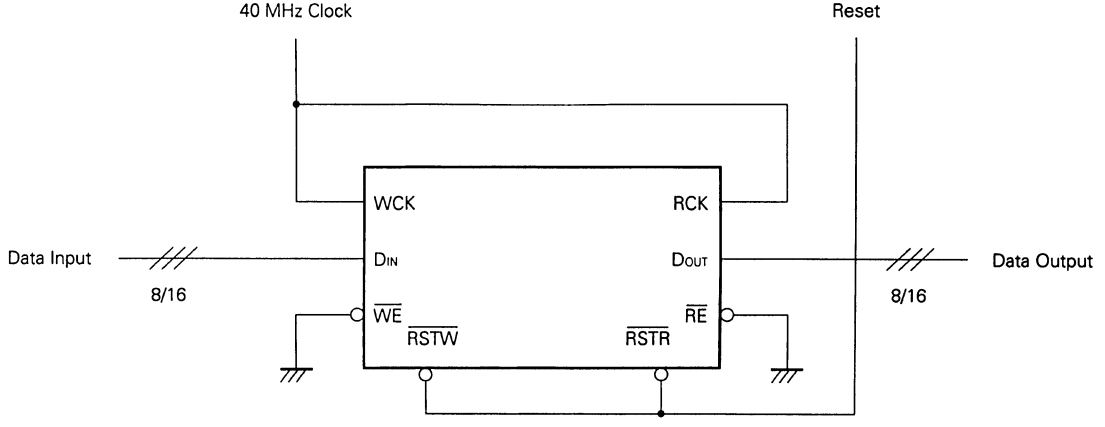
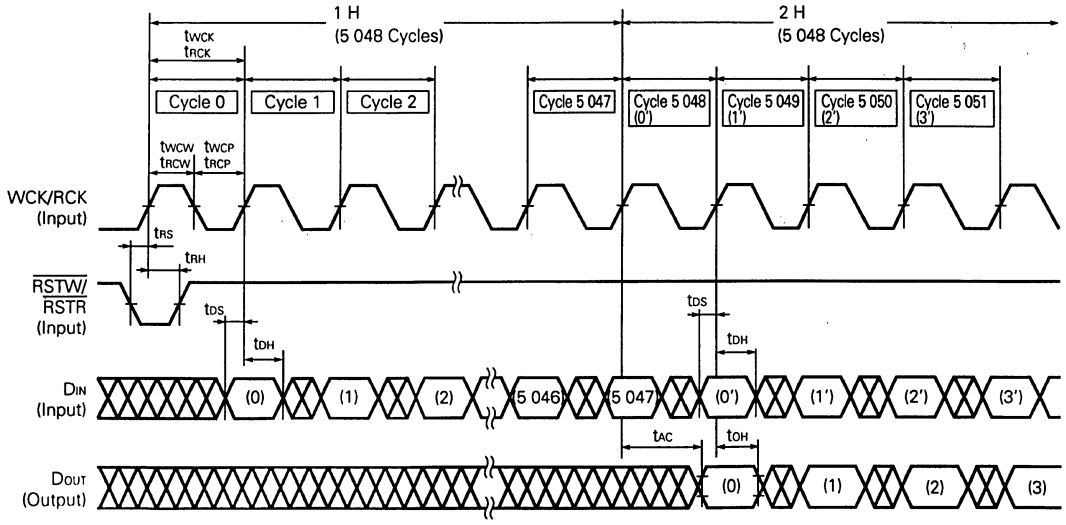
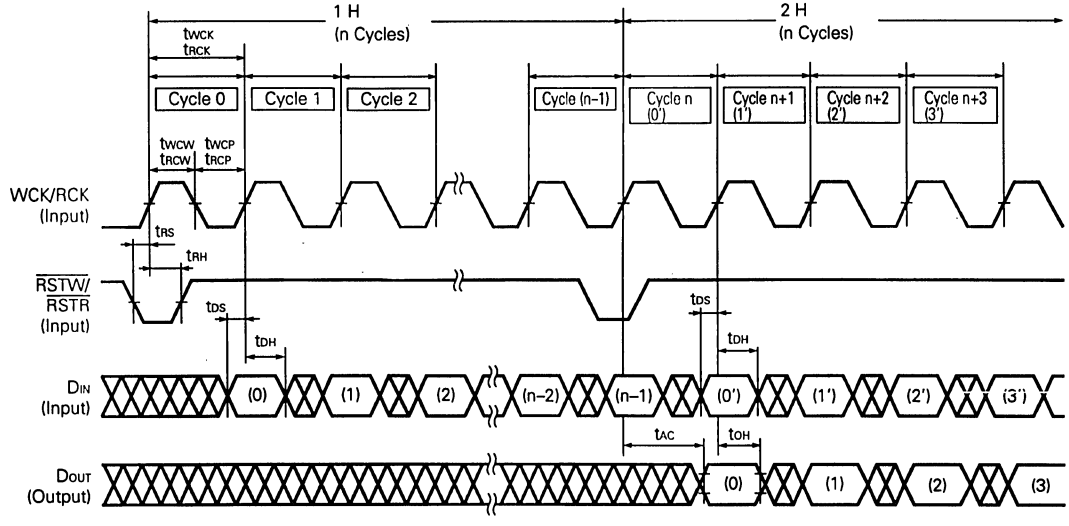


**Fig. 2 1 H Delay Line Timing**



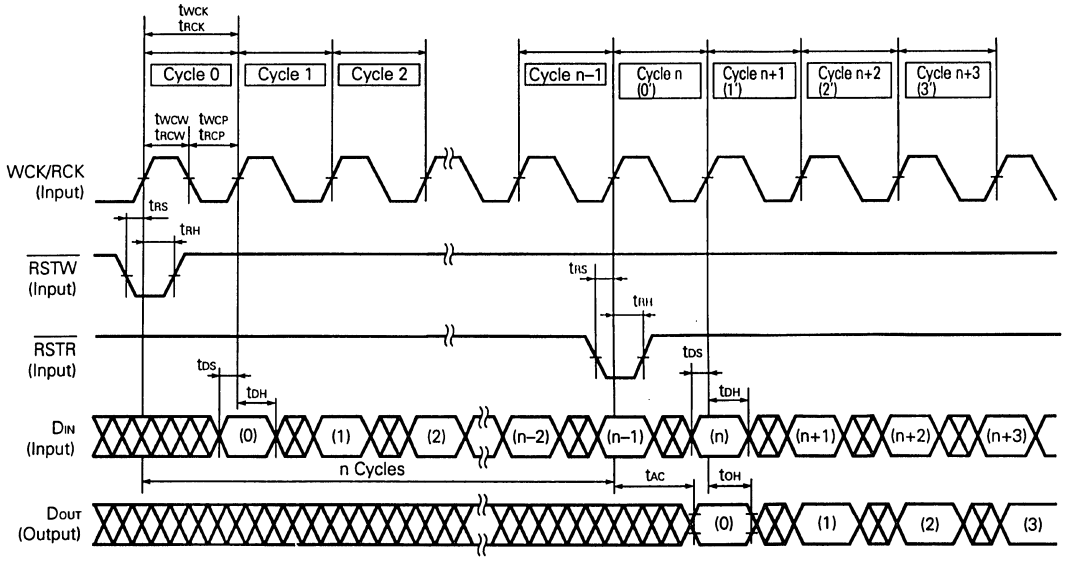

**Remark** $\overline{RE}$, $\overline{WE}$, $\overline{OE}$ = "L" level

**Fig. 3 n-Bit Delay Line Timing (1)**



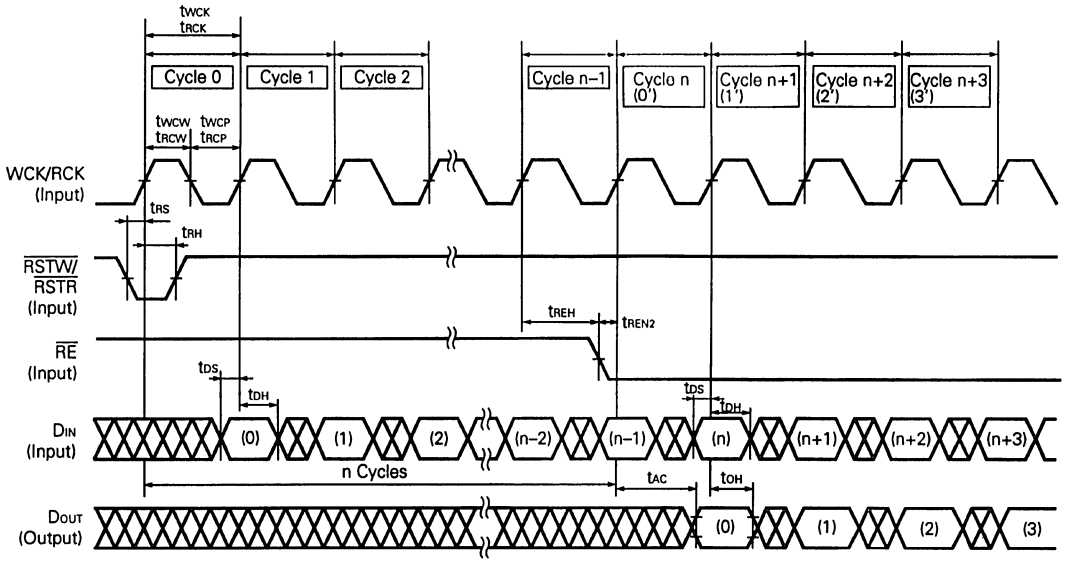

**Remark** $\overline{RE}$, $\overline{WE}$, $\overline{OE}$ = "L" level

**Fig. 4 n-Bit Delay Line Timing (2)**



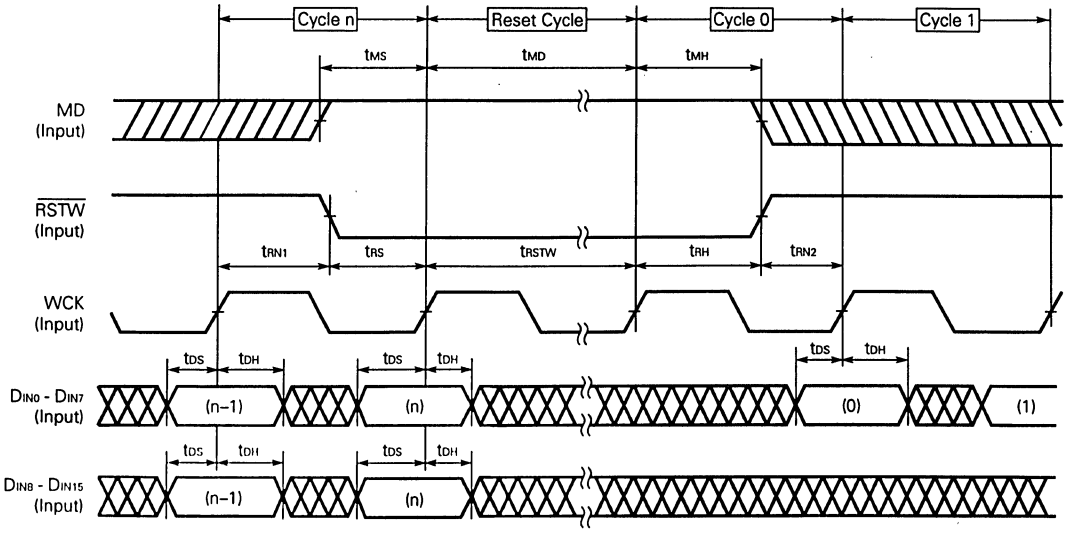

**Remark** $\overline{RE}$, $\overline{WE}$, $\overline{OE}$ = "L" level

**Fig. 5 n-Bit Delay Line Timing (3)**



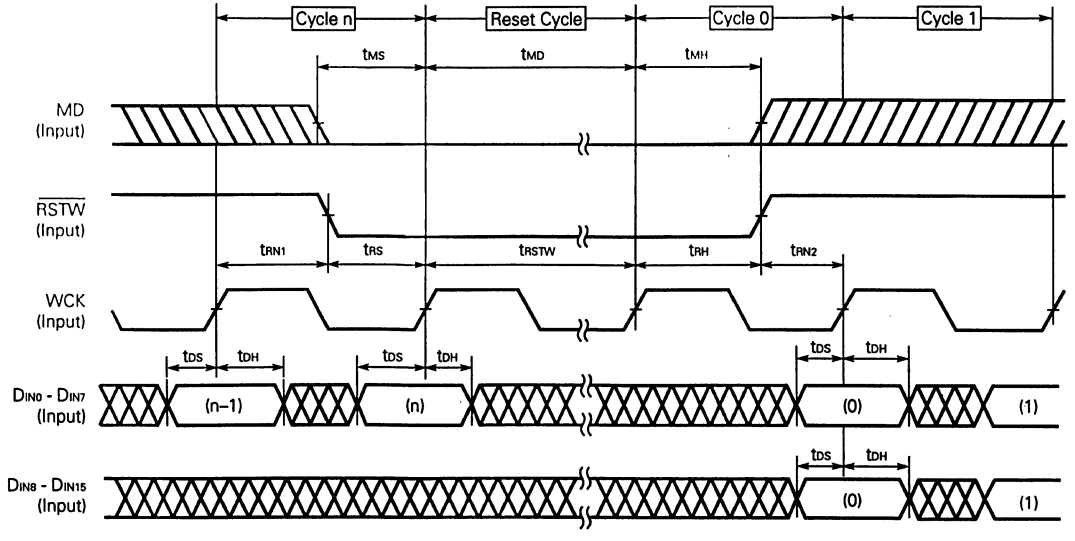

**Remark** $\overline{WE}$, $\overline{OE}$ = "L" level

**Fig. 6 Mode Set Cycle (Write) (1)**



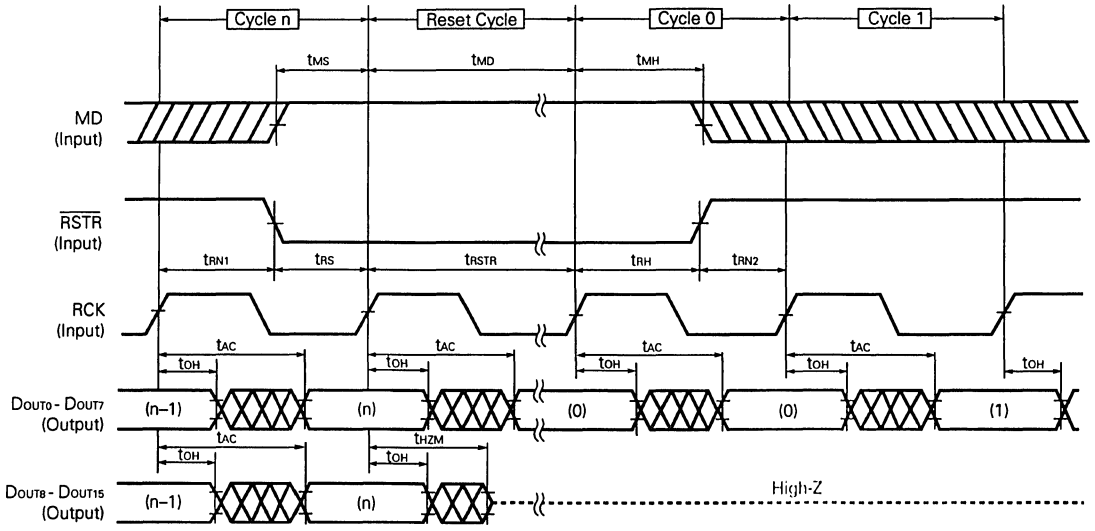

**Remark** $\overline{WE}$ = "L" level

**Fig. 7 Mode Set Cycle (Write) (2)**



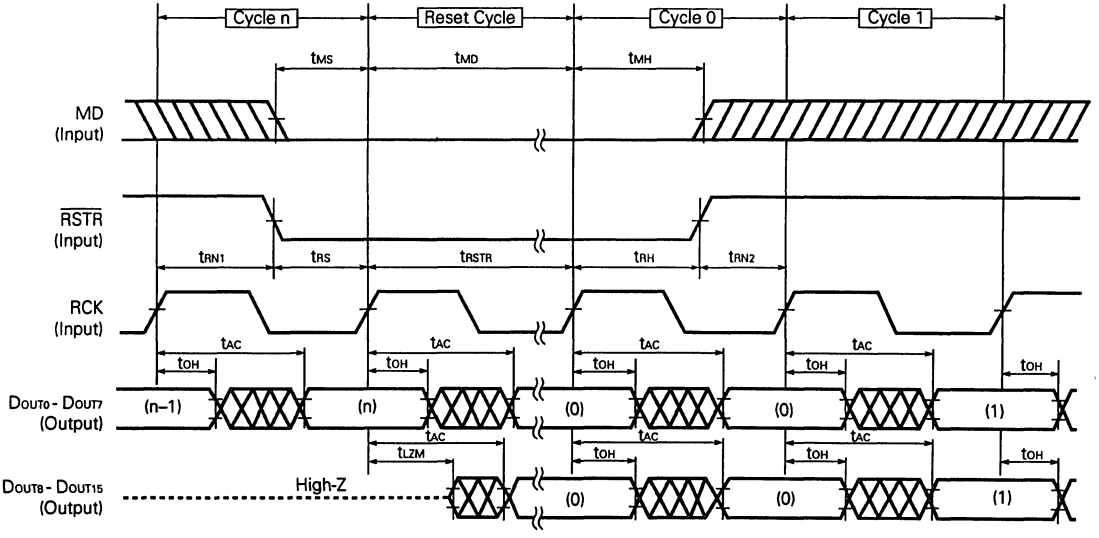

**Remark** $\overline{WE}$ = "L" level

**Fig. 8 Mode Set Cycle (Read) (1)**




**Remark** $\overline{RE}$, $\overline{OE}$ = "L" level

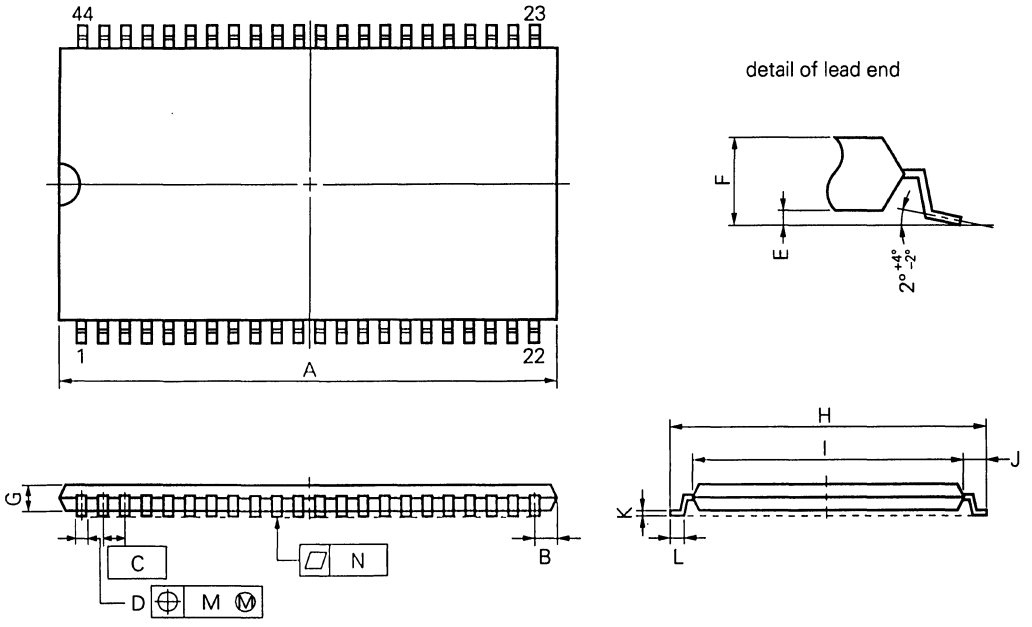
**Fig. 9 Mode Set Cycle (Read) (2)**




**Remark** $\overline{RE}$, $\overline{OE}$ = "L" level

## Package Drawing

### 44 PIN PLASTIC TSOP(II) (400 mil)




S44G5-80-7JF1-1

**NOTE**

Each lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 18.81 MAX. | 0.741 MAX. |
| B | 1.0 MAX. | 0.040 MAX. |
| C | 0.8 (T.P.) | 0.031 (T.P.) |
| D | 0.30±0.10 | $0.012^{+0.004}_{-0.005}$ |
| E | 0.05±0.05 | 0.002±0.002 |
| F | 1.1 MAX. | 0.044 MAX. |
| G | 0.97 | 0.038 |
| H | 11.76±0.2 | 0.463±0.008 |
| I | 10.16±0.1 | 0.400±0.004 |
| J | 0.8±0.2 | $0.031^{+0.009}_{-0.008}$ |
| K | $0.125^{+0.10}_{-0.05}$ | $0.005^{+0.004}_{-0.002}$ |
| L | 0.5±0.1 | $0.020^{+0.004}_{-0.005}$ |
| M | 0.13 | 0.005 |
| N | 0.10 | 0.004 |

## Recommended Soldering Conditions

Please consult with our sales offices for soldering conditions of the μPD485506.

**Type of Surface Mount Device**

**μPD485506G: 44-pin Plastic TSOP (II) (400 mil)**

[MEMO]

NEC

# MOS INTEGRATED CIRCUIT
# μPD485505

## LINE BUFFER
## 5K-WORD BY 8-BIT

9.15 4/98

Vendors: MARSHALL Reptron

### Description

The μPD485505 is a 5 048 words by 8 bits high speed FIFO (First In First Out) line buffer. Its CMOS static circuitry provides high speed access and low power consumption.

The μPD485505 can be used for one line delay and time axis conversion in high speed facsimile machines and digital copiers.

Moreover, the μPD485505 can execute read and write operations independently on an asynchronous basis. Thus the μPD485505 is suitable as a buffer for data transfer between units with different transfer rates and as a buffer for the synchronization of multiple input signals.

### Features

- 5 048 words by 8 bits
- Full static operation; data hold time = infinity
- Suitable for sampling one line of A3 size paper (16 dots/mm)
- Asynchronous read/write operations available
- Variable length delay bits; 21 to 5 048 bits (Cycle time: 25 ns)
  19 to 5 048 bits (Cycle time: 27 ns)

- Power supply voltage $V_{CC}$ = 5 V ± 10 %
- All input/output TTL compatible
- 3-state output

### Ordering Information

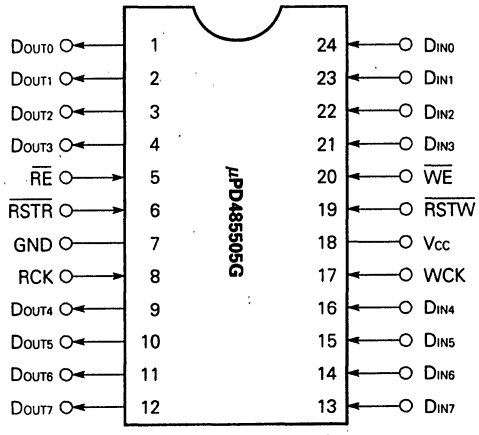
| Part Number | R/W Cycle Time | Package | Quality Grade |
|---|---|---|---|
| μPD485505G-25 | 25 ns | 24-pin plastic SOP (450 mil) | Standard |
| μPD485505G-27 | 27 ns Note | | |
| | | | |

**Note** Write cycle time is 25 ns.

Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

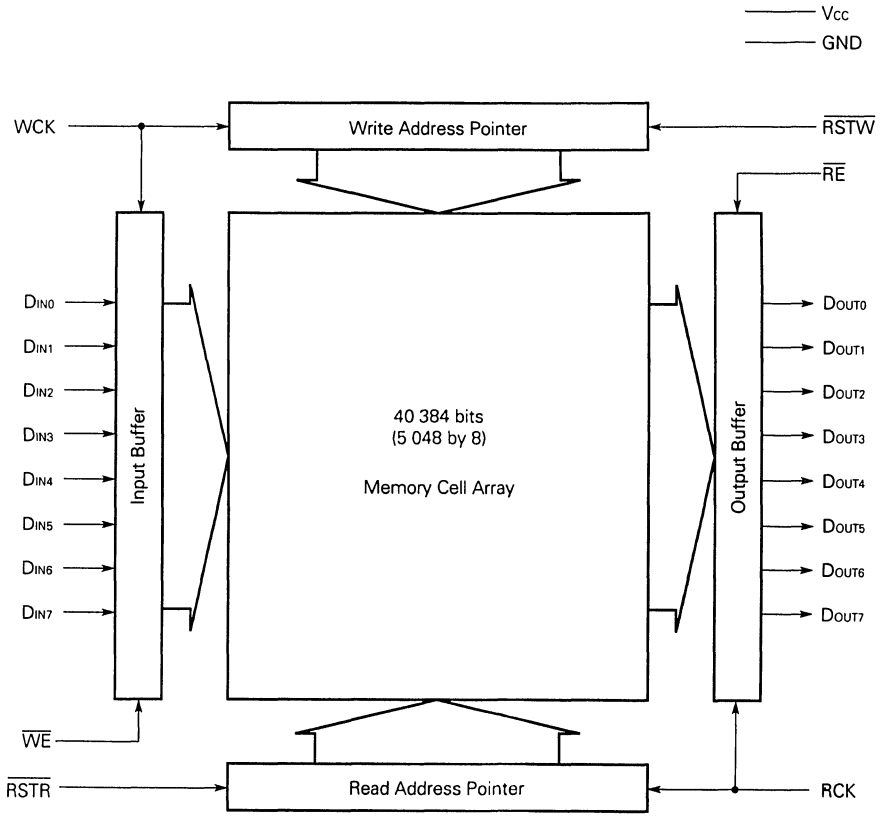
**Pin Configuration** (Marking side)

**24-pin Plastic SOP (450 mil)**




| | | |
|---|---|---|
| $D_{IN0}$ - $D_{IN7}$ | : | Data Inputs |
| $D_{OUT0}$ - $D_{OUT7}$ | : | Data Outputs |
| WCK | : | Write Clock Input |
| RCK | : | Read Clock Input |
| $\overline{WE}$ | : | Write Enable Input |
| $\overline{RE}$ | : | Read Enable Input |
| $\overline{RSTW}$ | : | Reset Write Input |
| $\overline{RSTR}$ | : | Reset Read Input |
| $V_{CC}$ | : | +5 V Power Supply |
| GND | : | Ground |

## Block Diagram



## Pin Function

| Pin | | | I/O | Function |
|---|---|---|---|---|
| Pin Number | Symbol | Pin Name | | |
| 13 - 16<br>21 - 24 | $D_{IN0}$<br>I<br>$D_{IN7}$ | Data Input | In | Write data input pins.<br>The data inputs are strobed by the rising edge of WCK at the end of a cycle and the setup and hold times ($t_{DS}$, $t_{DH}$) are defined at this point. |
| 1 - 4<br>9 - 12 | $D_{OUT0}$<br>I<br>$D_{OUT7}$ | Data Output | Out | Read data output pins.<br>The access time is regulated from the rising edge of RCK at the beginning of a cycle and defined by $t_{AC}$. |
| 19 | $\overline{RSTW}$ | Reset Write Input | In | Reset input pin for the initialization of the write address pointer.<br>The state of $\overline{RSTW}$ is strobed by the rising edge of WCK at the beginning of a cycle and the setup and hold times ($t_{RS}$, $t_{RH}$) are defined. |
| 6 | $\overline{RSTR}$ | Reset Read Input | In | Reset input pin for the initialization of the read address pointer.<br>The state of $\overline{RSTR}$ is strobed by the rising edge of RCK at the beginning of a cycle and the setup and hold times ($t_{RS}$, $t_{RH}$) are defined. |
| 20 | $\overline{WE}$ | Write Enable Input | In | Write operation control signal input pin.<br>When $\overline{WE}$ is in the disable mode ("H" level), the internal write operation is inhibited and the write address pointer stops at the current position. |
| 5 | $\overline{RE}$ | Read Enable Input | In | Read operation control signal input pin.<br>When $\overline{RE}$ is in the disable mode ("H" level), the internal read operation is inhibited and the read address pointer stops at the current position. The output changes to high impedance. |
| 17 | WCK | Write Clock Input | In | Write clock input pin.<br>When $\overline{WE}$ is enabled ("L" level), the write operation is executed in synchronization with the write clock. The write address pointer is incremented simultaneously. |
| 8 | RCK | Read Clock Input | In | Read clock input pin.<br>When $\overline{RE}$ is enabled ("L" level), the read operation is executed in synchronization with the read clock. The read address pointer is incremented simultaneously. |

## Operation Mode

**(1) Write Cycle**

When the $\overline{WE}$ input is enabled ("L" level), a write cycle is executed in synchronization with the WCK clock input.

The data inputs are strobed by the rising edge of the clock at the end of a cycle so that read data after a one-line (5 048 bits) delay and write data can be processed with the same clock.

When creating a variable length delay line by controlling $\overline{WE}$ or $\overline{RSTW}$, delay bits are as follows.

| Part Number | Cycle Time | Delay Bits |
|---|---|---|
| μPD485505-25 | 25 ns | 21 to 5 048 bits |
| μPD485505-27 | | |

Unless inhibited by $\overline{WE}$, the internal write address will automatically wrap around from 5 047 to 0 and begin incrementing again.

**(2) Read Cycle**

When the $\overline{RE}$ input is enabled ("L" level), a read cycle is executed in synchronization with the RCK clock input and data is output at $t_{AC}$.

When creating a variable length delay line by controlling $\overline{RE}$ or $\overline{RSTR}$, delay bits are as follows.

| Part Number | Cycle Time | Delay Bits |
|---|---|---|
| μPD485505-25 | 25 ns | 21 to 5 048 bits |
| μPD485505-27 | 27 ns | 19 to 5 048 bits |

When read and write cycles contend for the same line for a time axis conversion, etc., the old data (previous line) may be output for the last 21 bits in the case of 25 ns read cycle time, the last 19 bits in the case of 27 ns read cycle time.

Unless inhibited by $\overline{RE}$, the internal read address will automatically wrap around from 5 047 to 0 and begin incrementing again.

**(3) Write Reset Cycle/Read Reset Cycle**

After power up, the μPD485505 requires the initialization of internal circuits because the read and write address pointers are not defined at that time.

It is necessary to satisfy setup requirements and hold times as measured from the rising edge of WCK and RCK, and then input the $\overline{RSTW}$ and $\overline{RSTR}$ signals to initialize the circuit.

**Remark** Write and read reset cycles can be executed at any time and do not depend on the state of $\overline{RE}$ or $\overline{WE}$.

**Caution Write and read reset cycles can be executed asynchronously. However, 1/2 cycle and 500 ns is required after a write cycle to read the data written in a cycle.**

## Electrical Specifications

- All voltages are referenced to GND.

### Absolute Maximum Ratings

| Parameter | Symbol | Condition | Rating | Unit |
|---|---|---|---|---|
| Voltage on any pin relative to GND | $V_T$ | | −0.5[Note] to $V_{CC}$ + 0.5 | V |
| Supply voltage | $V_{CC}$ | | −0.5 to +7.0 | V |
| Output current | $I_O$ | | 20 | mA |
| Operating temperature | $T_{opt}$ | | 0 to +70 | °C |
| Storage temperature | $T_{stg}$ | | −55 to +125 | °C |

**Note** −3.0 V MIN. (Pulse width = 10 ns)

**Caution Exposing the device to stress above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational section of this specification. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.**

### Recommended Operating Conditions

| Parameter | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Supply voltage | $V_{CC}$ | | 4.5 | 5.0 | 5.5 | V |
| High level input voltage | $V_{IH}$ | | 2.4 | | $V_{CC}$ + 0.5 | V |
| Low level input voltage | $V_{IL}$ | | −0.3[Note] | | +0.8 | V |
| Ambient temperature | $T_a$ | | 0 | | 70 | °C |

**Note** −3.0 V MIN. (Pulse width = 10 ns)

### DC Characteristics (Recommended Operating Conditions unless otherwise noted)

| Parameter | Symbol | Test Condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Operating current | $I_{CC}$ | | | | 80 | mA |
| Input leakage current | $I_I$ | $V_I$ = 0 to $V_{CC}$, Other Input 0 V | −10 | | +10 | μA |
| Output leakage current | $I_O$ | $V_O$ = 0 to $V_{CC}$, $D_{OUT}$: High Impedance | −10 | | +10 | μA |
| High level output voltage | $V_{OH}$ | $I_{OH}$ = −1 mA | 2.4 | | | V |
| Low level output voltage | $V_{OL}$ | $I_{OL}$ = 2 mA | | | 0.4 | V |

### Capacitance ($T_a$ = +25 °C, f = 1 MHz)

| Parameter | Symbol | Test Condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input capacitance | $C_I$ | | | | 10 | pF |
| Output capacitance | $C_O$ | | | | 10 | pF |

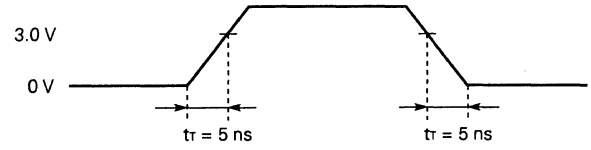
**AC Characteristics (Recommended Operating Conditions unless otherwise noted)[Notes 1, 2, 3]**

| Parameter | Symbol | μPD485505-25 | | μPD485505-27 | | Unit | Notes |
|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | | |
| Write clock cycle time | $t_{WCK}$ | 25 | | 25 | | ns | |
| Write clock pulse width | $t_{WCW}$ | 9 | | 9 | | ns | |
| Write clock precharge time | $t_{WCP}$ | 9 | | 9 | | ns | |
| Read clock cycle time | $t_{RCK}$ | 25 | | 27 | | ns | |
| Read clock pulse width | $t_{RCW}$ | 9 | | 9 | | ns | |
| Read clock precharge time | $t_{RCP}$ | 9 | | 9 | | ns | |
| Access time | $t_{AC}$ | | 18 | | 18 | ns | |
| Output hold time | $t_{OH}$ | 5 | | 5 | | ns | |
| Output low-impedance time | $t_{LZ}$ | 5 | 18 | 5 | 18 | ns | 4 |
| Output high-impedance time | $t_{HZ}$ | 5 | 18 | 5 | 18 | ns | 4 |
| Input data setup time | $t_{DS}$ | 7 | | 7 | | ns | |
| Input data hold time | $t_{DH}$ | 3 | | 3 | | ns | |
| $\overline{RSTW}/\overline{RSTR}$ Setup time | $t_{RS}$ | 7 | | 7 | | ns | 5 |
| $\overline{RSTW}/\overline{RSTR}$ Hold time | $t_{RH}$ | 3 | | 3 | | ns | 5 |
| $\overline{RSTW}/\overline{RSTR}$ Deselected time (1) | $t_{RN1}$ | 3 | | 3 | | ns | 6 |
| $\overline{RSTW}/\overline{RSTR}$ Deselected time (2) | $t_{RN2}$ | 7 | | 7 | | ns | 6 |
| $\overline{WE}$ Setup time | $f_{WES}$ | 7 | | 7 | | ns | 7 |
| $\overline{WE}$ Hold time | $t_{WEH}$ | 3 | | 3 | | ns | 7 |
| $\overline{WE}$ Deselected time (1) | $t_{WEN1}$ | 3 | | 3 | | ns | 8 |
| $\overline{WE}$ Deselected time (2) | $t_{WEN2}$ | 7 | | 7 | | ns | 8 |
| $\overline{RE}$ Setup time | $t_{RES}$ | 7 | | 7 | | ns | 9 |
| $\overline{RE}$ Hold time | $t_{REH}$ | 3 | | 3 | | ns | 9 |
| $\overline{RE}$ Deselected time (1) | $t_{REN1}$ | 3 | | 3 | | ns | 10 |
| $\overline{RE}$ Deselected time (2) | $t_{REN2}$ | 7 | | 7 | | ns | 10 |
| $\overline{WE}$ Disable time | $t_{WEW}$ | 0 | | 0 | | ms | |
| $\overline{RE}$ Disable time | $t_{REW}$ | 0 | | 0 | | ms | |
| Write reset time | $t_{RSTW}$ | 0 | | 0 | | ms | |
| Read reset time | $t_{RSTR}$ | 0 | | 0 | | ms | |
| Transition time | $t_T$ | 3 | 35 | 3 | 35 | ns | |

**Notes 1.** AC measurements assume $t_T$ = 5 ns.

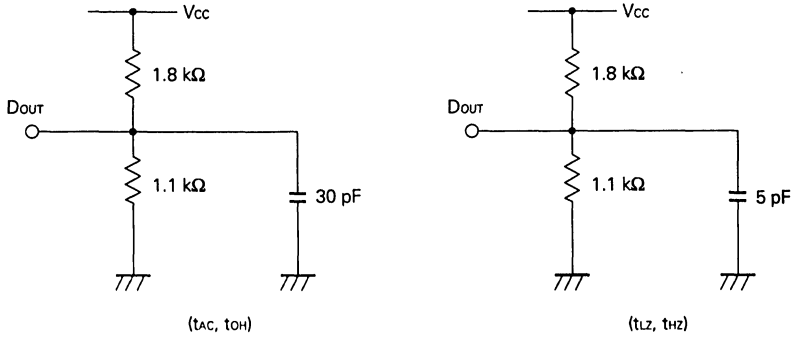
**2.** AC Characteristics test condition

Input Timing Specification




Output Timing Specification


Output Loads for Timing




**3.** Input timing reference levels = 1.5 V.

**4.** $t_{LZ}$ and $t_{HZ}$ are measured at ±200 mV from the steady state voltage. Under any conditions, $t_{LZ} \geq t_{HZ}$.

**5.** If either $t_{RS}$ or $t_{RH}$ is less than the specified value, reset operations are not guaranteed.

**6.** If either $t_{RN1}$ or $t_{RN2}$ is less than the specified value, reset operations may extend to cycles preceding or following the period of reset operations.

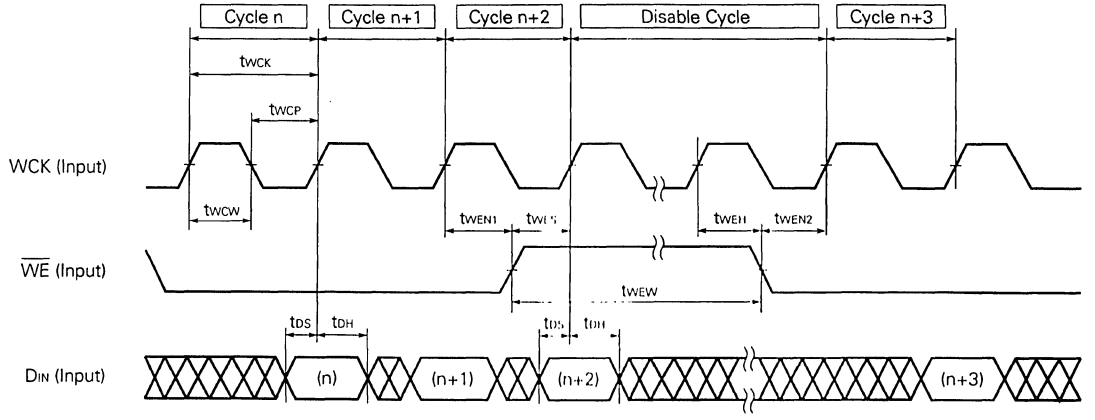
**7.** If either $t_{WES}$ or $t_{WEH}$ is less than the specified value, write disable operations are not guaranteed.

**8.** If either $t_{WEN1}$ or $t_{WEN2}$ is less than the specified value, internal write disable operations may extend to cycles preceding or following the period of write disable operations.

**9.** If either $t_{RES}$ or $t_{REH}$ is less than the specified value, read disable operations are not guaranteed.

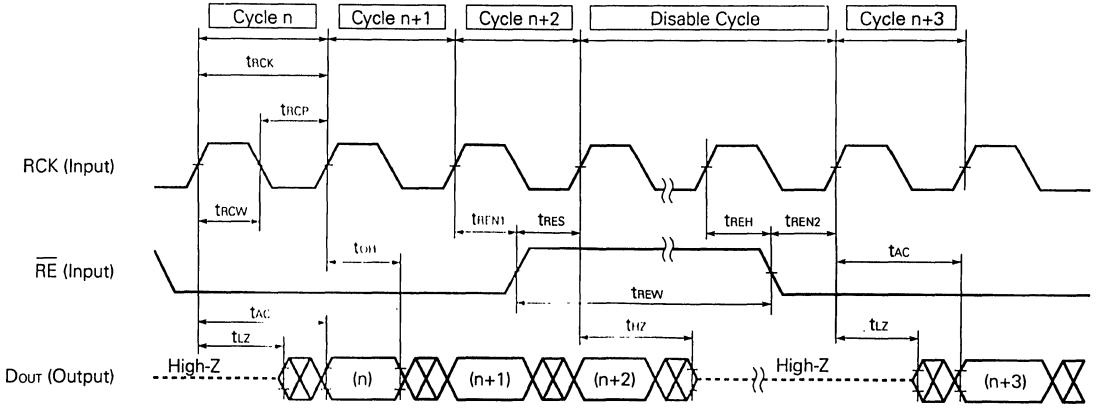
**10.** If either $t_{REN1}$ or $t_{REN2}$ is less than the specified value, internal read disable operations may extend to cycles preceding or following the period of read disable operations.

**Write Cycle**



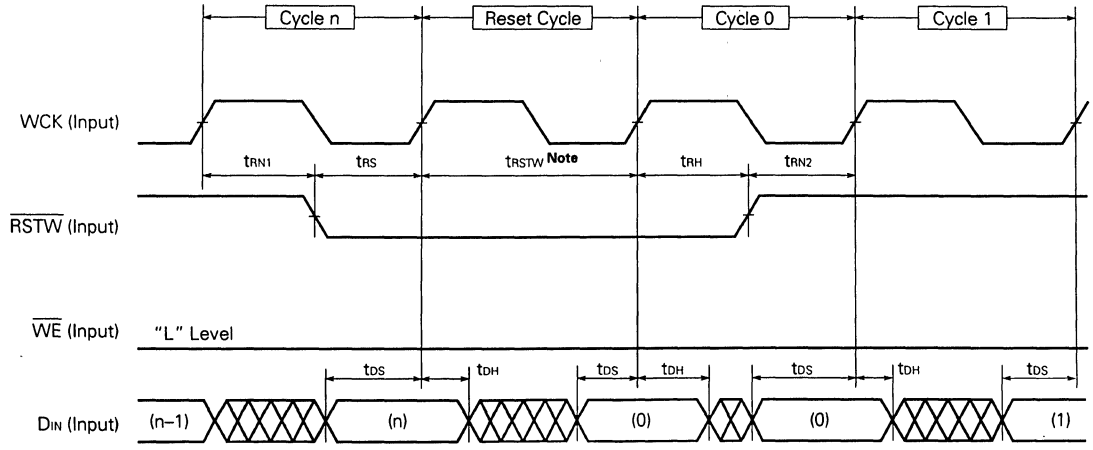

**Remark** $\overline{RSTW}$ = "H" level

**Read Cycle**



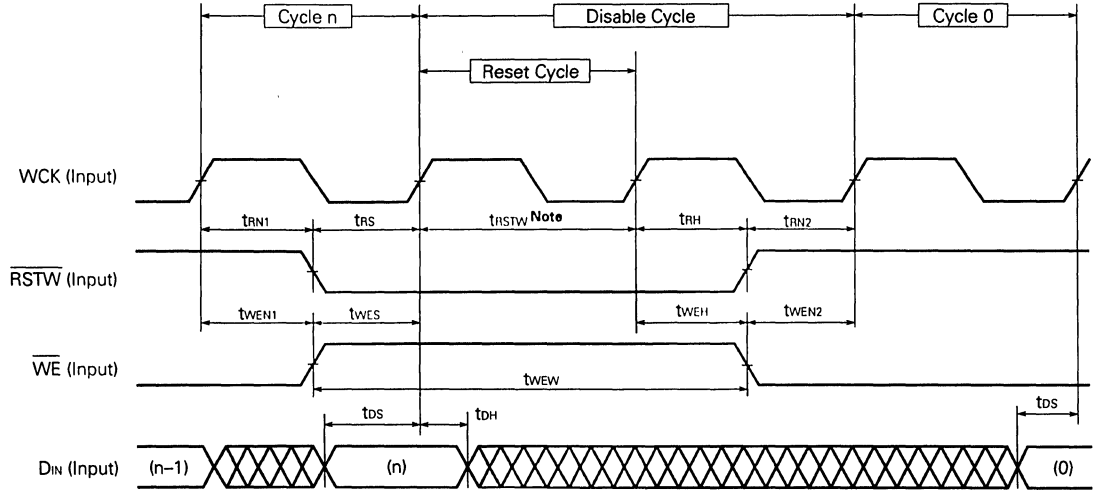

**Remark** $\overline{RSTR}$ = "H" level

**Write Reset Cycle ($\overline{WE}$ Controlled 1)**



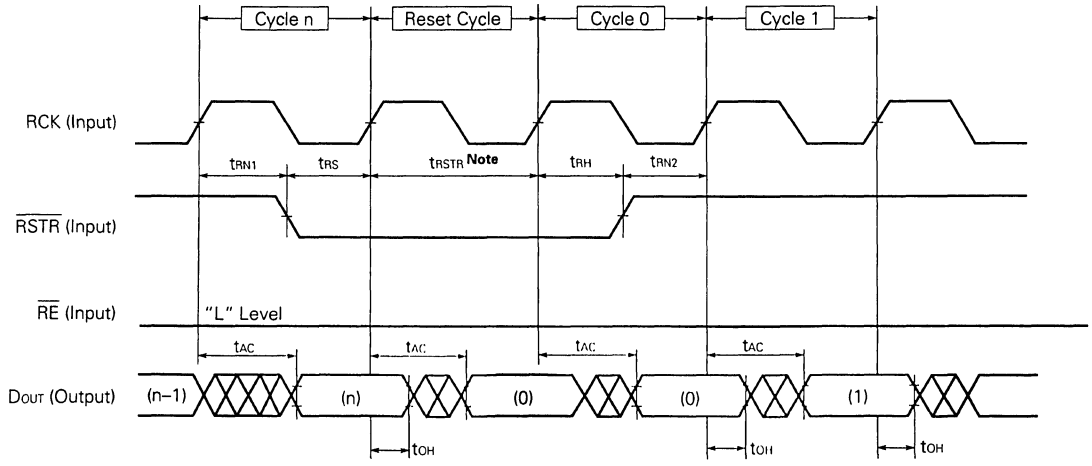

**Note** In write reset cycle, reset operation is executed even without a reset cycle ($t_{RSTW}$).
WCK can be input any number of times in a reset cycle.

**Write Reset Cycle ($\overline{WE}$ Controlled 2)**



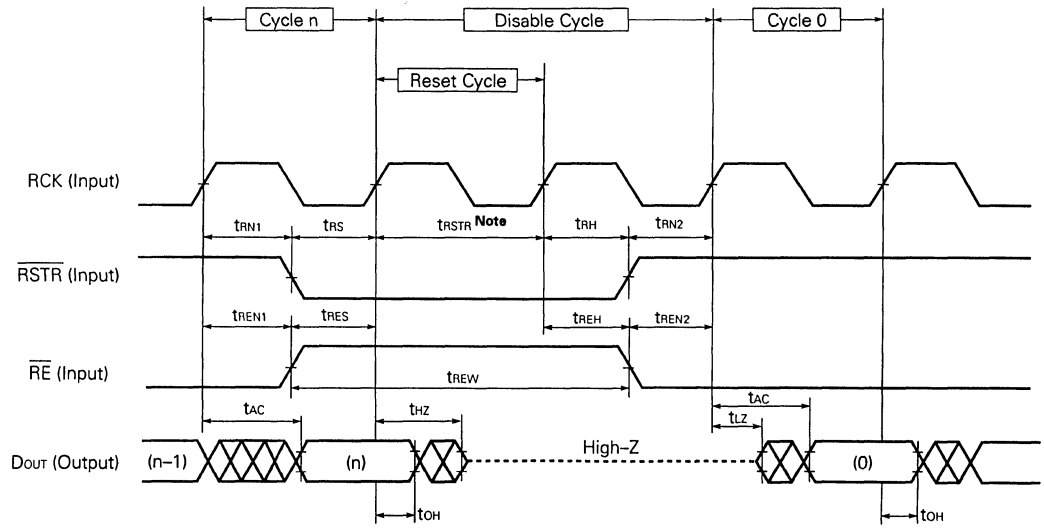

**Note** In write reset cycle, reset operation is executed even without a reset cycle ($t_{RSTW}$).
WCK can be input any number of times in a reset cycle.

**Read Reset Cycle ($\overline{RE}$ Controlled 1)**




**Note** In read reset cycle, reset operation is executed even without a reset cycle ($t_{RSTR}$). RCK can be input any number of times in a reset cycle.

**Read Reset Cycle ($\overline{RE}$ Controlled 2)**




**Note** In read reset cycle, reset operation is executed even without a reset cycle ($t_{RSTR}$). RCK can be input any number of times in a reset cycle.

## Application

- **1 H Delay Line**

  μPD485505 easily allows a 1 H (5 048 bits) delay line (see Figure 1, 2 below).

  It is also possible to change the number of delay bits depending on the cycle time as follows.

| Part Number | Cycle Time | Delay Bits |
|---|---|---|
| μPD485505-25 | 25 ns | 21 to 5 048 bits |
| μPD485505-27 | 27 ns | 19 to 5 048 bits |

To change the number of delay bits, you can choose the one of the following methods.

**Adjustments of the number of delay bits**

(1) Reset the cycle proportionate to the delay length (Figure 3).
(2) Shift the input timing of write reset ($\overline{RSTW}$) and read reset signals ($\overline{RSTR}$) according to the delay length (Figure 4).
(3) Shift the address by disabling $\overline{WE}$ or $\overline{RE}$ for the period proportionate to the delay length (Figure 5).

**Caution After power up, the μPD485505 requires the initialization of internal circuits because the read and write address pointers are not defined at that time.**

**Fig. 1 1 H Delay Line Circuit**

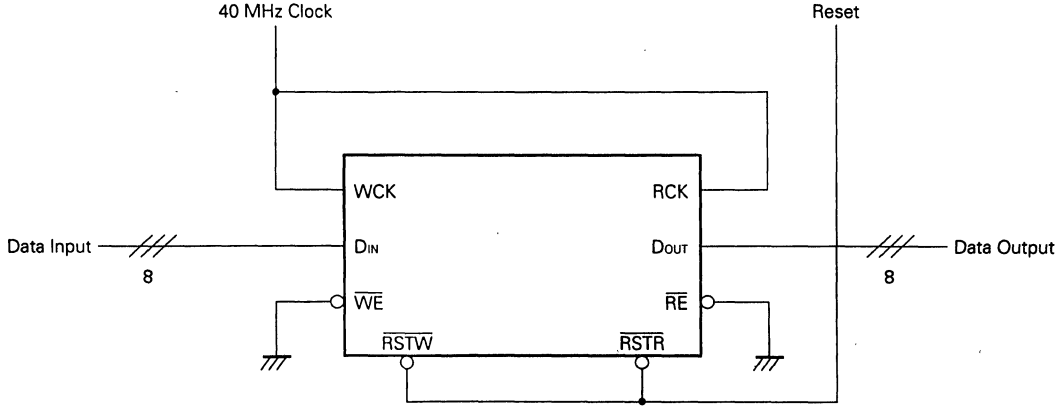
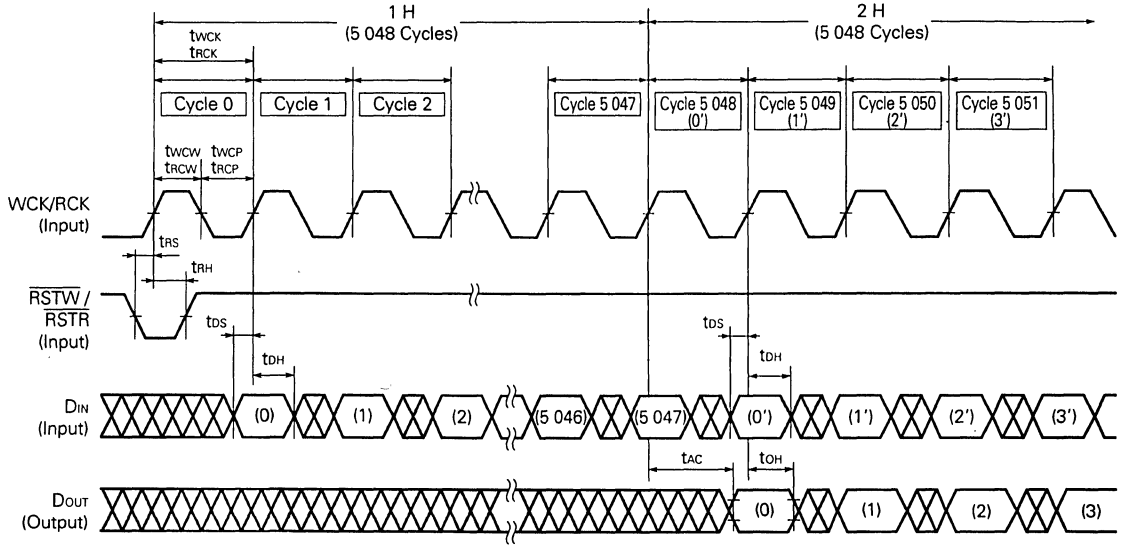
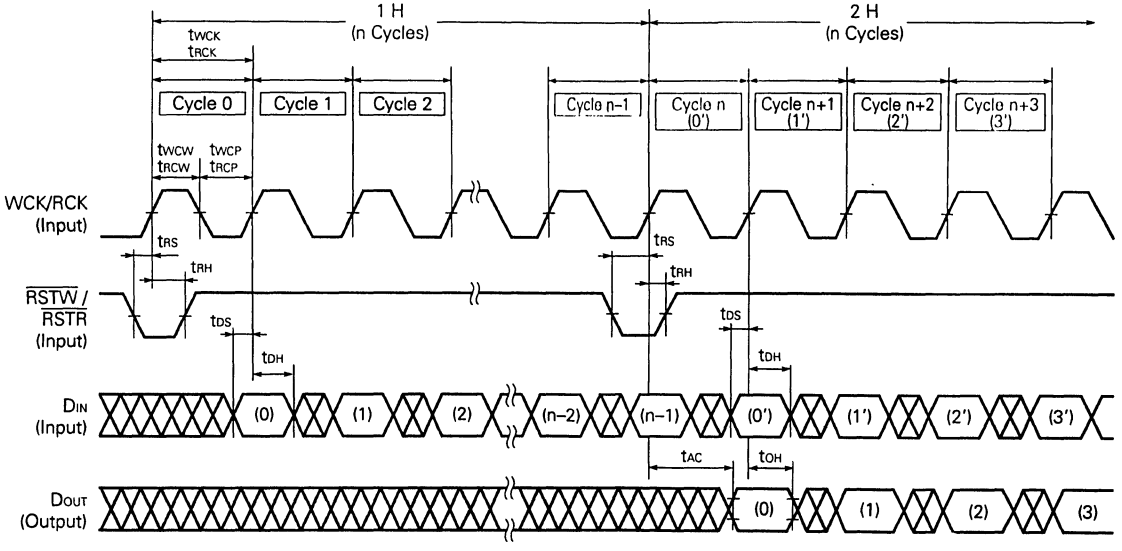


**Fig. 2 1 H Delay Line Timing**



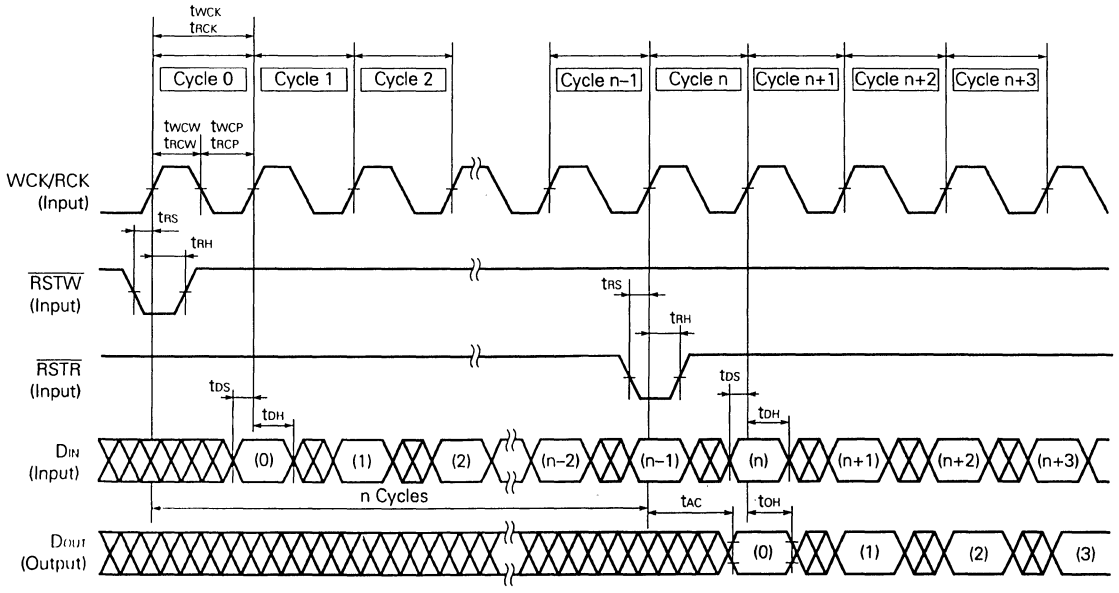

**Remark** $\overline{RE}$, $\overline{WE}$ = "L" level

**Fig. 3 n-Bit Delay Line Timing (1)**



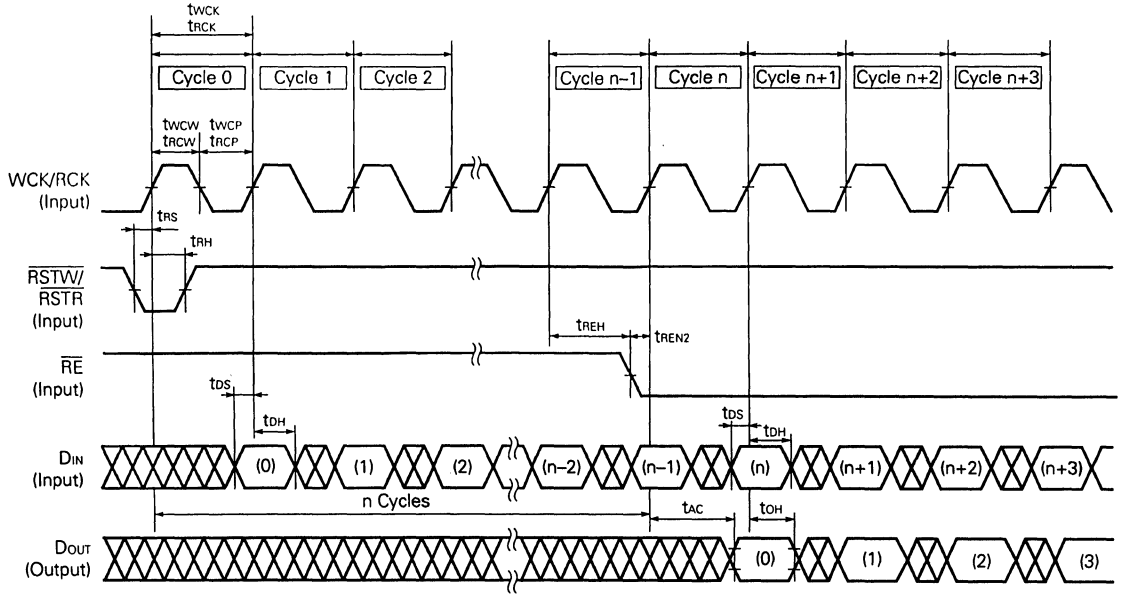

**Remark** $\overline{RE}$, $\overline{WE}$ = "L" level

**Fig. 4 n-Bit Delay Line Timing (2)**




**Remark** $\overline{RE}$, $\overline{WE}$ = "L" level

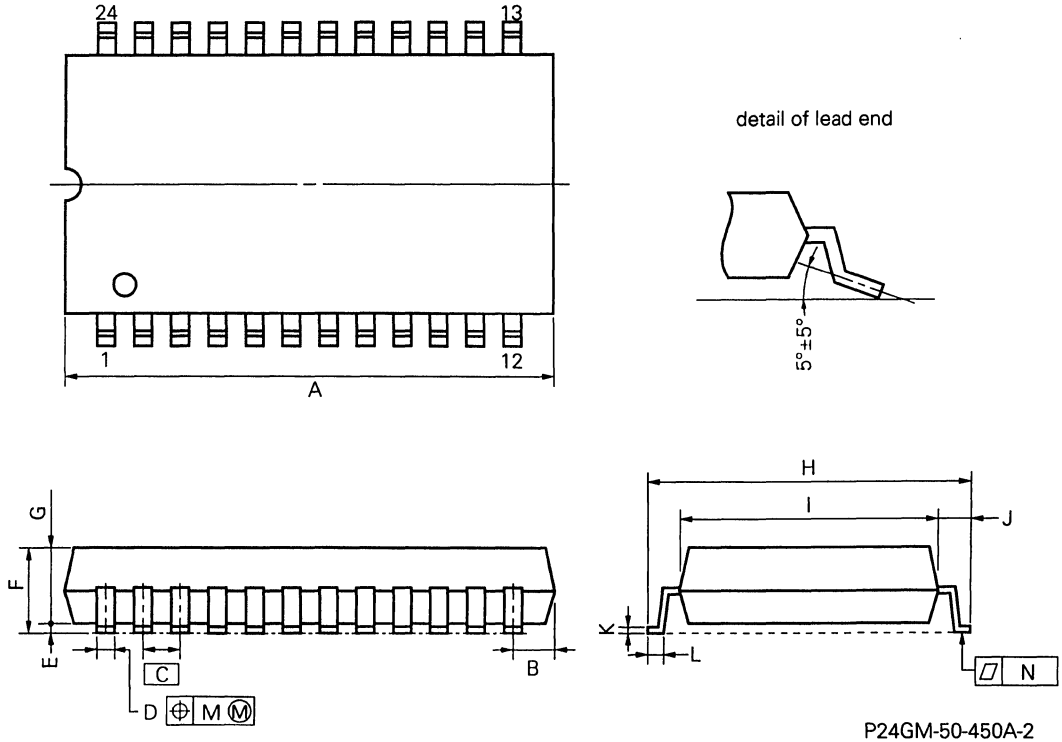
**Fig. 5 n-Bit Delay Line Timing (3)**




**Remark** $\overline{WE}$ = "L" level

## Package Drawing

### 24 PIN PLASTIC SOP (450 mil)




P24GM-50-450A-2

**NOTE**

Each lead centerline is located within 0.12 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 16.51 MAX. | 0.650 MAX. |
| B | 1.27 MAX. | 0.050 MAX. |
| C | 1.27 (T.P.) | 0.050 (T.P.) |
| D | $0.40\pm0.10$ | $0.016^{+0.004}_{-0.005}$ |
| E | $0.1^{+0.2}_{-0.1}$ | $0.004^{+0.008}_{-0.004}$ |
| F | 2.5 MAX. | 0.099 MAX. |
| G | 2.00 | 0.079 |
| H | $12.2\pm0.3$ | $0.480^{+0.013}_{-0.012}$ |
| I | 8.4 | 0.331 |
| J | 1.9 | 0.075 |
| K | $0.15^{+0.10}_{-0.05}$ | $0.006^{+0.004}_{-0.002}$ |
| L | $0.9\pm0.2$ | $0.035^{+0.009}_{-0.008}$ |
| M | 0.12 | 0.005 |
| N | 0.10 | 0.004 |

### Recommended Soldering Conditions

Please consult with our sales offices for soldering conditions of the μPD485505.

**Type of Surface Mount Device**

**μPD485505G: 24-pin Plastic SOP (450 mil)**

# Field Buffer

NEC

MOS INTEGRATED CIRCUIT

# μPD42280

## 2 M-BIT FIELD BUFFER

The μPD42280 is a high-speed field buffer equipped with a memory of 256K words × 8bit (262, 224 × 8bit) configuration. The high-speed and the low power consumption are realized in CMOS dynamic circuit.

The μPD42280 consists of FIFO (First In First Out) configuration, and the write/read operations are possible asynchronously and simultaneously.

Because it has refresh circuit internally, 1 field delay line and time axis conversion etc. are realized easily. Therefore it is suitable for YC separation between frames, interpolation between fields, reproduction of freeze picture and frame synchronizer in the digital TV, VCR systems.

### FEATURES

- 256K words × 8 bit FIFO configuration
- Write/read operations are possible asynchronously and simultaneously
- Reset is possible apart from write/read address (real time reset)
- Serial read cycle time : 30 ns (MIN.)
- Serial read access time : 25 ns (MAX.)
- Serial write cycle time : 30 ns (MIN.)
- Self refresh function incorporated
- Output enable
- All I/O TTL compatible
- CMOS low power consumption : ($t_{RCK}$, $t_{WCK}$ = 30 ns) $I_{DD}$ = 50 mA (TYP.)
- 28 pin plastic SOP (450 mil) .... μPD42280GU
- 28 pin plastic ZIP (400 mil) .... μPD42280V

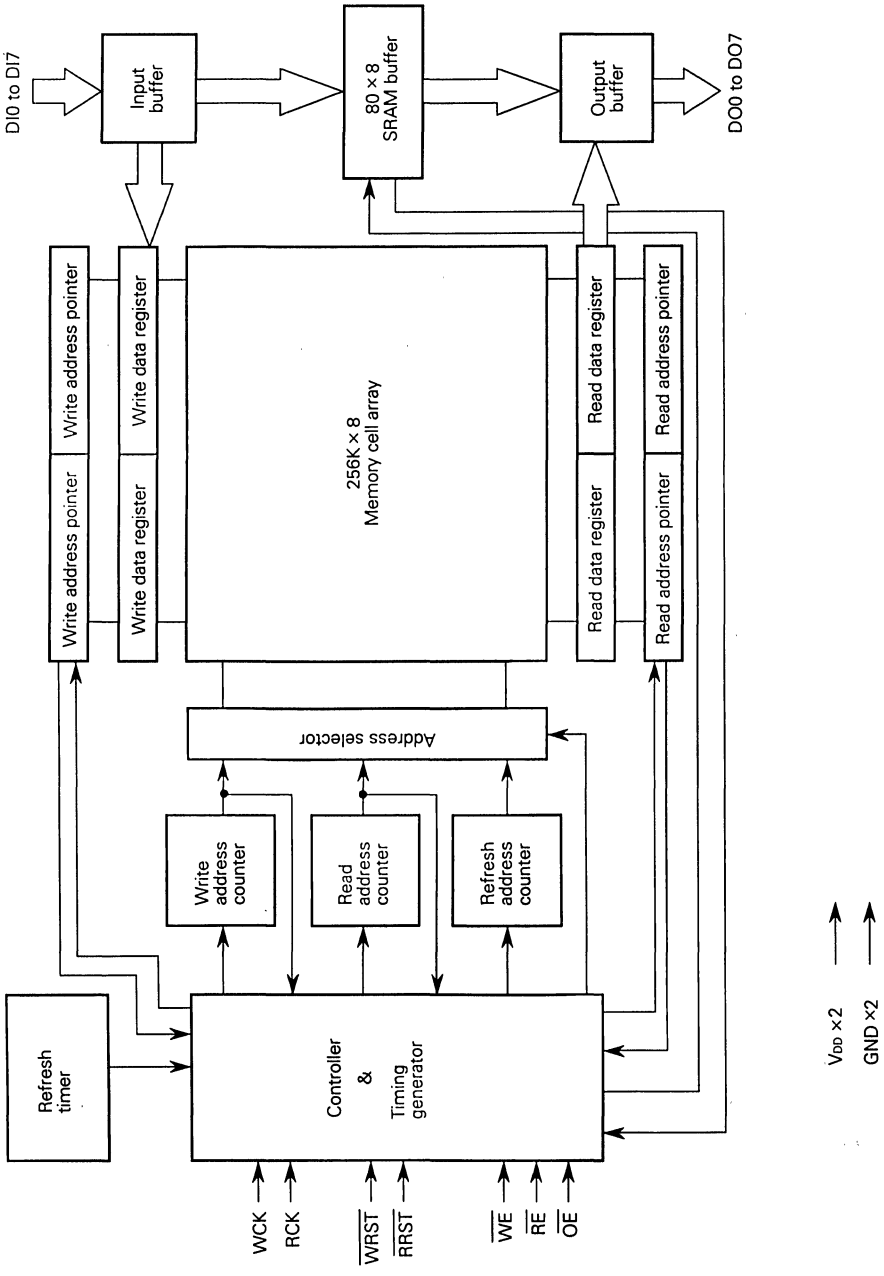
### ORDERING INFORMATION

| Part number | Package | Read cycle time (ns) | Access time (ns) | Write cycle time (ns) | Quality level |
|---|---|---|---|---|---|
| μPD42280GU-30 | 28-pin plastic SOP (450 mil) | 30 | 25 | 30 | Standard |
| μPD42280V-30 | 28-pin plastic ZIP (400 mil) | 30 | 25 | 30 | Standard |

Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

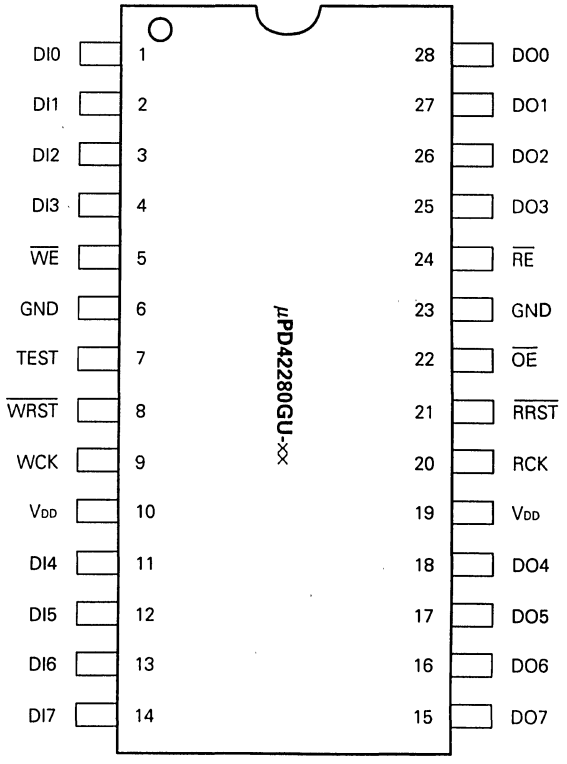
**The information in this document is subject to change without notice.**

## BLOCK DIAGRAM

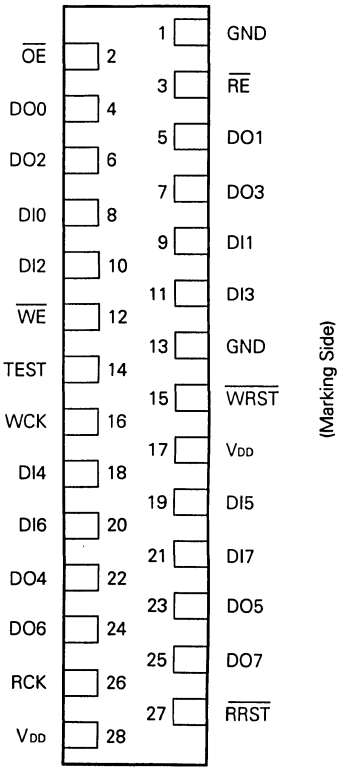


## PIN CONFIGURATION

**28 PIN SOP (450 mil)**
**(Top View)**




**28 PIN ZIP (400 mil)**
**(Bottom View)**




Pin name

| | | |
|---|---|---|
| DI0 to DI7 | : Data input | Input |
| DO0 to DO7 | : Data output | Output |
| WCK | : Write clock input | Input |
| RCK | : Read clock input | Input |
| $\overline{WE}$ | : Write enable input | Input |
| $\overline{RE}$ | : Read enable input | Input |
| $\overline{OE}$ | : Output enable input | Input |
| $\overline{WRST}$ | : Write reset input | Input |
| $\overline{RRST}$ | : Read reset input | Input |
| TEST | : Test pin | Input |
| VDD | : +5 V power source | — |
| GND | : Ground | — |

## 1. PIN FUNCTION

| Pin name | Input/Output | Function |
|---|---|---|
| DI0 to DI7 | I | This is a write data input. Fetching data is executed on the back rise up edge of WCK input cycle, and the setup/hold time ($t_{DS}$, $t_{DH}$) are specified against the edge. |
| DO0 to DO7 | O (3-state) | This is a read data output. The access time is specified from the front rise up edge of RCK cycle and determined by $t_{AC}$. It is 3 state output. |
| $\overline{WRST}$ | I | This is a reset input for initializing a write address. Fetching reset signals is executed on the front rise up edge of WCK input cycle, and the setup/hold time ($t_{RS}$, $t_{RH}$) are specified against the edge. |
| $\overline{RRST}$ | I | This is a reset input for initializing a read address. Fetching reset signals is executed on the front rise up edge of RCK input cycle, and the setup/hold time ($t_{RS}$, $t_{RH}$) are specified against the edge. |
| $\overline{WE}$ | I | This is a write operation control input. In case of high level, the internal write operation is prohibited, and the write address pointer is also stopped at the present position. $\overline{WE}$ signal is fetched on the front rise up edge of WCK input cycle. |
| $\overline{RE}$ | I | This is a read operation control input. In case of high level, the internal read operation is executed, and the read address pointer is stopped at the present position. $\overline{RE}$ signal is fetched on the front rise up edge of RCK input cycle. |
| $\overline{OE}$ | I | This is a read data control input. In case of high-level, DO0 to DO7 will be high impedance. In the read address pointer an increment is executed synchronously for RCK, not depending on the $\overline{OE}$ signal input level. $\overline{OE}$ signal is fetched on the front rise up edge of RCK input cycle. |
| WCK | I | This is a write clock input. The write operation is executed synchronously for the write clock when $\overline{WE}$ is in low level, and in the write address pointer, the increment is executed at the same time. |
| RCK | I | This is a read clock input. The read operation is done synchronously for the read clock, and when $\overline{RE}$ is in low level, in the read address pointer the increment is executed at the same time. |
| TEST | I | This is a pin for testing. It is to be fixed on low level on the practical operation. |

## 2. EXPLANATION FOR THE MEMORY AND FUNCTION BLOCK

### 2.1 MEMORY CELL ARRAY

This is a memory cell array in this product consisting of dynamic memory cells, with 256K × 8 (2M bit) configuration.

### 2.2 INPUT BUFFER

This is a buffer to input the signals of DI0 to DI7 to the write data register or SRAM buffer.

### 2.3 OUTPUT BUFFER

This is a buffer to output the data from the read data register or SRAM buffer to DO0 to DI7.

### 2.4 WRITE DATA REGISTER/WRITE ADDRESS POINTER

This is a register to temporarily store the data input to DI0 to DI7. The input data is stored in the address directed by the write address pointer. In the write address pointer, its content increases every WCK input. When the write data register is filled with data, the data are transferred to the memory cell array together, and the write address pointer is reset to the 0 address. The data are transferred by 64 words unit.

### 2.5 READ DATA REGISTER/READ ADDRESS POINTER

This is a register to temporarily store the data transferred together from the memory cell array. The data in the register directed by the read address pointer is output to DO0 to DO7. In the read address pointer, its content increases every RCK input. When the read data register is emptied, the data read from the memory cell array is transferred together to the register, and the read address pointer is reset to the 0 address. The data are transferred by 64 words unit.

### 2.6 SRAM

This is a buffer to store the data for 80 words after the write address is reset by $\overline{\text{WRST}}$. Also, when a read address is reset by the input of $\overline{\text{RRST}}$ signals, the data for 80 words after the reset is output (to DO0 to DO7) from SRAM buffer.

### 2.7 WRITE ADDRESS COUNTER

This is a counter to direct the row address of the write data. When the data is transferred to memory cell array from the write data register, the content of the counter increases. When input of the last address is attained, the content of the counter is reset to the 0 address.

### 2.8 READ ADDRESS COUNTER

This is a counter to direct the row address of the read data. When the data is transferred to the read data register from memory cell array, the content of the counter increases. When input of $\overline{\text{RRST}}$ signals or the last address is attained, the content of the counter is reset to the 0 address.

### 2.9 REFRESH ADDRESS COUNTER/REFRESH TIMER

This is a counter to direct the refresh address. Its content is increased one by one by the refresh timer. Because self refresh function is incorporated, the refresh operation is executed automatically.

### 2.10 ADDRESS SELECTOR

This is a selector which selects one of the addresses directed by the address counter, read address counter and refresh address counter as the row address of memory cell array.

### 2.11 CONTROLLER/TIMING GENERATOR

Each block is controlled by the control signals from the input pins of this block.

## 3. MEMORY OPERATION

### 3.1 WRITE OPERATION

When $\overline{WE}$ input is in low level, the data input to DI0 to DI7 is written into the write data register every 8 bits together synchronizing with WCK input.

The write data should be input to meet the setup time and the hold time for the back rise up edge of WCK input cycle.

When $\overline{WE}$ input is attained to a high level, the write operation is prohibited. The write address pointer is stopped at the position of high level input state. When the low level is input again, the operation is started from the stopped address.

Though the write operation is prohibited at any time, $\overline{WE}$ signal should be input to meet the set up time and the hold time for the rise up edge of WCK.

### 3.2 READ OPERATION

When $\overline{RE}$ input and $\overline{OE}$ input are in low level, the data is output to DO0 to DO7 from the read data register every 8 bits together synchronizing with RCK input.

The read data is output after access time ($t_{AC}$) from the rise up edge of RCK input cycle.

When $\overline{RE}$ input is attained to a high level, the read address pointer is stopped at the position of the high level input state. When inputting a low level again, the operation is started from the stopped address.

When $\overline{OE}$ input is attained to a high level, the output will be a high impedance. The content of the read address pointer increases synchronously with RCK input, not depending on the input level of $\overline{OE}$ signal.

Though the read operation can be prohibited at any time, $\overline{RE}$ signal/$\overline{OE}$ signal should be input to meet the setup time and hold time for the rise up edge of RCK.

When the new data is read, the write address should be preceded 200 to 262, 223 or less cycles than the read address.

When the old data is read, the difference between the write address and the read address should be 0 cycle (the read address and the write address are the same).

When the read address and the write address compete with each other by double speed conversion etc. in the same field, in the data of the last 192 words the old data (the data prior to 1 field) may be output.

### 3.3 RESET OPERATION

The reset signal can be input any time, without depending on $\overline{WE}$, $\overline{RE}$ and $\overline{OE}$ signals. $\overline{WRST}$ and $\overline{RRST}$ signals should be input to meet the setup time and the hold time for the rise up edge of WCK, RCK inputs. When the reset signal is input in disable cycles, the reset operation is executed after the disable cycles.

The output and input of data are possible from the cycle (0 address) where low level is input to $\overline{WRST}$, $\overline{RRST}$.

### 3.4 INITIALIZE

Following initializing should be taken when the power supply is ON.

(1) Stand-by period should be taken more than 100 μs.

(2) $\overline{WRST}$ and $\overline{RRST}$ signals should be input for initializing of the write address and read address.

(3) More than 82 dummy cycles should be taken where low level is input to $\overline{RE}$ and $\overline{WE}$.

(4) Ordinary operation is executed after next low level is input to $\overline{WRST}$ and $\overline{RRST}$.

**Remark** New data: Data written latest,
Old data : Stored data just before write operation

## 4. ELECTRIC CHARACTERISTICS

### ABSOLUTE MAXIMUM RATING ($T_A$ = +25 °C)

| Parameter | Symbol | Conditions | Ratings | Unit |
|---|---|---|---|---|
| Operational Power Supply | $V_{DD}$ | | −1.0 to +7.0 | V |
| Pin Voltage | $V_T$ | | −1.0 to $V_{DD}$ +0.5 (7.0 V or less) | V |
| Output Current | $I_O$ | | ±20 | mA |
| Operating Ambient Temperature | $T_A$ | | −20 to +70 | °C |
| Storage Temperature | $T_{stg}$ | | −55 to +125 | °C |

### RECOMMENDED OPERATION RANGE

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Power Supply Voltage 1 | $V_{DD}$ | | 4.5 | 5.0 | 5.5 | V |
| High Level Input Voltage | $V_{IH}$ | | 2.4 | | $V_{DD}$ + 0.5 | V |
| Low Level Input Voltage | $V_{IL}$ | | −1.0 | | 0.8 | V |
| Ambient Temperature | $T_A$ | | −20 | | +70 | V |

### DC CHARACTERISTICS (BASED ON THE RECOMMENDED OPERATING CONDITIONS)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Operating Current 1 | $I_{CC1}$ | $t_{WCK}$, $t_{RCK}$ = 30 ns | | 50 | 90 | mA |
| Standby Current | $I_{CCS}$ | WCK, RCK = 'L' (TTL input) | | 4 | 10 | mA |
| Input Leak Current | $I_I$ | $V_I$ = 0 to $V_{DD}$<br>Other input: 0 V | −10 | | 10 | μA |
| Output Leak Current | $I_O$ | $V_O$ = 0 to $V_{DD}$<br>Do: High impedance | −10 | | 10 | μA |
| High Level Output Voltage | $V_{OH}$ | $I_{OH}$ = −1 mA | 2.4 | | | V |
| Low Level Output Voltage | $V_{OL}$ | $I_{OL}$ = 2 mA | | | 0.4 | V |

### INPUT/OUTPUT CAPACITANCE ($T_A$ = +25 °C, f = 1 MHz)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_I$ | | | | 5 | pF |
| Output Capacitance | $C_O$ | | | | 7 | pF |

## AC CHARACTERISTICS (BASED ON THE RECOMMENDED OPERATING CONDITIONS) Note 1, 2, 3, 5

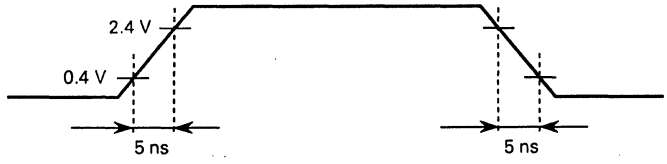
| Parameter | Symbol | μPD42280-30 | | Unit |
|---|---|---|---|---|
| | | MIN. | MAX. | |
| Write Clock (WCK) Cycle Time | $t_{WCK}$ | 30 | | ns |
| Write Clock Active Time | $t_{WCW}$ | 12 | | ns |
| Write Clock Precharge Period | $t_{WCP}$ | 12 | | ns |
| Read Clock (RCK) Cycle Time | $t_{RCK}$ | 30 | | ns |
| Read Clock Active Time | $t_{RCW}$ | 12 | | ns |
| Read Clock Precharge Period | $t_{RCP}$ | 12 | | ns |
| Access Time | $t_{AC}$ | | 25 | ns |
| Output Hold Time | $t_{OH}$ | 5 | | ns |
| Output Low Impedance Time from RCK Rise Note 4 | $t_{LZ}$ | 5 | 25 | ns |
| Output High Impedance Time from RCK Rise Note 4 | $t_{HZ}$ | 5 | 25 | ns |
| Input Data, Setup Time | $t_{DS}$ | 7 | | ns |
| Input Data, Hold Time | $t_{DH}$ | 3 | | ns |
| Reset Setup Time from WCK or RCK Rise Note 6 | $t_{RS}$ | 7 | | ns |
| Reset Hold Time from WCK or RCK Rise Note 6 | $t_{RH}$ | 3 | | ns |
| Reset Non Selection Time 1 from WCK or RCK Rise Note 7 | $t_{RN1}$ | 3 | | ns |
| Reset Non Selection Time 2 from WCK or RCK Rise Note 7 | $t_{RN2}$ | 7 | | ns |
| $\overline{WE}$ Setup Time from WCK Rise Note 8 | $t_{WFS}$ | 7 | | ns |
| $\overline{WE}$ Hold Time from WCK Rise Note 8 | $t_{WFH}$ | 3 | | ns |
| $\overline{WE}$ Non Selection Time 1 from WCK Rise Note 9 | $t_{WEN1}$ | 3 | | ns |
| $\overline{WE}$ Non Selection Time 2 from WCK Rise Note 9 | $t_{WEN2}$ | 7 | | ns |
| $\overline{RE}$ Setup Time from RCK Rise Note 8 | $t_{RES}$ | 7 | | ns |
| $\overline{RE}$ Hold Time from RCK Rise Note 8 | $t_{REH}$ | 3 | | ns |
| $\overline{RE}$ Non Selection Time 1 from RCK Rise Note 9 | $t_{REN1}$ | 3 | | ns |
| $\overline{RE}$ Non Selection Time 2 from RCK Rise Note 9 | $t_{REN2}$ | 7 | | ns |
| $\overline{OE}$ Setup Time from RCK Rise Note 10 | $t_{OES}$ | 7 | | ns |
| $\overline{OE}$ Hold Time from RCK Rise Note 10 | $t_{OEH}$ | 3 | | ns |
| $\overline{OE}$ Non Selection Time 1 from RCK Rise Note 11 | $t_{OEN1}$ | 3 | | ns |
| $\overline{OE}$ Non Selection Time 2 from RCK Rise Note 11 | $t_{OEN2}$ | 7 | | ns |
| $\overline{WE}$ High Level Period | $t_{WEW}$ | 0 | | ns |
| $\overline{RE}$ High Level Period | $t_{REW}$ | 0 | | ns |
| $\overline{OE}$ High Level Period | $t_{OEW}$ | 0 | | ns |
| $\overline{WRST}$ Low Level Period (Write Reset Period) | $t_{WRST}$ | 0 | | ns |
| $\overline{RRST}$ Low Level Period (Read Reset Period) | $t_{RRST}$ | 0 | | ns |
| Rise Up, Fall Down Time | $t_T$ | 3 | 35 | ns |

**Notes**

1. All voltages are determined by referring to the ground level.
2. Measurement by $t_T$ = 5 ns.
3. Input voltage standard levels in the timing specification are $V_{IH}$ = 2.4 V and $V_{IL}$ = 0.4 V. Transferring time $t_T$ is determined between $V_{IH}$ = 2.4 V and $V_{IL}$ = 0.4 V.
4. $t_{LZ}$, $t_{HZ}$ are measured by ±200 mV from a stationary state. And $t_{LZ}$ is equal to, or more than $t_{HZ}$.
5. The reference level of input signals is 1.5 V.
6. When the reset pulse which does not meet $t_{RS}$, $t_{RH}$ is input, the reset operation is not assured.
7. When the reset pulse which does not meet $t_{RN1}$, $t_{RN2}$ is input, the reset operation may affect the cycles before and after.
8. When $\overline{WE}$ ($\overline{RE}$) pulse which deos not meet $t_{WES}$, $t_{WEH}$ (or $t_{RES}$, $t_{REH}$) is input, the write (read) disable operation is not assured.
9. When $\overline{WE}$ ($\overline{RE}$) pulse which does not meet $t_{WEN1}$, $t_{WEN2}$ (or $t_{REN1}$, $t_{REN2}$) is input, the write (read) disable operation may affect the cycles before and after.
10. When $\overline{OE}$ pulse which deos not meet $t_{OES}$, $t_{OEH}$ is input, the output disable operation is not assured.
11. When $\overline{OE}$ pulse which deos not meet $t_{OEN1}$, $t_{OEN2}$ is input, the output disable operation may affect the cycles before and after.

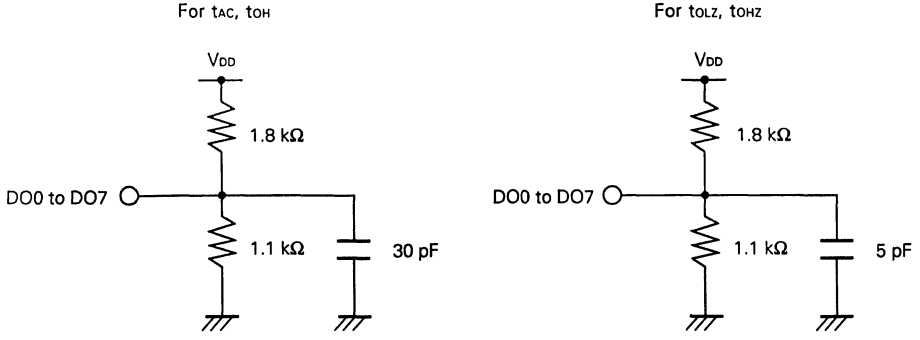
## AC CHARACTERISTICS MEASUREMENT CONDITION

### Input timing specification




### Output timing specification


### DO external load



**RESTRICTIONS**

1. When the new data is read, the write address should precede 200 to 262, 223 or less cycles than the read address. When the old data is read, the difference between the write address and the read address should be 0 cycle (the read address and the write address are same).
2. As this product transfers the data between the data register and the memory cell array by 64 word unit, when the low level is input to $\overline{WRST}$ at the n address after the 80 address, the old data up to the m address shown in the following formula after the n address is not guaranteed.

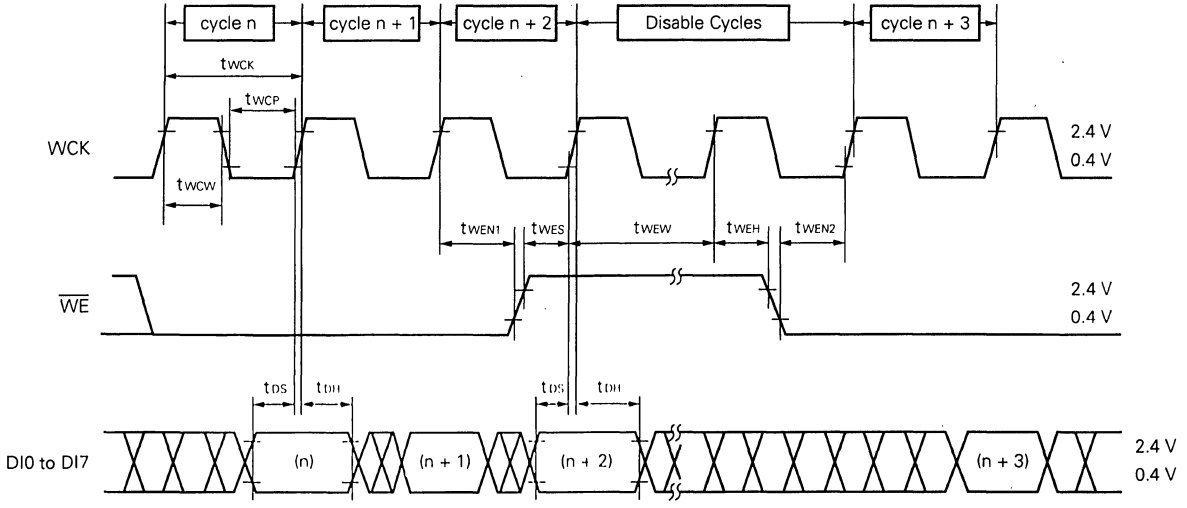
$$m = 143 + \text{int}\left(\frac{n - 80}{64}\right) \times 64$$

$\text{int}\left(\frac{n - 80}{64}\right)$ shows only integral parts of the value which the n – 80 is divided by 64.
For example: in case of n = 280, it is 3.
Therefore, m = 335, the old data from 280 address to 335 address is not guaranteed.

3. The low level of $\overline{WRST}$ signals should be input taking the interval of more than 82 address from last $\overline{WRST}$ low level input.
4. The low level of $\overline{RRST}$ signals should be input taking the interval of more than 82 address from last $\overline{RRST}$ low level input.
5. $\overline{RE}$ signals should not rise at 0 address and reset cycle.

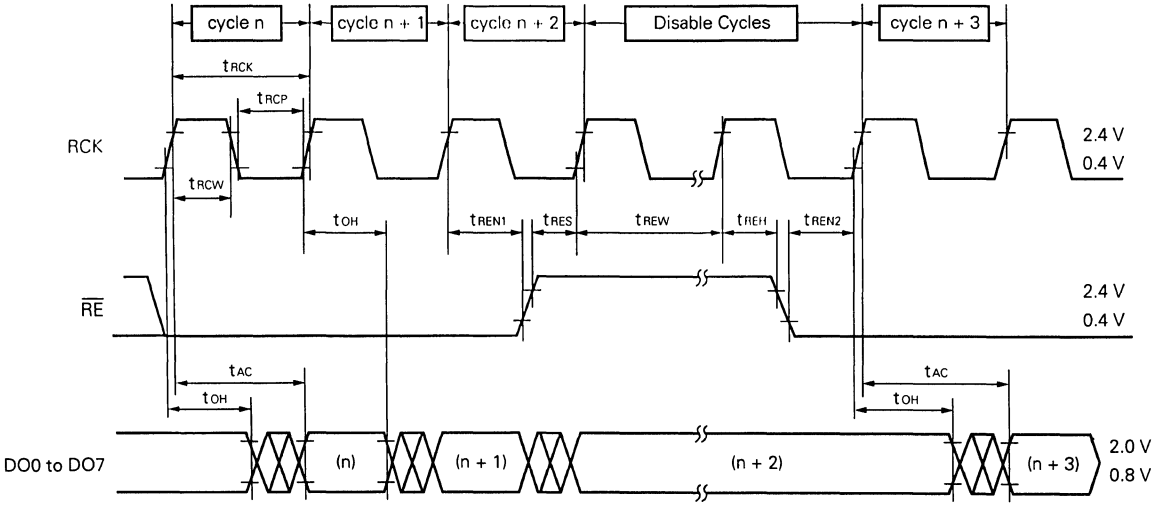
## TIMING DIAGRAM

### Write cycle



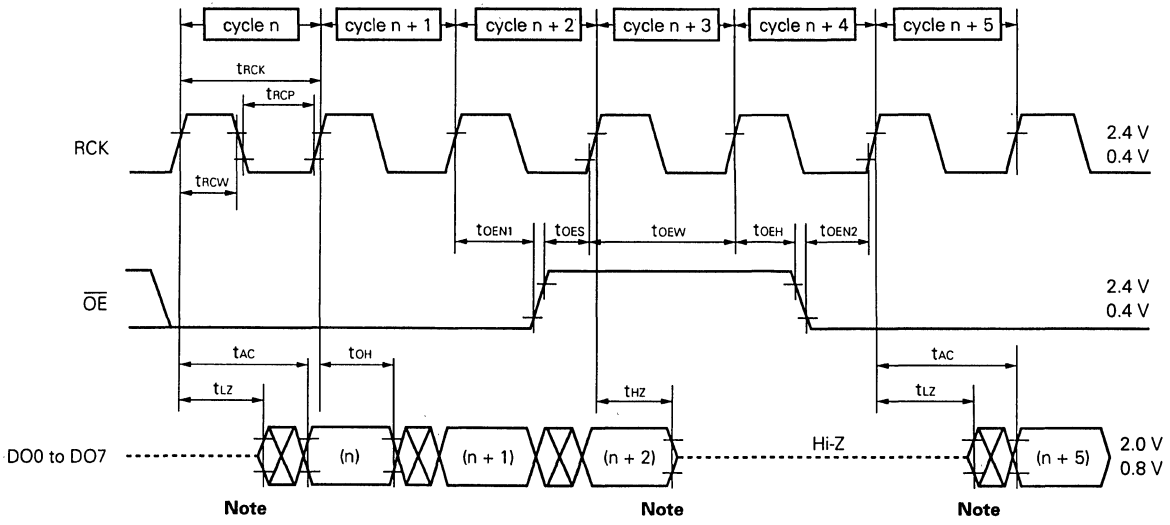

**Remark** $\overline{WRST}$ = "H" level

### Read cycle ($\overline{RE}$ control)




**Remark** $\overline{RRST}$ = "H" level, $\overline{OE}$ = "L" level

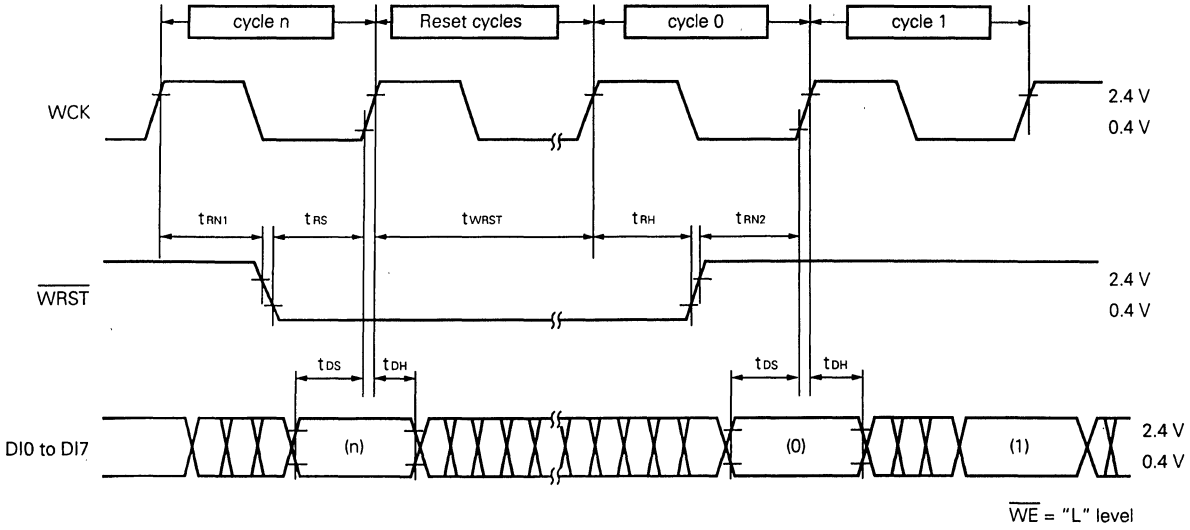
**Read cycle ($\overline{OE}$ control)**




**Remark** $\overline{RRST}$ = "H" level, $\overline{RE}$ = "L" level

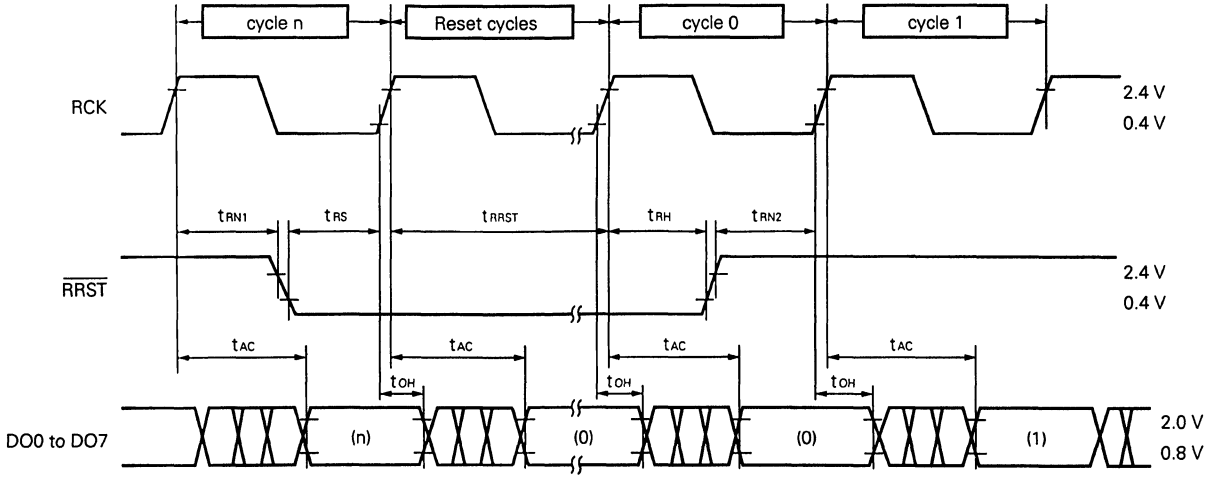
**Note** $t_{LZ}$ and $t_{HZ}$ are measured in ±200 mV point from the steady state of DO.

### Write reset cycle



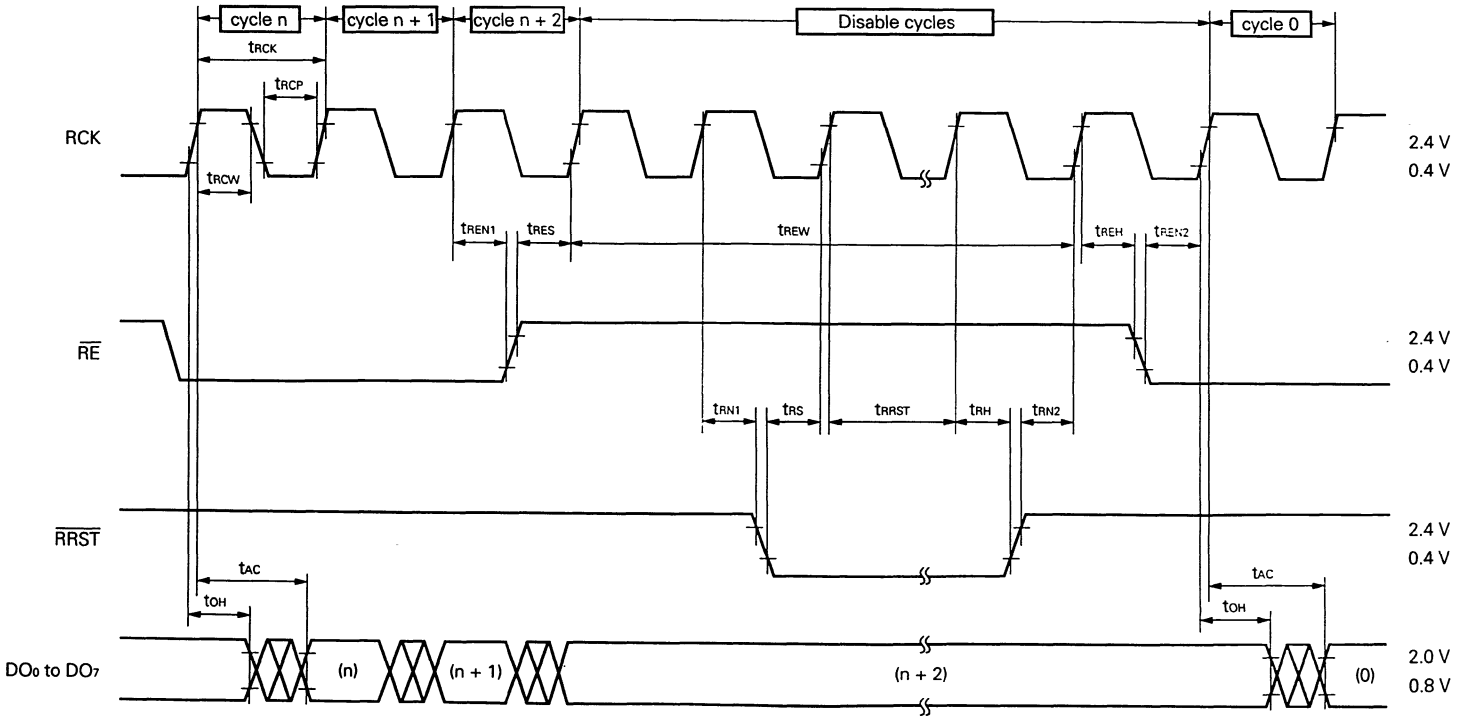

WE = "L" level

### Read reset cycle

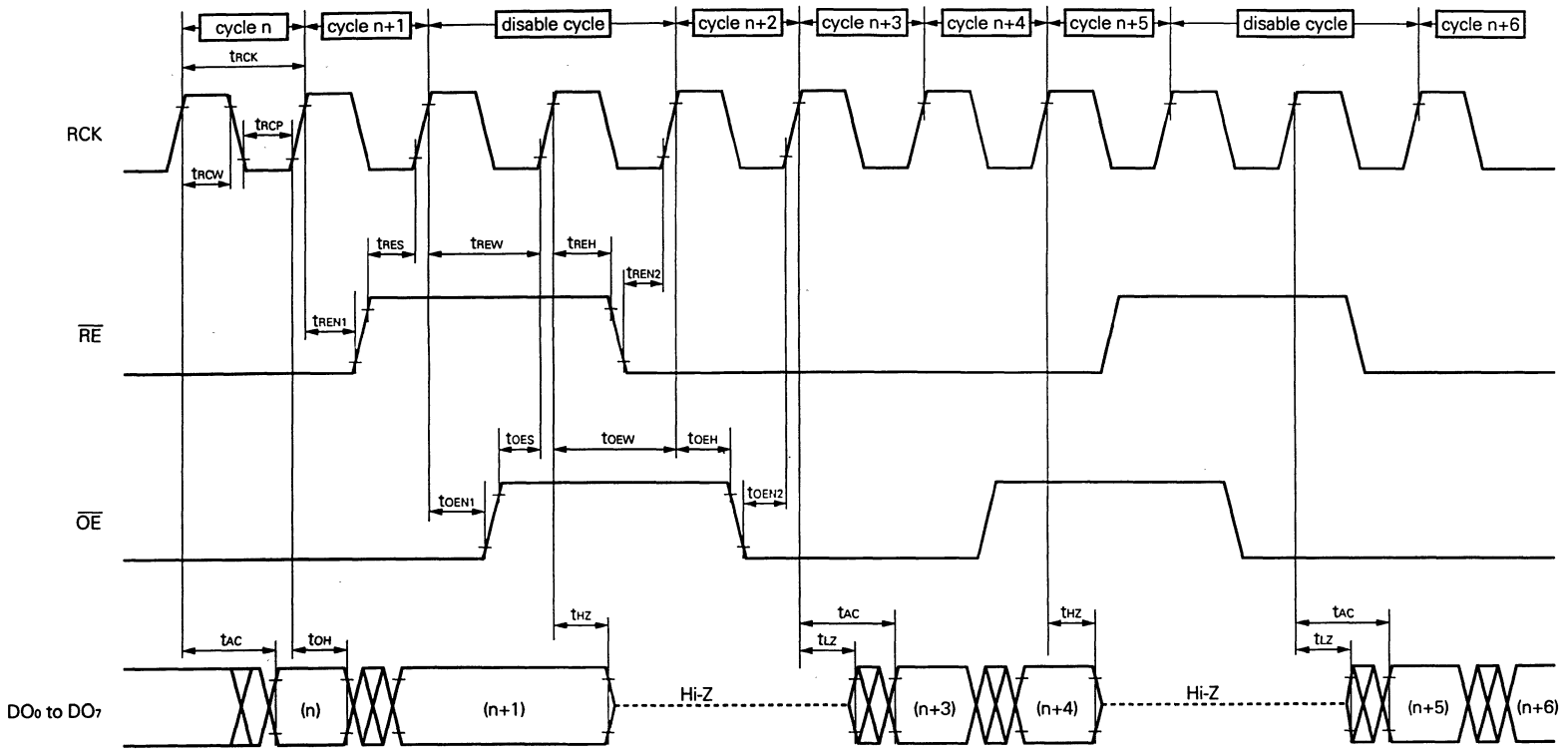



RE, OE = "L" level

**Read cycle ($\overline{RE}$, $\overline{RRST}$ control)**



**Read cycle ($\overline{RE}$, $\overline{OE}$ control)**

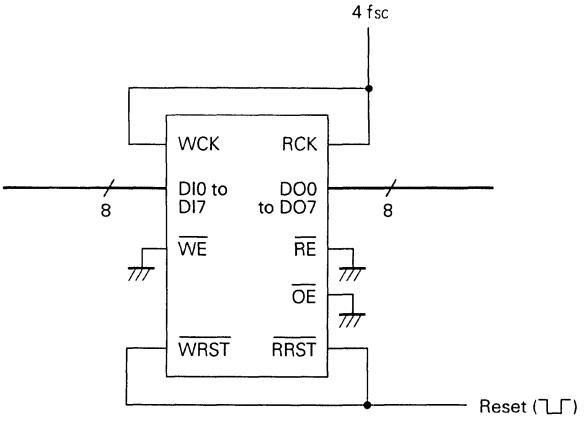


## 5. APPLICATION EXAMPLE

**(1) 1 field delay line (the old data read)**

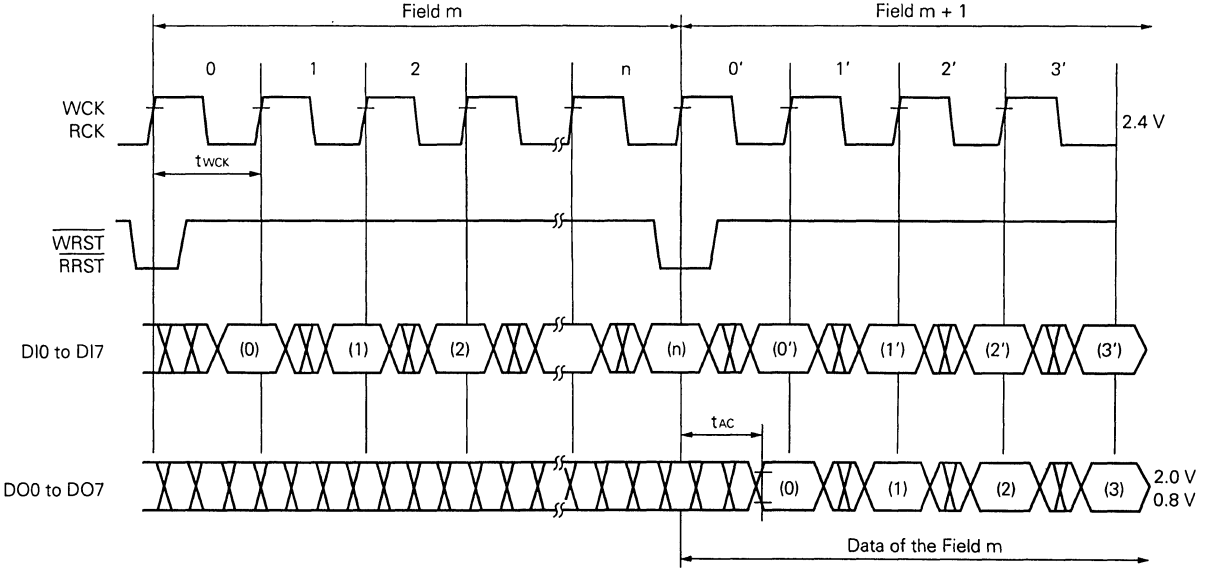
With connection as shown in Fig. 1 by inputting the reset every 1 field cycle (in common with $\overline{WRST}$ and $\overline{RRST}$), 1 field delay line is realized easily (See Fig. 2). When the difference between the write address and the read address is 0 (the read address and the write address are same), the old data is read as shown in Fig. 2

**Fig. 1 Circuit of 1 field delay line**




fsc : Color subcarrier frequency

**Fig. 2 1 field delay line timing diagram**

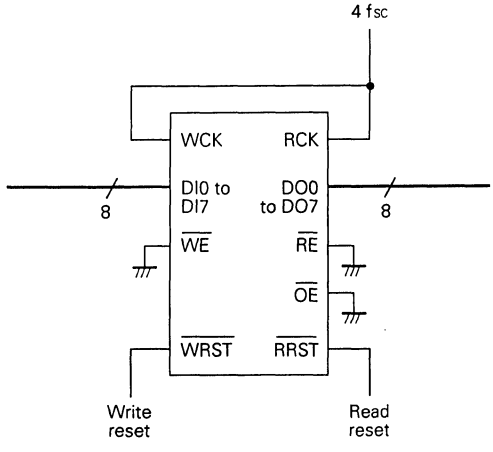



The application circuits and their parameters are for references only and are not intended for use in actual design-in's.

**(2) The new data read**

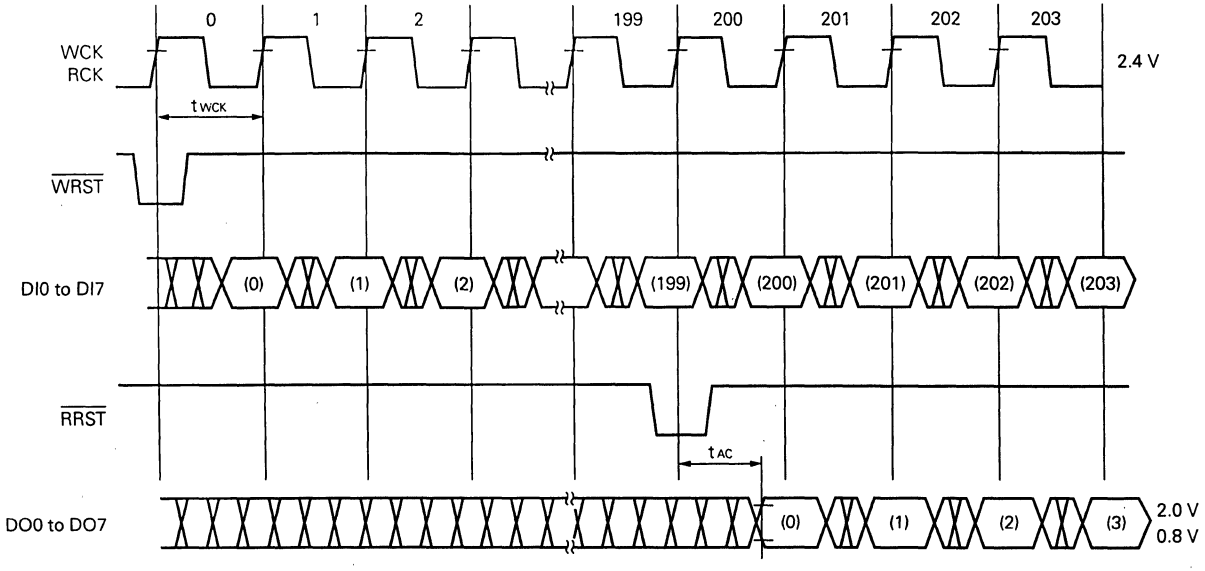
With connection as shown in Fig. 3 by inputting $\overline{RRST}$ 200 cycle after $\overline{WRST}$, new data (the latest written data) can be read. (See Fig.4)

**Fig. 3 Circuit of new data read**




fsc : Color subcarrier frequency

**Fig. 4 The new data read timing diagram**



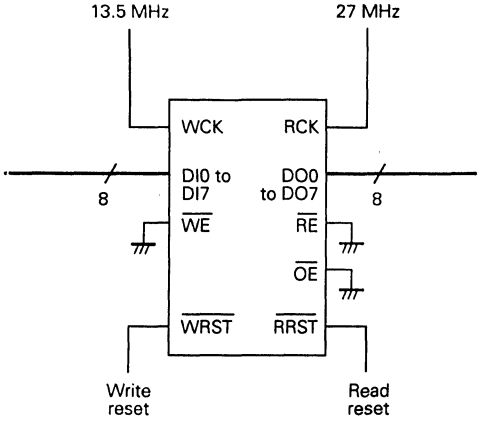

The application circuits and their parameters are for references only and are not intended for use in actual design-in's.

**(3) Double speed conversion**

By reading with double cycles for the write cycle, the double speed conversion can be done. Fig. 5 shows an example circuit in which the data is written by 13.5 MHz and the data is read by 27 MHz. In this example, the same field is read 2 times (Timing Fig. 6).

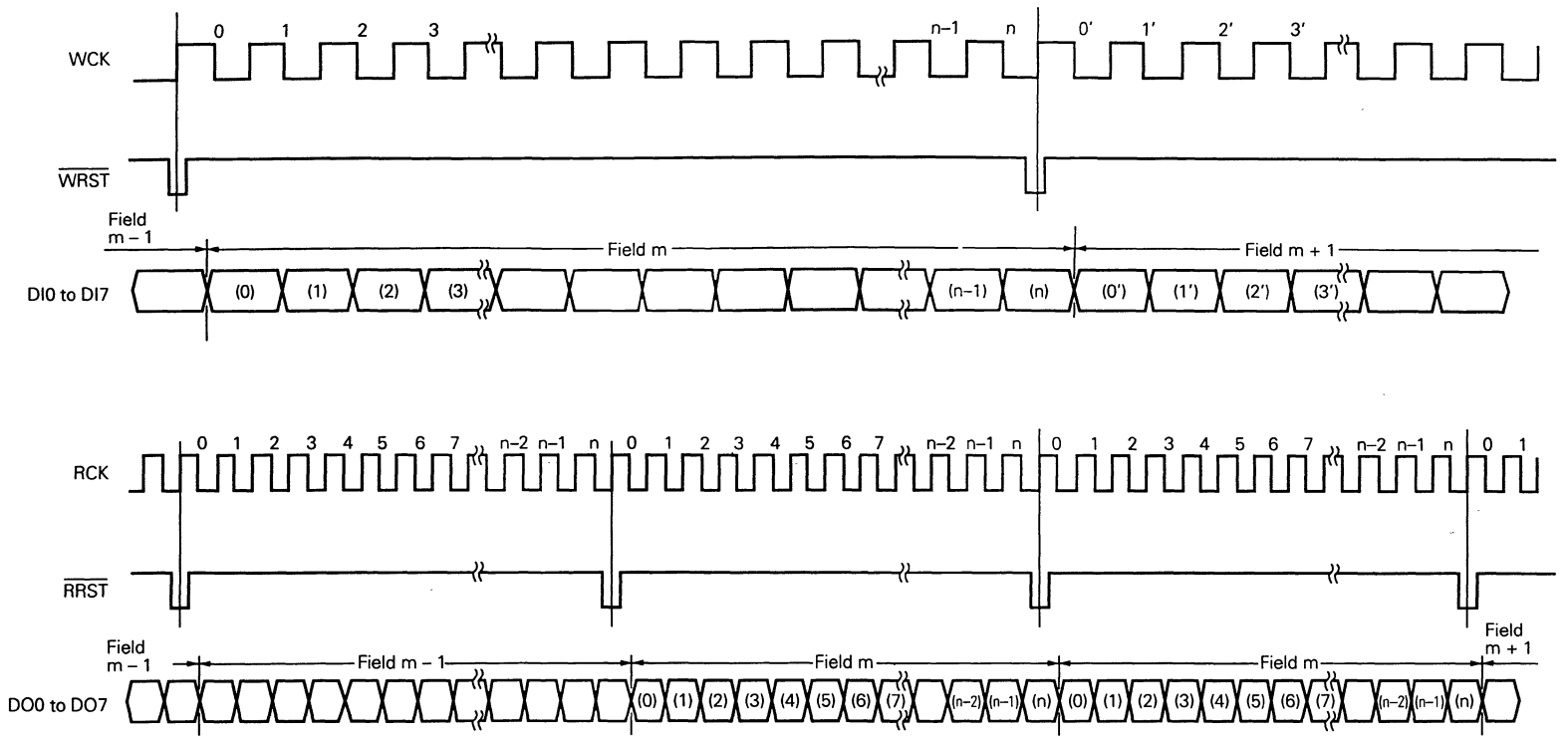
**Caution Note that when the read and the write compete each other in the same field like this application, in the last 192 words, the data before 1 field may be output.**

**Fig. 5 Circuit of double speed conversion**




The application circuits and their parameters are for references only and are not intended for use in actual design-in's.

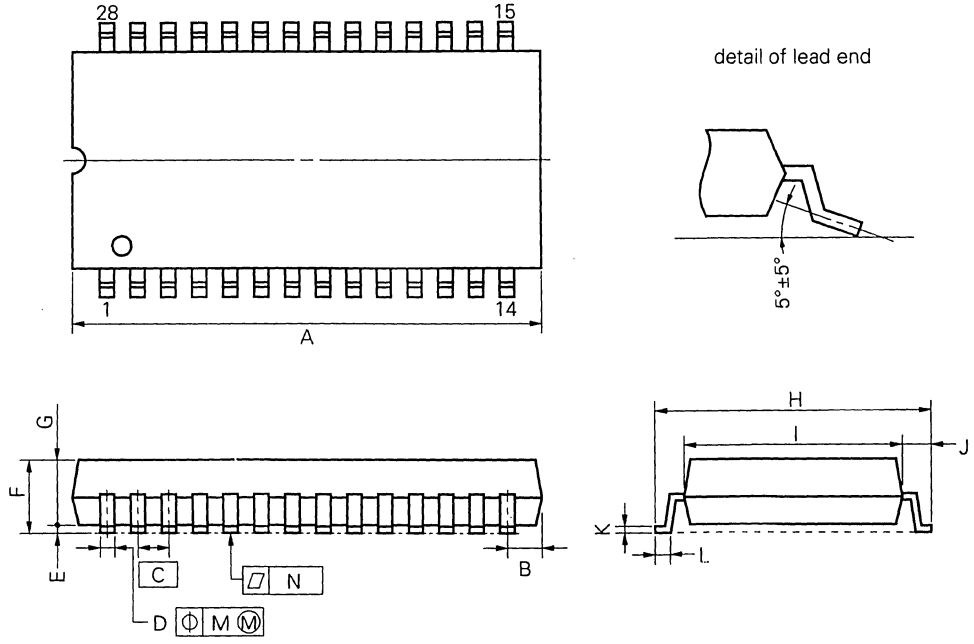
**Fig. 6 Double speed conversion timing diagram**




(In the last 192 words, the data before 1 field may be output.)

## 6. PACKAGE DRAWINGS

### 28 PIN PLASTIC SOP (450 mil)



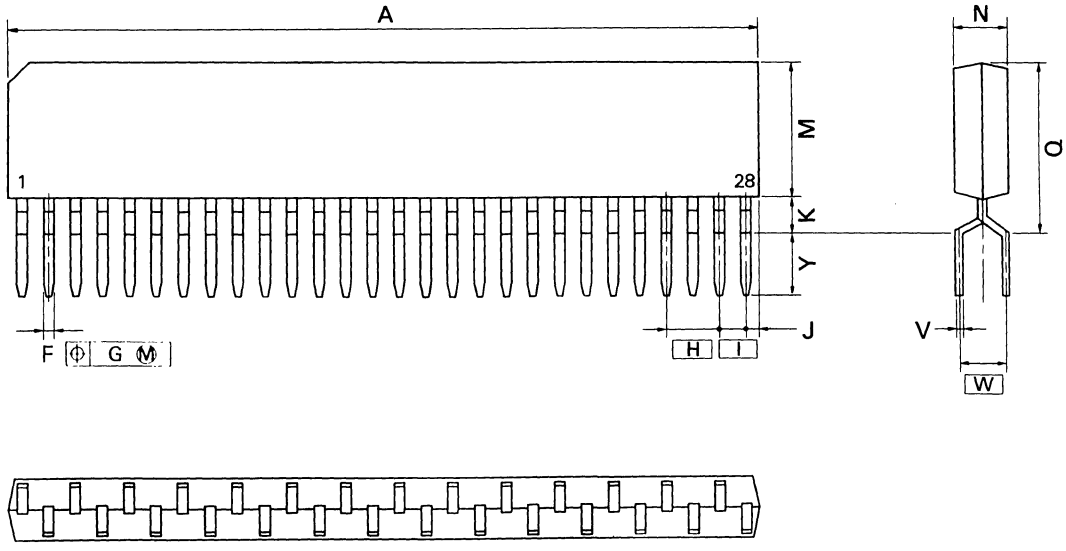

P28GM-50-450A2-2

**NOTE**

Each lead centerline is located within 0.12 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 19.05 MAX. | 0.750 MAX. |
| B | 1.27 MAX. | 0.050 MAX. |
| C | 1.27 (T.P.) | 0.050 (T.P.) |
| D | $0.40\pm0.10$ | $0.016^{+0.004}_{-0.005}$ |
| E | $0.1\pm0.1$ | $0.004^{+0.005}_{-0.004}$ |
| F | 3.0 MAX. | 0.119 MAX. |
| G | 2.55 | 0.100 |
| H | $11.8\pm0.3$ | $0.465^{+0.012}_{-0.013}$ |
| I | 8.4 | 0.331 |
| J | 1.7 | 0.067 |
| K | $0.15^{+0.10}_{-0.05}$ | $0.006^{+0.004}_{-0.002}$ |
| L | $0.7\pm0.2$ | $0.028^{+0.008}_{-0.009}$ |
| M | 0.12 | 0.005 |
| N | 0.10 | 0.004 |

## 28PIN PLASTIC ZIP (400mil)




P28V-254-400A

**NOTE**

Each lead centerline is located within 0.25 mm (0.010 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 36.83 MAX. | 1.450 MAX. |
| F | $0.5^{+0.1}$ | $0.020^{+0.004}_{-0.005}$ |
| G | $\phi$0.25 | $\phi$0.010 |
| H | 2.54 | 0.100 |
| I | 1.27 | 0.050 |
| J | 1.27 MAX. | 0.050 MAX. |
| K | 1.0 MIN. | 0.039 MIN. |
| M | 8.9 MAX. | 0.350 MAX. |
| N | $2.8^{\pm0.2}$ | $0.110^{+0.009}_{-0.008}$ |
| Q | 10.16 MAX. | 0.400 MAX. |
| V | $0.25^{+0.10}_{-0.05}$ | $0.010^{+0.004}_{-0.003}$ |
| W | 2.54 | 0.100 |
| Y | $3.3^{\pm0.5}$ | $0.130^{\pm0.02}$ |

## 7. RECOMMENDED SOLDERING CONDITIONS

The following conditions (see table below) must be met when soldering μPD42280.

For more details, refer to our document **"SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL"** (IEI-127).

Please consult with our sales offices in case other soldering process is used, or in case other soldering is done under different conditions.

**TYPE OF SURFACE MOUNT DEVICE**

**μPD42280GU: 28-pin plastic SOP (450 mil)**

| Soldering process | Soldering conditions | Symbol |
|---|---|---|
| Infrared ray reflow | Peak temperature of package surface: 235 °C or below,<br>Reflow time: 30 seconds or below (210 °C or higher),<br>Number of reflow processes: MAX. 2<br>Exposure limit [Note] : 3 days (20 hours pre-baking is required at 125 °C afterwards)<br>**[Remark]**<br>(1) Please start the second reflow process after the temperature, raised by the first reflow process, returns to normal.<br>(2) Please avoid removing the residual flux with water after the first reflow process. | IR35-203-2 |
| VPS | Peak temperature of package surface: 215 °C or below,<br>Reflow time: 40 seconds or below (200 °C or higher),<br>Number of reflow processes: MAX. 2<br>Exposure limit [Note] : 3 days (20 hours pre-baking is required at 125 °C afterwards)<br>**[Remark]**<br>(1) Please start the second reflow process after the temperature, raised by the first reflow process, returns to normal.<br>(2) Please avoid removing the residual flux with water after the first reflow process. | VP15-203-2 |
| Wave Soldering | Solder temperature: 260 °C or below,<br>Flow time: 10 seconds or below,<br>Temperature of pre-heat: 120 °C MAX. (Plastic surface temperature)<br>Number of flow process: 1<br>Exposure limit [Note] : 3 days (10 hours pre-baking is required at 125 °C afterwards) | WS60-203-1 |
| Partial heating method | Terminal temperature: 300 °C or below,<br>Time: 3 seconds or below (Per one side of the device). | — |

**Note** Exposure limit before soldering after dry-pack package is opened.
Storage conditions: 25 °C and relative humidity at 65 % or less.

**Caution Do not apply more than one soldering method at any one time, except for "Partial heating method".**

## TYPE OF THROUGH HOLE MOUNT DEVICE

### μPD42280V: 28-pin plastic ZIP (400 mil)

| Soldering process | Soldering conditions |
|---|---|
| Wave soldering<br>(For leads only) | Solder temperature: 260 °C or below,<br>Flow time: 10 seconds or below |
| Partial heating method | Terminal temperature: 300 °C or below,<br>Time: 3 seconds or below (Per one lead) |

**Caution Do not jet molten solder on the surface of package.**

# Rambus DRAM

NEC

MOS INTEGRATED CIRCUIT

# μPD488170L

## 18M-BIT Rambus DRAM
## 1M-WORD X 9-BIT X 2-BANK

### Description

The 18-Megabit Rambus™ DRAM (RDRAM™) is an extremely-high-speed CMOS DRAM organized as 2M words by 9 bits and capable of bursting up to 256 bytes of data at 2 ns per byte. The use of Rambus Signaling Logic (RSL) technology makes this 500 MHz transfer rate achievable while using conventional system and board design methodologies. Low latency is attained by using the RDRAM's large internal sense amplifier arrays as high speed caches.

RDRAMs are general purpose high-performance memory devices suitable for use in a broad range of applications including main memory, graphics, video, and any other application where high-performance and low cost are required.

Detailed information about product features and specifications can be found in the following document. Please make sure to read this document before starting design.

**Rambus DRAM user's manual (Reference Manual)** : IEU-1401

Rambus and RDRAM are trademarks of Rambus Inc.

### Features

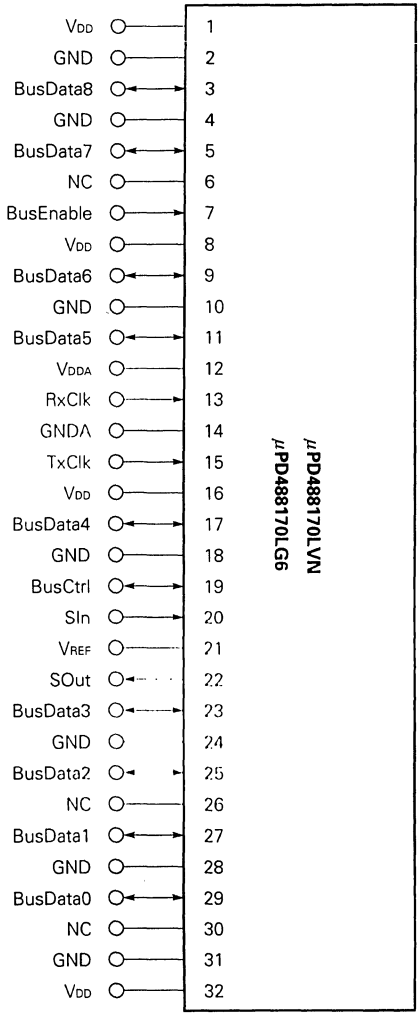
- Rambus Interface
- 500 MB/sec peak transfer rate per RDRAM
- RSL interface
- Synchronous protocol for fast block-oriented transfers
- Direct connection to Rambus ASICs, MPUs, and Peripherals
- 40 ns from start of read request to first byte; 2 ns per byte thereafter
- Features for graphics include random-access mode, write-per-bit and mask-per-bit operations
- Dual 2K-Byte sense amplifiers act as caches for low latency accesses
- Multiple power-saving modes
- On-chip registers for flexible addressing and timing
- Low pincount-only 15 active signals
- Standardized pinout across multiple generations of RDRAMs
- 3.3 volt operation

### Ordering Information

| Part Number | Clock Frequency | Operation Voltage | Package |
|---|---|---|---|
| μPD488170LVN-A50 | 250MHz | 3.3±0.15 V | 32-pin plastic SVP (11 × 25) |
| μPD488170LVN-A45 | 225MHz | 3.3±0.15 V | 32-pin plastic SVP (11 × 25) |
| μPD488170LG6-A50 | 250MHz | 3.3±0.15 V | 72/36-pin plastic SSOP type |
| μPD488170LG6-A45 | 225MHz | 3.3±0.15 V | 72/36-pin plastic SSOP type |

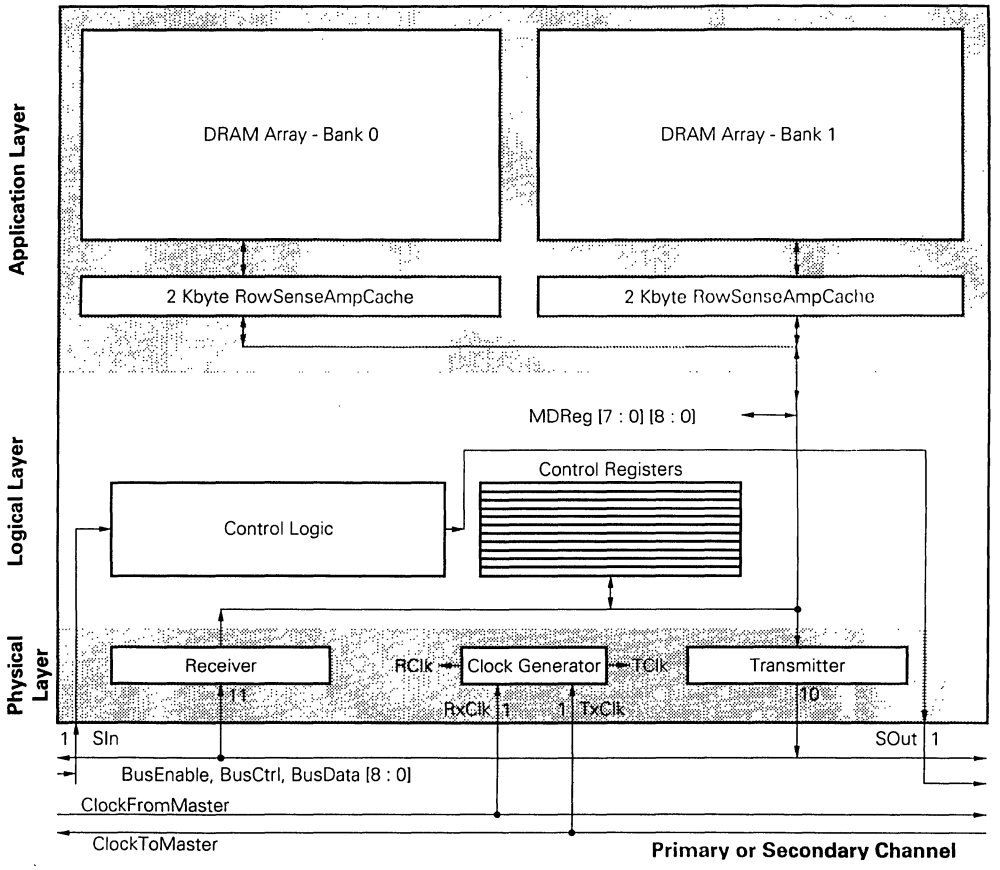
## Pin Configuration (Marking Side)

**32-pin plastic SVP (11 × 25)**
**72/36-pin plastic SSOP type**




| | |
|---|---|
| BusData 0 - BusData 8 | : Bus Data (Input/Output) |
| RxClk | : Receive Clock (Input) |
| TxClk | : Transmit Clock (Input) |
| $V_{REF}$ | : Logic Threshold Voltage (Input) |
| BusCtrl | : BusCtrl (Input/Output) |
| BusEnable | : BusEnable (Input) |
| $V_{DD}$, $V_{DDA}$ | : Power Supply |
| GND, GNDA | : Ground |
| SIn | : Serial Input (Input) |
| SOut | : Serial Output (Output) |
| NC | : No Connection |

## Block Diagram



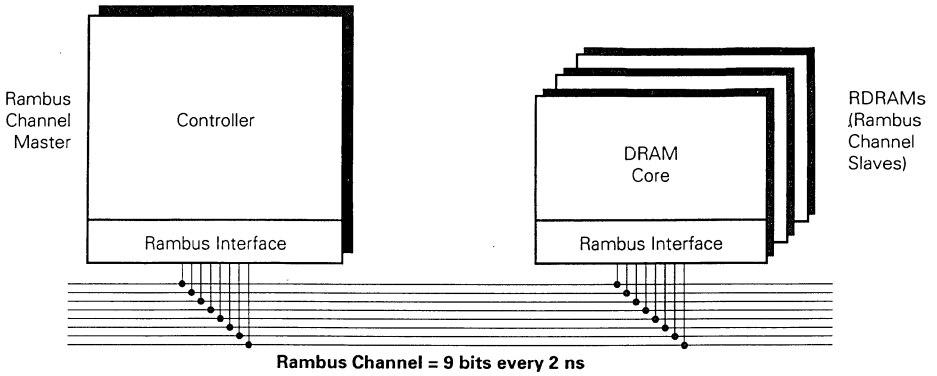
## 1. Pin Function

| Signal | I/O | Description |
|---|---|---|
| BusData [8:0] | I/O | Signal lines for request, write data, and read data packets. The request packet contains the address, operation codes, and the count of the bytes to be transferred. This is a low-swing, active-low signal referenced to $V_{REF}$. |
| RxClk | I | Receive clock. Incoming request and write data packets are aligned to this clock. This is a low-swing, active-low signal referenced to $V_{REF}$. |
| TxClk | I | Transmit clock. Outgoing acknowledge and read packets are aligned with this clock. This is a low-swing, active-low signal referenced to $V_{REF}$. |
| $V_{REF}$ | I | Logic threshold voltage for low swing signals. |
| BusCtrl | I/O | Control signal to frame packets, to transmit part of the operation code, and to acknowledge requests. Low-swing, active-low signal referenced to $V_{REF}$. |
| BusEnable | I | Control signal to enable the bus. Long assertions of this signal will reset all devices on the Channel. This is a low-swing, active-low signal referenced to $V_{REF}$. |
| $V_{DD}$, $V_{DDA}$ | | +3.3 V power supply. $V_{DDA}$ is a separate analog supply. |
| GND, GNDA | | Circuit ground. GNDA is a separate analog ground. |
| SIn | I | Initialization daisy chain input. TTL levels. Active high. |
| SOut | O | Initialization daisy chain output. TTL levels. Active high. |

## 2. Rambus System Overview

A typical Rambus memory system has three main elements: the Rambus Channel, the RDRAMs, and a Rambus Interface on a controller. The logical representation of this is shown in the following figure.

**Figure 2-1. Logical Representation**

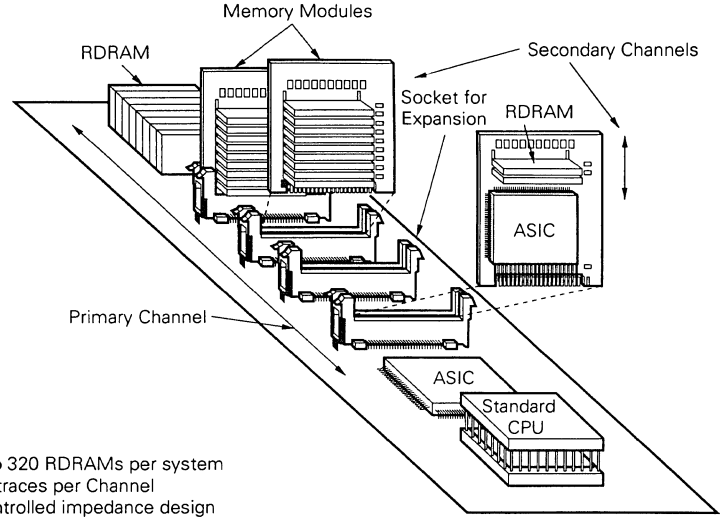



The Rambus Channel is a synchronous, high-speed, byte-wide bus that is used to directly connect Rambus devices together. Using only 13 high-speed signals, the Channel carries all address, data, and control information to and from devices through the use of a high level block-oriented protocol.

The Rambus Interface is implemented on both master and slave devices. Rambus masters are the only devices capable of generating transaction requests and can be ASIC devices, memory controllers, graphics engines, peripheral chips, or microprocessors. RDRAMs are slave devices and only respond to requests from master devices.

The following figure shows a typical physical implementation of a Rambus system. It includes a controller ASIC that acts as the Channel master and a base set of RDRAMs soldered directly to the board. An RSocket™ is included on the Channel for memory upgrade using RModule™ expansion cards.

**Figure 2-2. A Rambus System Example**

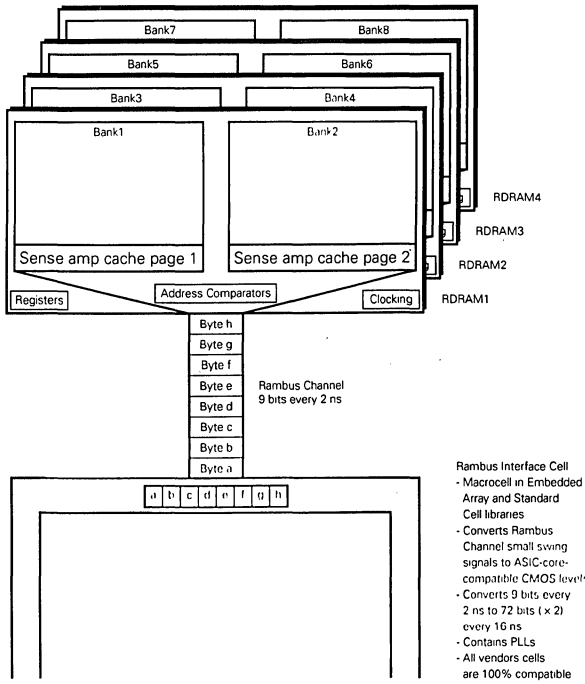


### 3. Rambus Signaling Logic

RSL technology is the key to attaining the high data rates available in Rambus systems. By employing high quality transmission lines, current-mode drivers, low capacitive loading, low-voltage signaling, and precise clocking, systems reliably transfer data at 2 nanosecond intervals on a Rambus Channel with signal quality that is superior to TTL or GTL-based interfaces.

All Rambus Interfaces incorporate special logic to convert signals from RSL to CMOS levels for internal use. In addition, these interfaces convert the Channel data rate of one byte every 2 nanoseconds to an internal data rate of 8 bytes every 16 nanoseconds as shown in the following figure. Although the bandwidth remains the same, the use of a wide internal bus eases internal timing requirements for chip designers.

**Figure 3-1. Converting the Channel Data Rate**



## 4. Register Space Map

The following table summarizes the registers included in all 18M RDRAMs.

**Table 4-1. Registers Space Map**

| Register Name | Adr[20:10] | Adr[9:2] | Register Number |
|---|---|---|---|
| Device Type[3:0][8:0] | xx...xx | 00000000 | 0 |
| DeviceId[3:0][8:0] | xx...xx | 00000001 | 1 |
| Delay[3:0][8:0] | xx...xx | 00000010 | 2 |
| Mode[3:0][8:0] | xx...xx | 00000011 | 3 |
| RefInterval[3:0][8:0] | xx...xx | 00000100 | 4 |
| RefRow[3:0][8:0] | xx...xx | 00000101 | 5 |
| RasInterval[3:0][8:0] | xx...xx | 00000110 | 6 |
| MinInterval[3:0][8:0] | xx...xx | 00000111 | 7 |
| AddressSelect[3:0][8:0] | xx...xx | 00001000 | 8 |
| DeviceManufacturer[3:0][8:0] | xx...xx | 00001001 | 9 |
| Undefined | xx...xx | 0000101x | 10-11 |
| Undefined | xx...xx | 000011xx | 12-15 |
| Undefined | xx...xx | 0001xxxx | 16-31 |
| Undefined | xx...xx | 001xxxxx | 32-63 |
| Undefined | xx...xx | 01xxxxxx | 64-127 |
| Row[3:0][8:0] | xx...xx | 10000000 | 128 |
| Undefined | xx...xx | 10000001 | 129 |
| Undefined | xx...xx | 1000001x | 130-131 |
| Undefined | xx...xx | 100001xx | 132-135 |
| Undefined | xx...xx | 10001xxx | 136-143 |
| Undefined | xx...xx | 1001xxxx | 144-159 |
| Undefined | xx...xx | 1010xxxx | 160-175 |
| Undefined | xx...xx | 1011xxxx | 176-191 |
| Undefined | xx...xx | 1100xxxx | 192-207 |
| Undefined | xx...xx | 1101xxxx | 208-223 |
| Undefined | xx...xx | 1110xxxx | 224-239 |
| Undefined | xx...xx | 1111xxxx | 240-255 |

**(1) Device Type Register**

This register specifies RDRAM configuration and size.

**(2) DeviceId Register**

This register specifies RDRAM base address.

**(3) Delay Register**

This register specifies RDRAM programmable CAS delay values.

**(4) Mode Register**

This register specifies RDRAM programmable output drive current.

**(5) RefInterval Register**

This register specifies RDRAM programmable refresh interval.

RefInterval Register is used to time the refresh interval for devices which require refresh.

**(6) RefRow Register**

This register specifies RDRAM refresh row and bank address.

The RefRow register contains read-write fields. It is used to keep track of the bank and row being refreshed. Normally this register is only read or written for testing purposes. The fields are aliased in the following way:

RowField[7:1] equals RefRow[0][7:1]
RowField[9:8] equals RefRow[2][1:0]
BankField[3] equals RefRow[1][3]

**(7) RasInterval Register**

This register specifies RDRAM programmable RAS delay values. The RasInterval Register contains four write-only fields. When a rowmiss occurs, or when a row is being refreshed during a burst refresh operation, it is necessary for the control logic of an RDRAM to count the appropriate number of clock cycles ($t_{CYCLE}$) for four intervals. This is done with a counter which is loaded successively with three values from the RasInterval Register. This counter is not available for read access and must be tested indirectly.

**(8) MinInterval Register**

This register specifies RDRAM refresh and powerdown control.

This register provides the minimum values for three time intervals for framing packets.

The time intervals are specified in clock cycle ($t_{CYCLE}$) units.

**Caution MinInterval Register[3][2] = 0 is necessary. Because, 18M RDRAM cannot accept Power Down request.**

**(9) AddressSelect Register**

This register specifies RDRAM address mapping.

**(10) DeviceManufacturer Register**

This register specifies RDRAM manufacturer information.

This register specifies the manufacturer of the device. Additional bits are available for manufacturer specific information, e.g. stepping or revision numbers.

**(11) Row Register**

This register specifies RDRAM current sensed row in each bank.

The detailed functional description is provided in RDRAM Reference Manual.

## 5. Packet Formation

### 5.1 Packet Summary

The following table summarizes the transmit/receive functionality of the two RDRAM types for the different packet classes.

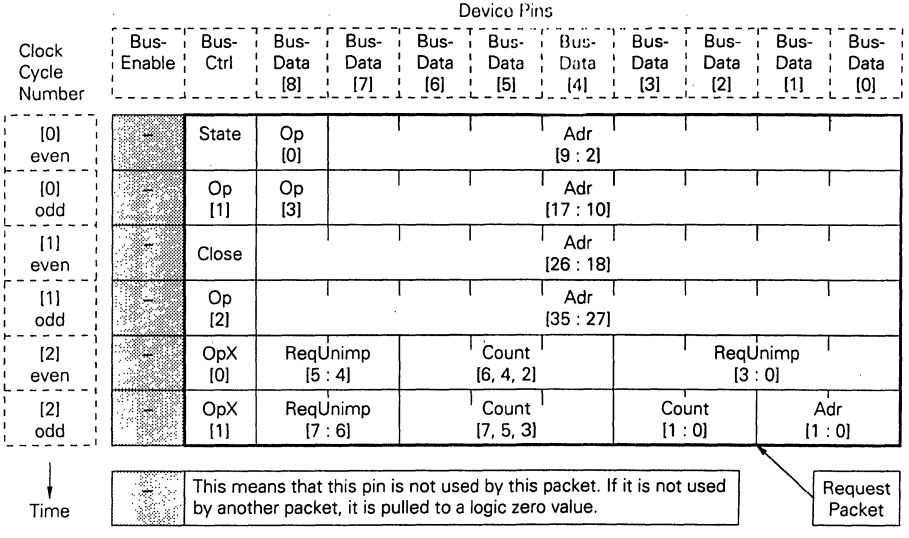
**Table 5-1. Transmitting/Receiving Devices for Packet Types**

| Packet Type | Initiating Devices | μPD488170L |
|---|---|---|
| Request Packet | Transmit | Receive |
| Acknowledge Packet | Receive | Transmit |
| Read Data Packet | Receive | Transmit |
| Write Data Packet | Transmit | Receive |
| Serial Address Packet | Transmit | Receive |
| Serial Control Packet | Transmit | Receive |
| Serial Mode Packet | Transmit | Receive |

### 5.2 Request Packet

The request packet format is shown in the following figure.

**Figure 5-1. Request Packet Format**




The vertical axis in all packet figures in the following sections shows time in units of clock cycles, with each clock cycle broken into even and odd bus ticks. The timing is relative, measured from the beginning of the packet.

### 5.2.1 Start Field

A device should start framing a request packet when it sees this bit asserted to a logical one and it is not looking for an acknowledge packet nor framing an earlier request packet.

### 5.2.2 Op[3:0], OpX[1:0] Fields

The command opcode also determines which packets (in addition to the request packet) will form the transaction. A detailed functional description of the actions that an RDRAM takes for each implemented command is provided in **"Rambus DRAM user's manual (Reference Manual)"**. The following table summarizes the functionality of each subcommand:

**Table 5-2. Subcommand Summary**

| SubCommand | Description |
|---|---|
| Rseq | Read sequential data from memory space. |
| Rnsq | Read non-sequential (random-access) data from memory space. |
| Wseq | Write sequential data to memory space. |
| Wnsq | Write non-sequential (random-access) data to memory space. |
| Wbns | Write non-sequential (random-access) data to memory space with non-contiguous byte masking. |
| Npb | Write data is from data packet. There is no bit mask. |
| Dpb | Write data is from data packet. The bit mask is in the MDReg. |
| Bpb | Write data is from data packet. The bit mask is also from the data packet. |
| Mpb | Write data is from MDReg. The bit mask is from the data packet. |
| Rreg | Read sequential data from register space. |
| Wreg | Write sequential data to register space. |
| WregB | Broadcast write with no Okay acknowledge permitted. |
| Alt | Alternate command (same function as the primary command - intended for use in future shared memory multiprocessor systems). |

The memory read commands are formed using the Rseq and Rnsq subcommands to select sequential or nonsequential (random) access. The Alt and "null" subcommands select between two equivalent command sets ("null" means no subcommand). The "Alt" subcommands are reserved for use in future shared memory multiprocessor systems.

- RrrrAaa  Rrrr = {Rseq, Rnsq}
  Aaa = {Alt, null}

The following table summarizes the available write commands and shows how they are formed from a 3×4 matrix of the Wwww and Bbb subcommands. The Alt and "null" subcommands have the same meaning as in the memory read commands.

- WwwwBbbAaa Wwww = {Wseq, Wnsq, Wbns}
  Bbb = {Npb, Dpb, Bpb, Mpb}
  Aaa = {Alt, null}

**Table 5-3. Write Commands**

| Bbb subcommand | Wwww subcommands | | |
|---|---|---|---|
| | Wseq (seqential-access with contiguous byte masking) | Wnsq (non-sequential-access) | Wbns (non-sequential-access with non-contiguous-byte-masking) |
| Npb | WseqNpb | WnsqNpb | WbnsNpb |
| Dpb | WseqDpb | WnsqDpb | WbnsDpb |
| Mpb | WseqMpb | WnsqMpb | WbnsMpb |
| Bpb | WseqBpb | WnsqBpb | Not implemented |

There are three Wwww subcommands. They control the accessing pattern and the use of non-contiguous byte masking.

- Wseq - octbyte blocks in the RDRAM core are accessed in sequential (ascending little-endien) address order. Contiguous byte masking is controlled with the Adr[2:0] and Count[2:0] fields of the request packet.

- Wnsq - octbyte blocks in the RDRAM core are accessed in non-sequential address order. The addresses for the octbyte blocks within the sensed row come from serial address packets which are received on the BusEnable pin.
  The address order is arbitrary.

- Wbns - octbyte blocks in the RDRAM core are accessed in non-sequential address order, as in the Wnsq subcommand. In addition, byte masks are transmitted with the write data, permitting arbitrary non-contiguous byte masking of this write data. The bytemask octbytes are not included in the total octbyte transfer count ; i.e. a Count[7:3] field of 31 implies 4 bitmask octbytes and 32 write data octbytes, for a data packet size of 36 octbytes.

There are four Bbb subcommands. They select the type of bit masking to be applied to the write data.

- Npb (no-per-bit) - There is no bit mask applied to the write data. The MDReg is not used or modified.

- Dpb (data-per-bit) - The MDReg is used as a bit mask, the write data comes from the data packet. The same bit mask is used for each octbyte. This is also called persistent bit masking. The MDReg is not modified.

- Mpb (mask-per-bit) - The bit mask comes from the data packet, the write data comes from the MDReg. The same data is used for each octbyte. This is also called color masking. The MDReg is not modified.

- Bpb (both-per-bit) - The bit mask and the write data come from the data packet. The MDReg is not used, but is modified as a side effect (the WwwwBpbAaa commands are used to load the MDReg for the WwwwDpbAaa and WwwwMpbAaa commands). This is also called non-persistent bit masking.
  The bitmask octbytes are included in the total octbyte transfer count ; i.e. a Count[7:3] field of 31 implies 16 bitmask octbytes and 16 write data octbytes.

#### 5.2.2.1 Op[3:0] and OpX[1:0] Fields for 18M RDRAM

The Op and OpX fields are summarized in the following table.

**Table 5-4. Op[3:0] and OpX[1:0] Fields - Command Encodings**

| Op[3:0] | OpX[1:0] = 00 | OpX[1:0] = 01 | OpX[1:0] = 10 | OpX[1:0] = 11 |
|---|---|---|---|---|
| 0000 | Rseq | Rnsq | Rsrv | Rsrv |
| 0001 | RseqAlt | RnsqAlt | Rsrv | Rsrv |
| 0010 | Rsrv | Rsrv | Rsrv | Rsrv |
| 0011 | Rsrv | Rsrv | Rsrv | Rsrv |
| 0100 | WseqNpb | WseqDpb | WseqBpb | WseqMpb |
| 0101 | WseqNpbAlt | WseqDpbAlt | WseqBpbAlt | WseqMpbAlt |
| 0110 | Rreg | Rsrv | Rsrv | Rsrv |
| 0111 | Wreg | Rsrv | Rsrv | Rsrv |
| 1000 | WnsqNpb | WnsqDpb | WnsqBpb | WnsqMpb |
| 1001 | WnsqNpbAlt | WnsqDpbAlt | WnsqBpbAlt | WnsqMpbAlt |
| 1010 | Rsrv | Rsrv | Rsrv | Rsrv |
| 1011 | Rsrv | Rsrv | Rsrv | Rsrv |
| 1100 | WbnsNpb | WbnsDpb | Rsrv | WbnsMpb |
| 1101 | WbnsNpbAlt | WbnsDpbAlt | Rsrv | WbnsMpbAlt |
| 1110 | Rsrv | Rsrv | Rsrv | Rsrv |
| 1111 | WregB | Rsrv | Rsrv | Rsrv |

The command opcode determines whether the other fields of the request packet are implemented (Imp) or unimplemented (Unimp). This is summarized in the table below.

**Table 5-5. 18M RDRAM Request Packet Fields - Imp or Unimp**

| Command | Adr[35:3] | Adr[2] | Adr[1:0] | Count[7:3] | Count[2] | Count[1:0] | Close | ReqUnimp[7:0] |
|---|---|---|---|---|---|---|---|---|
| Rsrv | Unimp(0.0) | Unimp(0) | Unimp(00) | Unimp(0.0) | Unimp(1) | Unimp(11) | Unimp(0) | Unimp(0.0) |
| Rseq | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| Rnsq | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| RseqAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| RnsqAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqNpb | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqDpb | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqBpb | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqMpb | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqNpbAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqDpbAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqBpbAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqMpbAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| Rreg | Imp | Imp | Unimp(00) | Unimp(0.0) | Unimp(1) | Unimp(11) | Unimp(0) | Unimp(0.0) |
| Wreg | Imp | Imp | Unimp(00) | Unimp(0.0) | Unimp(1) | Unimp(11) | Unimp(0) | Unimp(0.0) |
| WnsqNpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqDpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqBpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqMpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqNpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqDpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqBpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqMpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsNpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsDpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsMpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsNpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsDpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsMpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WregB | Imp | Imp | Unimp(00) | Unimp(0.0) | Unimp(1) | Unimp(11) | Unimp(0) | Unimp(0.0) |

### 5.2.3 Adr[35:0] Field

The Adr field is used as either a memory or register space address depending upon the OP[3:0] and OpX[1:0] fields. Devices extract a portion of the Adr field to match against their DeviceId register (IdMatch), thus selecting the device to which the request is directed. The remainder of the Adr field accesses the desired region of the device's memory or register space. The memory read and write commands and the Rreg and Wreg commands will only take place if there is an IdMatch. The IdMatch criteria is ignored for the WRegB commands, with all responding devices performing the required actions.

The Rambus protocol uses quadbyte resolution in the data packet for register space read and write commands; i.e. one quadbyte is the smallest data item that may be transferred, and all transfers are an integral number of quadbytes. The Adr[35:2] field is the quadbyte address. The Adr[1:0] field is Unimp for these commands, and should be driven with "00" by initiating devices.

The Rambus protocol uses octbyte resolution in the data packet for memory space read and write commands; i.e. one octbyte is the smallest data item that may be transferred, and all transfers are an integral number of octbytes. The Adr[35:3] field is the octbyte address.

Some commands use the Adr[2:0] field to specify contiguous byte masking. Refer to **"Rambus DRAM user's manual (Reference Manual)"**.

### 5.2.4 Count[7:0] Field

The following table summarizes the transfer count ranges for 18M RDRAMs:

**Table 5-6. Transfer Count Summary**

| Count Range | μPD488170L |
|---|---|
| Maximum count for memory space | 32 octbytes |
| Minimum count for memory space | 1 octbyte |
| Maximum count for register space | 1 quadbyte |
| Minimum count for register space | 1 quadbyte |

Register space read and write commands use a transfer count of one quadbyte, regardless of the Count[7:0] field value.

Memory space read and write commands specify the number of octbytes to be transferred with the Count[7:3] field. An offset-by-one-encoding is used so that "00000" specifies one octbyte, "00001" specifies two octbytes, and so on up to "11111" which specifies thirty-two octbytes. The transfer count does include the octbytes containing bitmasks (for commands using the Bpb subcommand). The transfer count does not include the octbytes containing non-contiguous ByteMasks (for commands using the Wbns subcommand).

Some commands use the Count[2:0] field to specify contiguous byte masking. Refer to **"Rambus DRAM user's manual (Reference Manual)"**.

Memory space transactions to RDRAMs are not allowed to cross internal row address boundaries within the device. Attempts to do so have Undef (undefined) results. These row boundaries are at 2kbyte intervals for 18M RDRAMs.

### 5.2.5 Adr[2:0] and Count[2:0] Fields for Contiguous Byte Masking

An initiating device wishing to transfer an arbitrary number of contiguous bytes to a starting address on an arbitrary byte boundary may do so with the Adr[2:0] and Count[2:0] fields for some of the commands. These commands include:

- RrrrAaa
- WseqBbbAaa

The transfer count and starting address are given by:

- MasterCount[7:0] specifies the number of bytes which the master device wishes to transfer.
- Adr[35:0] specifies the starting byte address (this is the same as the Adr[35:0] field in the request packet)

Where the convention used by the initiating device for the count is that Master-Count[7:0] = "00000000" means one byte, MasterCount[7:0] = "00000001" means two bytes and MasterCount[7:0] = "11111111" means 256 bytes (an offset-by-one encoding; the data block count is equal to MasterCount[7:0]+1).

The initiating device converts this internal count value into a value for the request packet with the following formula. Little-endien byte addressing is used for specifying bytes within octbytes.

$$\text{Count}[7:0] = \text{Adr}[2:0] + \text{MasterCount}[7:0] \qquad \text{(Eq 5-1)}$$

Where "+" denotes unsigned integer addition of two bit fields (short fields are zero-extended on the left). If the value of Adr[2:0] + MasterCount[7:0] is greater than 255 (it may be as much as 262), then the initiating device must break the request into two transactions.

The Adr[2:0] and Count[2:0] field generate masks for individual bytes within an octbyte. The Adr[35:3] and Count[7:3] field have the octbyte resolution previously described. The following tables show how the byte masks are generated. In the case of memory read transactions, the byte masks that are generated do not affect the data that is returned by the RDRAM; all data bytes in the first and last octbytes are returned in the read data packet.

In the case of memory write transactions, ByteMaskLS[7:0] applies to the first octbyte at Mem[AV][7:0][8:0]. Byte MaskMS[7:0] applies to the last octbyte at Mem[AV+CV][7:0][8:0]. All intermediate octbytes use a byte mask of 11111111 (a one means the byte is written, a zero means it is not). Here AV is the value of the Adr[35:3] field when interpreted as an unsigned, 33 bit integer, and CV is the value of the Count[7:3] field when interpreted as an unsigned, 5 bit interger. If the Count[7:3] is "00000" (one octbyte), the ByteMaskLS[7:0] and ByteMaskMS[7:0] masks are logically 'anded' together to give the effective byte mask.:

**Table 5-7. Adr[2:0] to ByteMaskLS[7:0] Encoding**

| Adr[2:0] | ByteMaskLS[7:0] | Adr[2:0] | ByteMaskLS[7:0] |
|---|---|---|---|
| 000 | 11111111 | 100 | 11110000 |
| 001 | 11111110 | 101 | 11100000 |
| 010 | 11111100 | 110 | 11000000 |
| 011 | 11111000 | 111 | 10000000 |

**Table 5-8. Count[2:0] to ByteMaskMS[7:0] Encoding**

| Count[2:0] | ByteMaskMS[7:0] | Count[2:0] | ByteMaskMS[7:0] |
|---|---|---|---|
| 000 | 00000001 | 100 | 00011111 |
| 001 | 00000011 | 101 | 00111111 |
| 010 | 00000111 | 110 | 01111111 |
| 011 | 00001111 | 111 | 11111111 |

The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

### 5.2.6 Close Field

The Close field causes a currently accessed row to be explicitly restored (written back to the core if its Dirty flag for that bank is set) after the current access has completed. This reduces the latency of a subsequent access to a different row of memory space in the same bank of that device; i.e. the $t_{ReterSensedClean}$ Nack timing is used rather than the $t_{RetrySensedDirty}$ timing.

**Table 5-9. Close Field Encodings**

| Close Field | Description |
|---|---|
| 0 | Don't restore the currently accessed row of memory. It may be left sensed and either clean or dirty. |
| 1 | Restore the currently accessed row of memory if its Dirty flag is set. It will be left sensed and clean. |

### 5.2.7 ReqUnimp[7:0] Fields

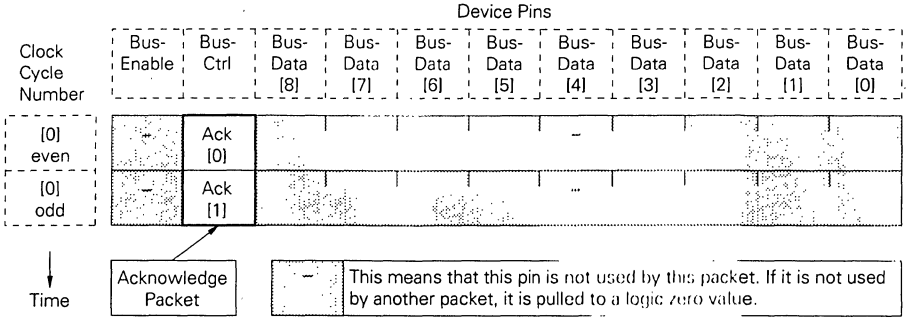
These fields are unimplemented (Unimp) in the request packet. They should be driven as zeroes by initiating devices which satisfy this Version of the Rambus protocol.

Responding devices which satisfy this Version of the Rambus protocol should ignore these fields and process the request according to the content of the Start, Op[3:0], OpX[1:0], Adr[35:0], Count[7:0], and Close fields. The protocol Version number of a device is held in the DeviceType register.

## 5.3 Acknowledge Packet

The Ack[1:0] field carries the acknowledge encoding from the responding device(s) to the initiating device and any other listening devices. The following figure shows the format of the acknowledge packet.

**Figure 5-2. Acknowledge Packet Format**




The following table summarizes the four combinations of the Ack[1:0] field. The Ack3 combination is Undef. The Okay combination indicates that the read or write access to the specified space will take place.

When a responding device acknowledges a request with a Nack, then there will be no immediate change in the state of the device's memory space or register space. The responding device will take the appropriate steps to make the requested region of memory or register space accessible when the initiating device makes a subsequent request. The initiating device will need to wait some device-dependent length of time until the requested region is available.

There are three possible reasons for an RDRAM to respond with Nack. They are summarized below. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

- $t_{PostMemWriteDelay}$ or $t_{PostRegWriteDelay}$ violation
- RowMiss (this causes a delay of $t_{RetrySensedClean}$ or $t_{RetrySensedDirty}$)
- ongoing refresh (this causes a delay of up to $t_{RetryRefresh}$)

**Table 5-10. Ack[1:0] Encodings**

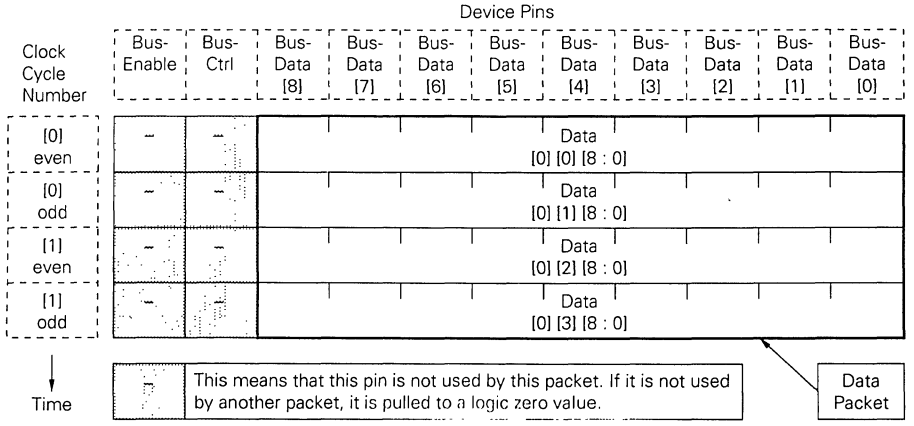
| Commands allowed to use the Ack Combination | Ack [1:0] | Name | Description | Spec Undef |
|---|---|---|---|---|
| All commands | 00 | Nonexistent | Indicates passive acceptance of the request (WregB), or indicates that the addressed device did not respond (all other commands). | Spec |
| All commands but WregB | 01 | Okay | Indicates that the request was accepted by the addressed by the addressed (responding) device. | Spec |
| All commands | 10 | Nack | Indicates that the request could not be accepted because the state of the responding device prevented an access at the fixed timing slot. | Spec |
| All commands but WregB | 11 | Ack3 | This should not be returned by this responding device. Initiating devices will, when presented with this combination, have an undefined response. | Undef |

## 5.4 Data Packet

The following figure shows the format of a data packet for register space read and write commands. It consists of 1 quadbyte driven on the BusData[8:0] wires for RDRAMs.

Other responding devices may support data packet lengths longer than one quadbyte.

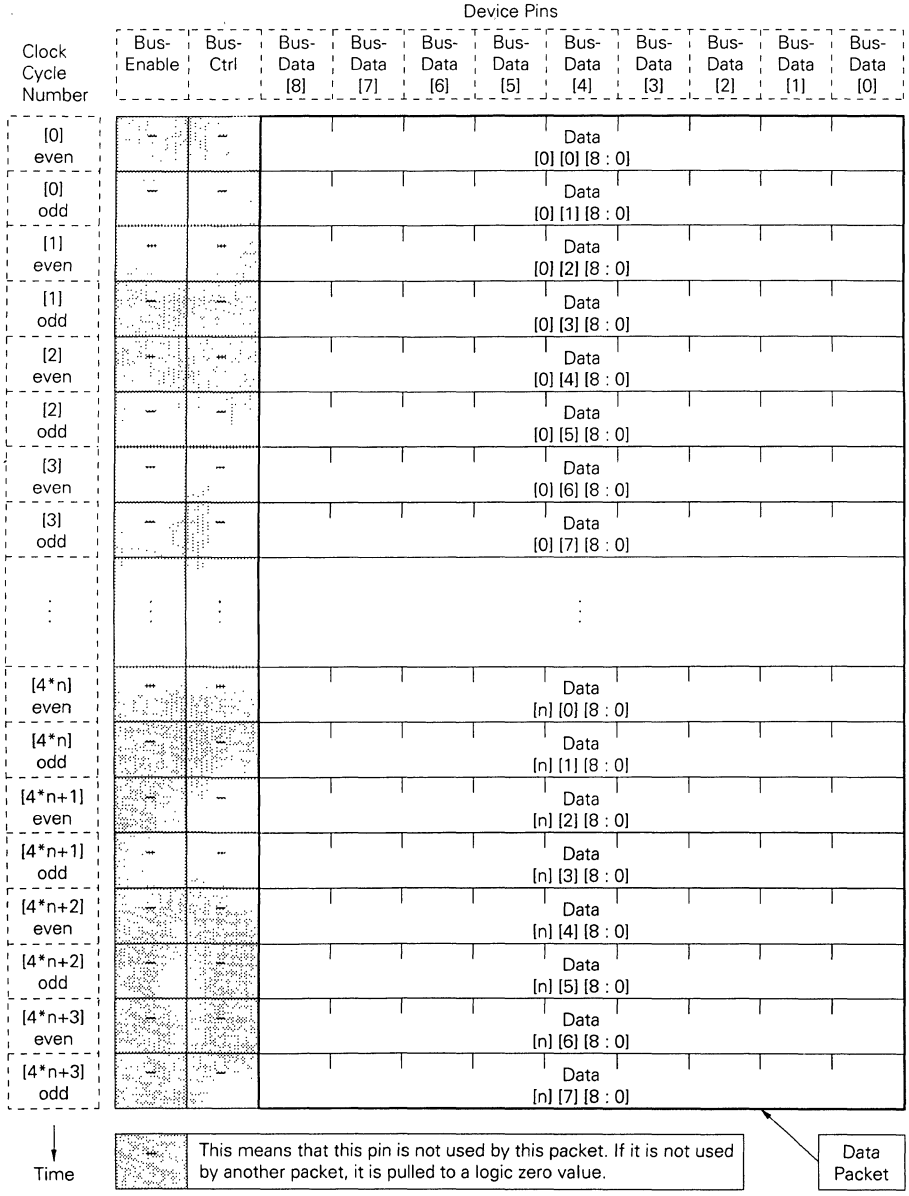
**Figure 5-3. Data Packet Format (Register Space)**




The following figure shows the format of a data packet for memory space read and write commands. For most of these commands, it consists of 1 to 32 octbytes driven on the BusData[8:0] wires. In the figure, "n" is either the CV value (if the transaction is allowed to complete) or the last count value (if the transaction is terminated prematurely by the serial control packet). "CV" is the value of the Count[7:3] field when interpreted as an unsigned, 5 bit integer.

The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 5-4. Data Packet Format (Memory Space)**



### 5.5 Serial Address Packet Format

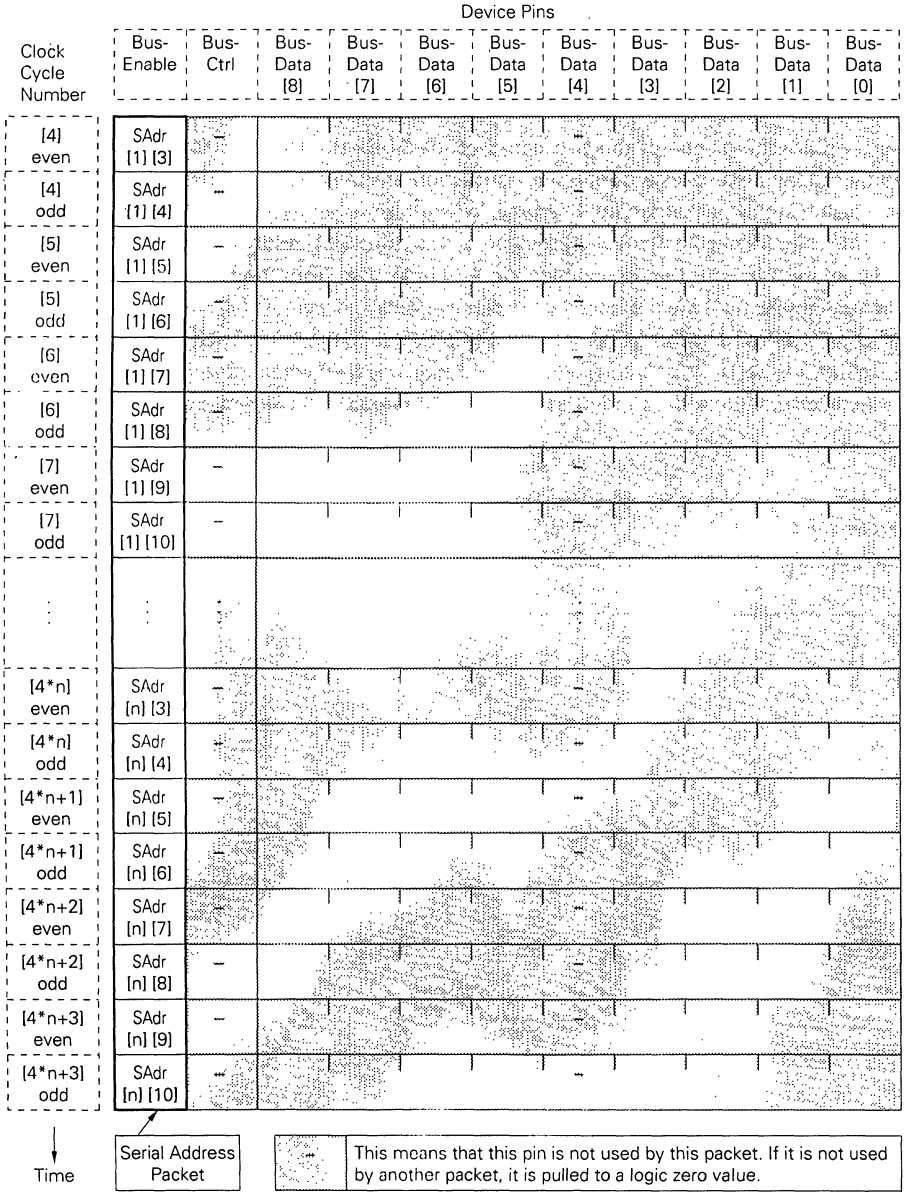
The serial address packet is transmitted by the initiating device and received by the responding devices. It provides eight low-order address bits for each octbyte which is accessed in memory space (a non-sequential or random-access transfer). These eight address bits are transferred serially on the BusEnable pin of the RDRAM, and are thus called a serial address. Each eight bit serial address accesses an octbyte of data within the RowSenseAmpCache of one of the two banks of the RDRAM. The complete set of serial addresses transmitted by the initiating device during the transaction are referred to as a serial address packet. The commands which use this packet are the RnsqAaa, WnsqBbbAaa, and WbnsBbbAaa classes of commands.

The high order bits for each octbyte are provided by the Adr[35:11] address bits from the request packet. The low-order address bits for the first octbyte are Adr[10:3], also from the request packet. The low-order address bits for octbytes [n:1] are provided by the serial address packet. As before, "n" is either the CV value (if the transaction is allowed to complete) or the last count value (if the transaction is terminated prematurely by the serial control packet). "CV" is the value of the Count[7:3] field when interpreted as an unsigned, 5 bit integer. The detailed functional description is provided in **"Rembus DRAM user's manual (Reference Manual)"**.

**Table 5-11. Serial Address Fields (i = n:1)**

| Serial Address Field | Description | Unimp Imp |
|---|---|---|
| SAdr[i][10:3] | Low-order address bits for each octbyte. | Imp |

**Figure 5-5. Serial Address Packet Format**



### 5.5.1 Serial Control Packet Format

The serial control packet is transmitted by the initiating device and received by the responding devices. It provides for the early termination of a memory space read or write transaction (before the specified data count in the Count[7:3] field has elapsed). It consists of eight bits transferred serially on the BusCtrl pin of the device, thus it is referred to as a serial control packet. The eight bits have the same timing alignment as the serial address packet. The commands which use this packet are all of those which access memory space. The register read and write commands do not use the serial control packet. The 18M RDRAM implements this packet.

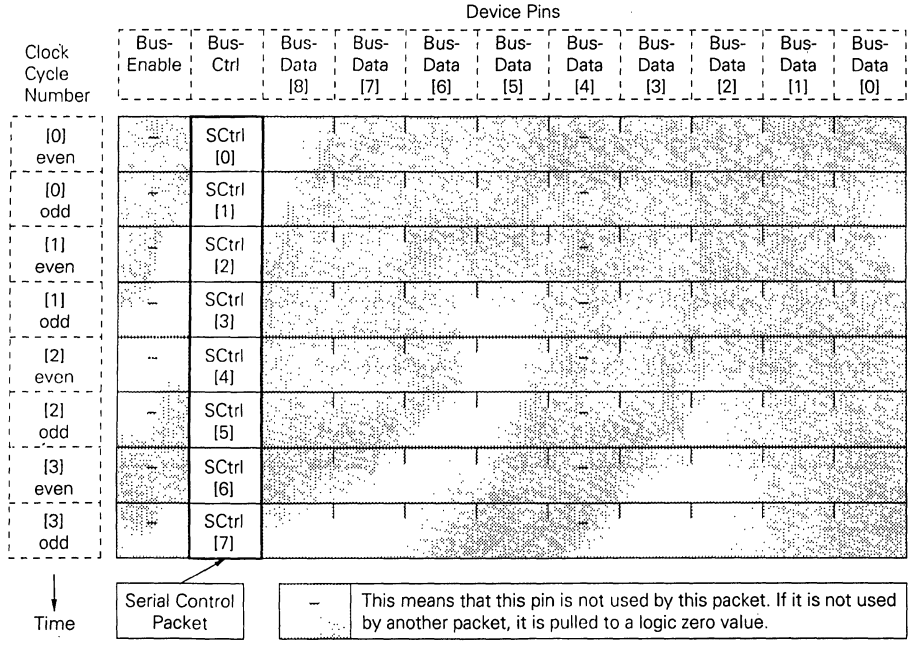
The termination occurs on octbyte data packet boundaries. The next figure shows the format of the serial control packet. The following table summarizes the function of the bits within the serial control packet. Note that the bits in the even bus ticks must be zero in order for framing to work properly (otherwise, one of these bits would be interpreted as the Start bit of a new request packet). The SCtrl[5] bit is used to control termination, and the other three odd bus tick bits are unimplemented.

**Table 5-12. Serial Control Fields**

| Serial Control Fields | Description | Unimp Imp |
|---|---|---|
| SCtrl[0] | This bit must be a zero due to framing requirements. | Imp |
| SCtrl[1] | unimplemented | Unimp(0) |
| SCtrl[2] | This bit must be a zero due to framing requirements. | Imp |
| SCtrl[3] | unimplemented | Unimp(0) |
| SCtrl[4] | This bit must be a zero due to framing requirements. | Imp |
| SCtrl[5] | 0 means don't terminate the current access.<br>1 means terminate the current access. | Imp |
| SCtrl[6] | This bit must be a zero due to framing requirements. | Imp |
| SCtrl[7] | unimplemented | Unimp(0) |

If a memory read transaction (RrrrAaa) is terminated by asserting the SCtrl[5] bit to a logical one, the data octbyte with which it is associated is not transmitted by the responding device. The initiating device may start a new transaction once the transmission of the read data packet has ceased. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

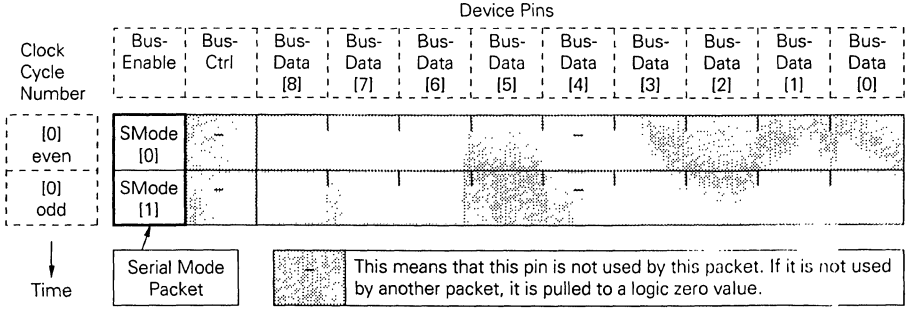
**Figure 5-6. Serial Control Packet Format**



### 5.5.2 Serial Mode Packet Format

The serial mode packet transmitted by initiating devices, and received by responding device. Its format is shown in the following figure.

**Figure 5-7. Serial Mode Packet Format**




The serial mode packet modifies the state of the Count00[7:0] and Count11[7:0] counters.

These counters cause operating mode transitions when they reach special values. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

A serial mode packet with the SMode[1:0] field set to 00 is the default. Most transitions are caused by blocks of sequential serial mode packets, each with the SMode[1:0] field set to 11. The serial mode packets should never set SMode[1:0] field to 01 or 10. This is because in some of the operating modes, the clock generator is unlocked (the frequency is correct but not the phase). When this happens, the BusEnable receiver is unable to discriminate anything other than long pulses of zeros or ones. Because the frequency of the clock generator is correct, it can count the length of these pulses with moderate accuracy.

**Table 5-13. Serial Mode Fields**

| SMode[1:0] | Description | Spec/Rsrv/ Undef |
|---|---|---|
| 00 | Increments Count00[3:0], clears Count11[7:0]. | Spec |
| 01 | – | Undef |
| 10 | – | Undef |
| 11 | Increments Count11[7:0], clears Count00[3:0] | Spec |

## 6. State Diagram

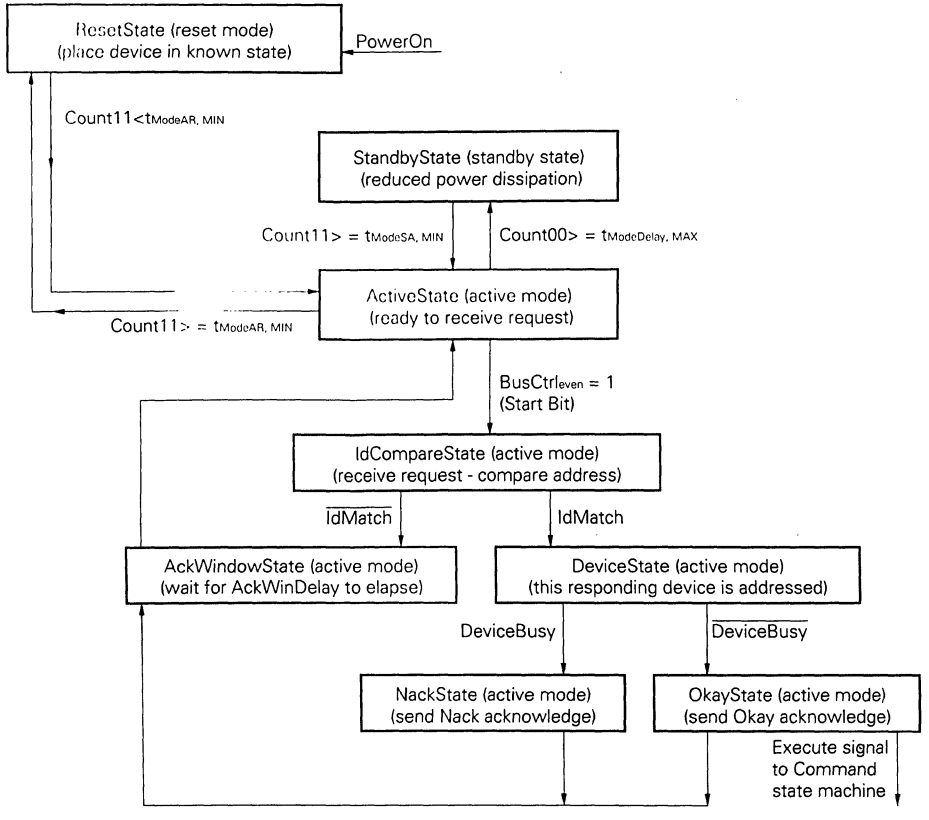
The following figure is a state diagram of the Frame state machine. The operating mode of the device depends upon which of the nine states it is in:

- reset mode - ResetState
- standby mode - StandbyState
- active mode - ActiveState, IdCompareState, DeviceState, OkayState, NackState, AckWindowState

This section will only discuss the first three states (ResetState, StandbyState, ActiveState). The remaining five states which are shown shaded in the state diagram (IdCompareState, DeviceState, OkayState, NackState, AckWindowState) will be dealt with in the **"Rambus DRAM user's manual (Reference Manual)"**.

The device will enter ResetState when power is initially applied (PowerOn). In ResetState, the device will be in the reset operating mode, in which all control registers assume a known state. If power has just been applied, the device will pass through ActiveState and settle in StandbyState, and remain there until serial mode packets are received from an initiating device.

**Figure 6-1. Frame State Machine - State Diagram**



ActiveState is the state in which all decisions are made to transition to the states for the other operating modes. From here, the device will also enter the transaction-framing states. Refer to **"Rambus DRAM user's manual (Reference Manual)"**.

After poweron, the device will re-enter ResetState when the value of the Count11[7:0] counter is greater than or equal to $t_{ModeAR,MIN}$. The device will leave ResetState when the value of the Count11[7:0] counter is less than $t_{ModeSA,MIN}$. This will happen when an SMode[1:0] field of 00 is received, causing the Count11[7:0] counter to clear.

The device will enter StandbyState when the value of the Count00[3:0] counter is greater than or equal to $t_{ModeDelay,MAX}$. The device will leave StandbyState when the value of the Count11[7:0] counter is greater than or equal to $t_{ModeSA,MIN}$.

**Caution The device will enter PowerDownState when the PD bit is set (after a delay of txxxx). But PD (MinInterval Register [3][2]) = 0 is necessary. Because, 18M RDRAM cannot accept Power Down request.**

## 6.1 Parameters for Operating Mode Transitions

The following table summarizes the parameter values associated with operating mode transitions of a responding device. A minimum and maximum value are given for the parameters to account for implementation differences. In all cases, the SMode[1:0] field of the consecutive serial mode packets must have the value 11 to cause an operating mode transition (with the exception of the $t_{ModeDelay,MAX}$ as mentioned in the previous section). Initiating devices must use values within the minimum and maximum SMode packet count requirements shown above to control operating mode transitions.

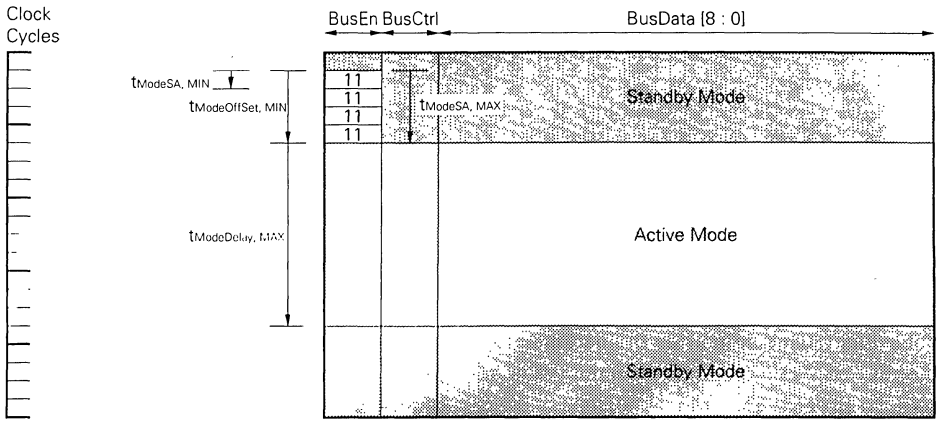
**Table 6-1. Responding Device Parameters for Operating Mode Transitions**

| Count Parameter Name | Minimum (clock cycles) | Maximum (clock cycles) | Description |
|---|---|---|---|
| $t_{ModeSA}$ | 1 | 4 | Number of SMode packets to cause a transition from Standby-Mode to ActiveMode |
| Rsrv | 5 | 9 | Reserved for future functionality |
| Undef | 10 | 15 | Undefined |
| Rsrv | 21 | 189 | Reserved for future functionality |
| Undef | 190 | 207 | Undefined |
| Rsrv | 225 | 253 | Reserved for future functionality |
| Undef | 254 | 271 | Undefined |
| $t_{ModeAR}$ | 272 | – | Number of SMode packets to cause a transition from Active-Mode to ResetMode |
| $t_{ModeOffSet}$ | 4 | – | Offset from beginning of SMode packet to request packet for standby to active transition |
| $t_{ModeDelay}$ | – | 10 | Delay from end of SMode packet to request packet for standby to active transition |

## 6.2 Standby Mode and Active Mode

The following figure shows the basic transitions between active and standby modes in response to serial mode packets

**Figure 6-2. Basic ActiveMode/StandbyMode Transitions**




This is a timing diagram, with time increasing in the downward direction. The time scale is in clock cycles, as shown on the left scale. The value of each of the eleven low-swing signal pins of the responding device is shown with the assumption that $t_{TR}$ is zero (the responding device is located at the master end of the Channel).

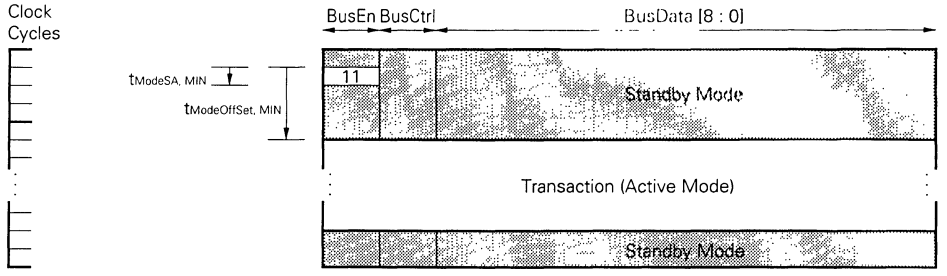
Serial mode packets with an SMode[1:0] field are shown as a box with a "11" label in the BusEn column. The BusEnable defaults to a logical zero value. The initiating device has transmitted $t_{ModeSA,MAX}$ serial mode packets with SMode[1:0] equal to 11 (this is the longest sequence permitted for invoking a standby to active transition). After the first $t_{ModeSA,MIN}$ serial mode packets, the device begins the transition to active mode. It reaches active mode after $t_{ModeOffSet, MIN}$ clock cycles after the start of the first serial mode packet. It remains there for $t_{ModeOffSet,MAX}$ clock cycles after the last serial mode packet.

The responding device is in active mode when it begins framing the request packet. A transaction may begin in any of the clock cycles with the light shading above (labeled "Active Mode").

If the serial mode packet(s) causing a standby to active mode transition are not followed by a transaction with $t_{ModeOffSet,MAX}$ clock cycles after the last serial mode packet, then the responding device will return to standby mode.

The next figure shows the case in which a transaction is started as early as possible after a serial mode packet which causes a standby to active mode transition.

**Figure 6-3. ActiveMode/StandbyMode Transition - Early Transaction**




A transaction is composed of packet types other than serial mode packets, and will be defined in the next chapter. These other packet types lie entirely inside the heavy black box in the above two figures. When a transaction has completed, the device returns to standby mode. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

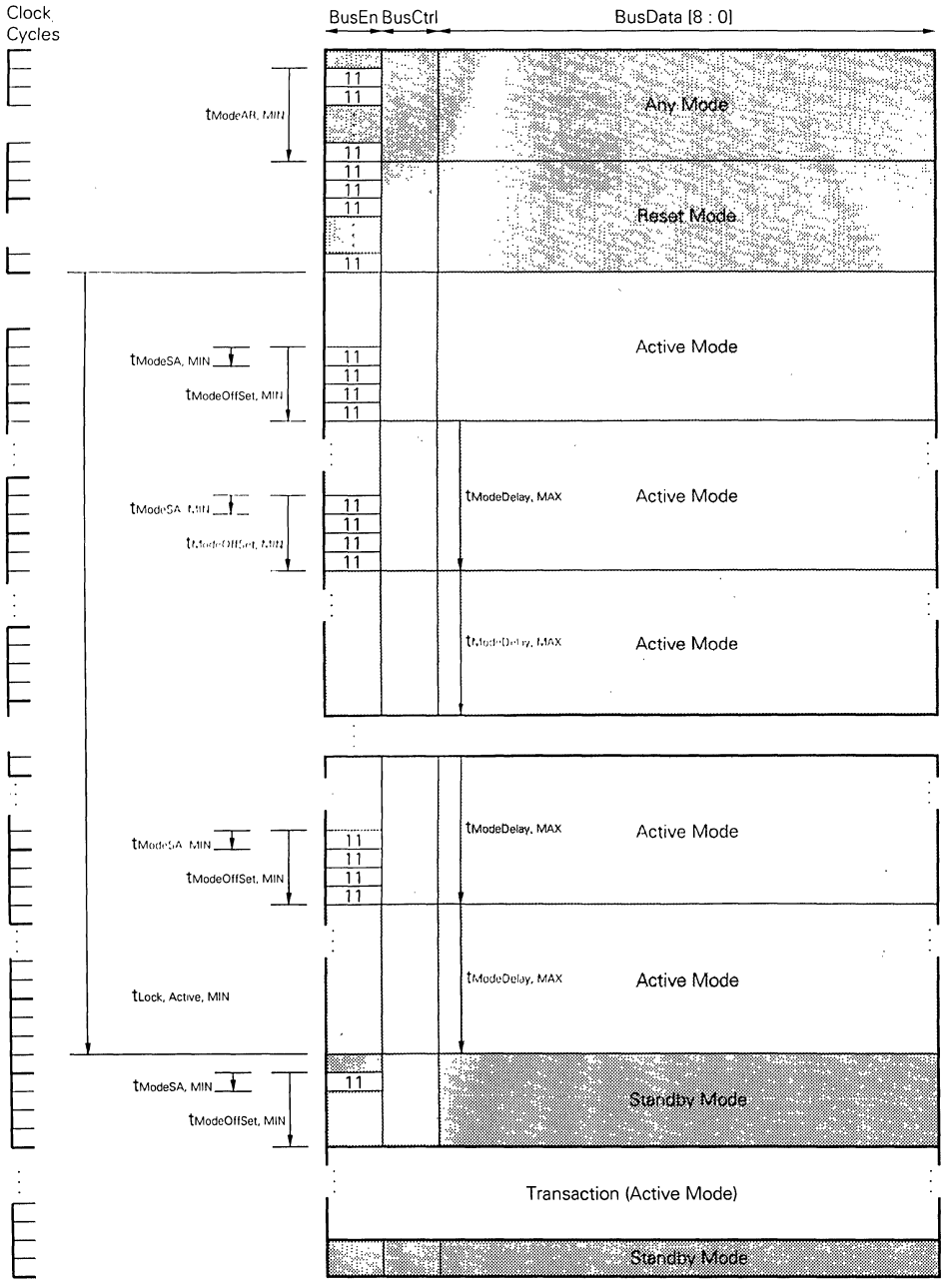
## 6.3 ResetMode

Reset mode is entered when a consecutive sequence of $t_{ModeAR,MIN}$ serial mode packets with a value of 11 are seen by a responding device (shown in the following figure). In reset mode, all devices enter a known state from which they may be Initialized. The device remains in reset mode for as long as serial mode packets with 11 value are received. When one or more serial mode packets with a value of 00 are seen, the responding device enters the active mode state.

Although devices enter the active mode state immediately, their clock circuitry requires a time $t_{Lock,MIN}$ to resynchronize. Initiating devices must wait this long after the transition out of reset mode before starting any transactions.

Note that in order to keep the devices in active mode during this synchronization process, it is necessary to provide a burst of serial mode packets every $t_{ModeDelay,MAX}$ clock cycles. This burst is shown as $t_{ModeSA,MAX}$ in length, but may, of course, be as short as $t_{ModeSA,MIN}$. If the device is not kept in active mode during synchronization, then the synchronization process requires $t_{Lock,MIN}$ clock cycles.

**Figure 6-4. ResetMode to ActiveMode Transition**



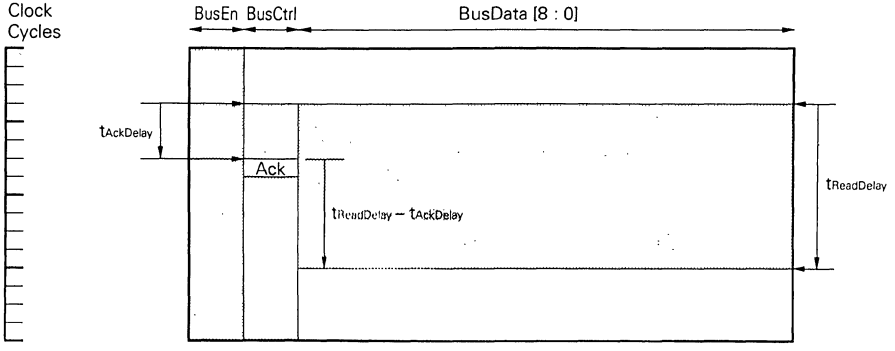
## 7. Transactions

### 7.1 Read Transactions

The following figure shows the basic form of a memory space or register space read transaction. There are request and acknowledge packets, with the same $t_{AckDelay}$ and $t_{AckWinDelay}$ timing constraints as already discussed ($t_{AckWinDelay}$ will not be shown explicitly on any further transaction diagrams in this document.

When the responding device transmits an Okay acknowledge packet to the initiating device, it will also transmit a data packet with read data. This packet is sent a time $t_{ReadDelay}$ after the end of the request packet. The $t_{ReadDelay}$ value is in $t_{CYCLE}$ units and is programmed into the ReadDelay field of the Delay register of each responding device. It is not required to be the same for all devices within a Rambus system, but the difference ($t_{ReadDelay}$ - $t_{AckDelay}$) is required to be the same. This allows initiating devices to use the acknowledge packet to determine when the read data packet begins. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 7-1. Read Transaction**



## 7.2 Write Transactions

The following figure shows the basic form of a memory space or register space write transaction. There are request and acknowledge packets, with the same $t_{AckDelay}$ and $t_{AckWinDelay}$ timing constraints as already discussed.

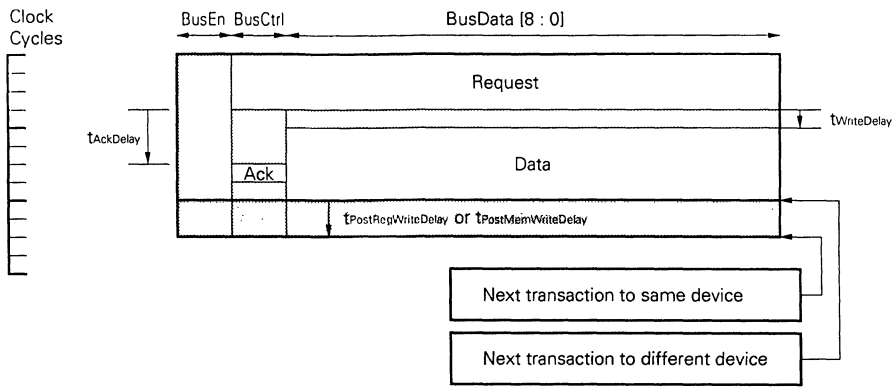
When the initiating device transmits a request packet to the responding devices, it will also transmit a data packet with write data. This packet is sent a time $t_{WriteDelay}$ cycles after the end of the request packet. The $t_{WriteDelay}$ is in $t_{CYCLE}$ units and is programmed into the WriteDelay field of the Delay register of each responding device. It is required to be the same for all devices within a Rambus system. A responding device will see the same $t_{WriteDelay}$ interval between the request and write data packets whether the device is on the Primary Channel or on a Secondary Channel.

If the responding device returns an Okay acknowledge packet, then the transaction is complete at the end of the acknowledge window or at the end of the write data packet, whichever is later. The next request packet can be transmitted in the following clock cycle except for the case in which a register or memory space write to a device is followed by any other transaction to that device. In that case, one of the following two intervals must be inserted between the two transactions, where the memory or register case depends upon the first transaction.

- $t_{PostRegWriteDelay}$ if the current transaction is a register space access
- $t_{PostMemWriteDelay}$ if the current transaction is a memory space access

If the responding device returns a Nack or Nonexistent acknowledge packet for a write command, then no write data packet is required by the responding device. The current transaction is complete at the end of the acknowledge window, or when the initiating device stops transmitting the write data packet, whichever is later. The next request packet can be transmitted in the following clock cycle. For the case of a Nack or Nonexistent, the initiating device must terminate the write data packet before another initiating device is given control of the Rambus Channel for a transaction. This is part of the arbitration mechanism used by the initiating devices. The arbitration mechanism is not specified in this document because it does not use the Rambus Channel. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 7-2. Write Transaction**



### 7.3 Read Transactions with Serial Address Packet

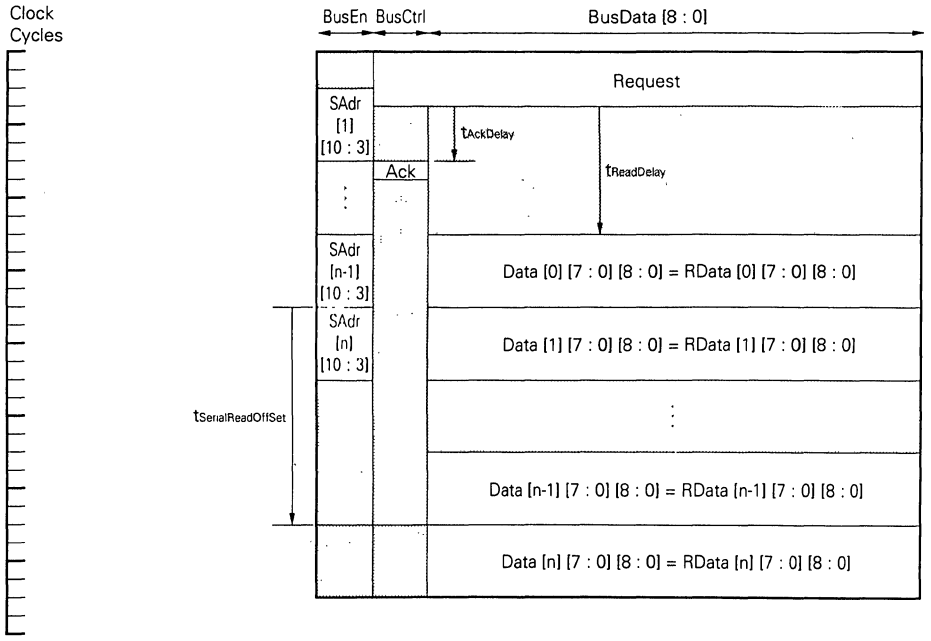
The following figure shows a memory space read transaction for a command which uses the serial address packet. For a transaction which moves (n+1) octbytes of read data, the serial address packet will be (4 × n) clock cycles in length (recall that the low-order address bits for the first octbyte of read data come from the request packet).

Each serial address subpacket (each SAdr[i][10:3] field) is transmitted by the initiating device a time $t_{SerialReadOffSet}$ clock cycles before the octbyte of read data to which it corresponds. This means that the serial address packet will move with the read data packet, with a constant offset.

- $t_{SerialReadOffSet}$ is the delay from the beginning of a serial address subpacket to the beginning of the read data subpacket (octbyte) with which it is associated.

The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 7-3. Read Transaction with Serial Address Packet**



### 7.4 Write Transactions with Serial Address Packet

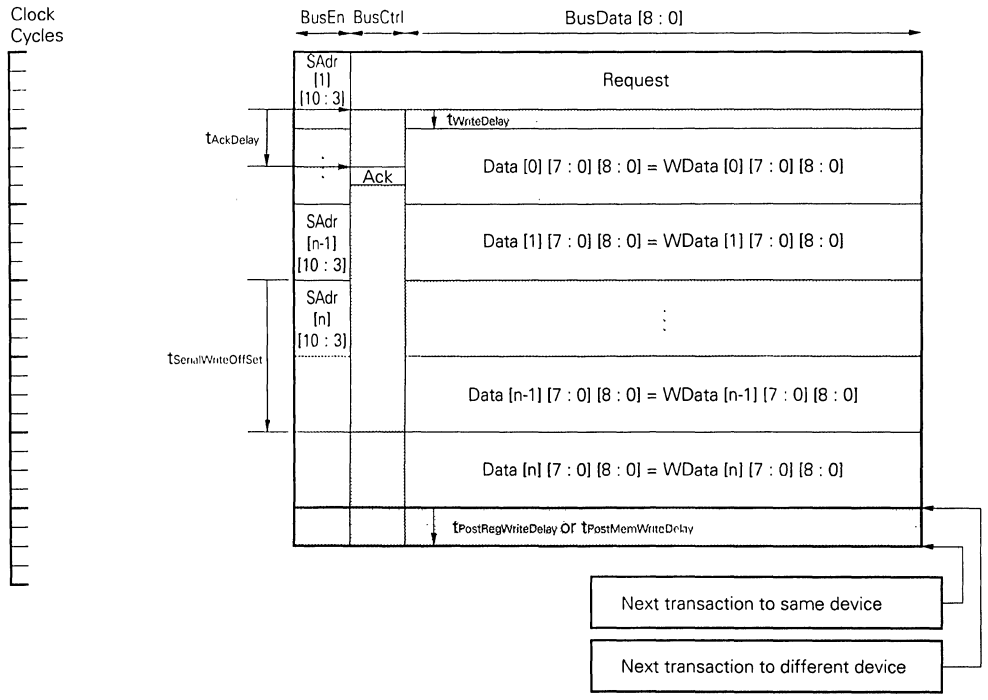
The following figure shows a memory space write transaction for a command which uses the serial address packet. For a transaction which moves (n+1) octbytes of write data, the serial address packet will be (4 × n) clock cycles in length (recall that the low-order address bits for the first octbyte of write data come from the request packet).

Each serial address subpacket (each SAdr[i][10:3] field) is transmitted by the initiating device a time $t_{SerialWriteOffSet}$ clock cycles before the octbyte of write data to which it corresponds. This means that the serial address packet will move with the write data packet, with a constant offset.

- $t_{SerialWriteOffSet}$ is the delay from the beginning of a serial address subpacket to the beginning of the write data subpacket (octbyte) with which it is associated.

Note that this offset interval is measured at the initiating device or the responding device; it will be the same at either point since the serial address packet and write data packet are moving in the same direction - from initiating device to responding device.

**Figure 7-4. Write Transaction with Serial Address Packet**



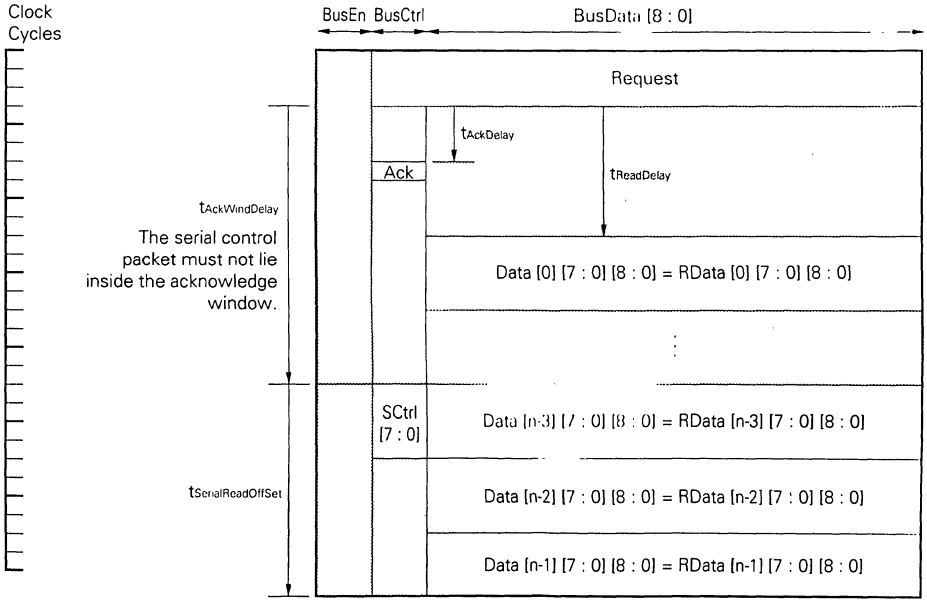
### 7.5 Read Transactions with Serial Control Packet

The following figure shows a memory space read transaction for a command which uses the serial control packet. This packet is used to terminate a transaction before the (CV+1) octbytes of read data have been transferred, where CV is the value of the Count[7:3] Field when interpreted as an unsigned, five bit integer. In the example shown, the read data is terminated after (n) octbytes have been transferred.

The serial control packet is transmitted by the initiating device a time $t_{SerialReadOffSet}$ clock cycles before the end of the last read data octbyte which is transmitted by the responding device.

The serial control packet is also constrained to lie entirely outside the $t_{AckWindDelay}$ interval, as shown in the figure, in order to avoid interference with the acknowledge packet which is being returned by the responding device. Violation of this constraint will produce undefined (Undef) results. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 7-5. Read Transaction with Serial Control Packet**



## 7.6 Write Transactions with Serial Control Packet

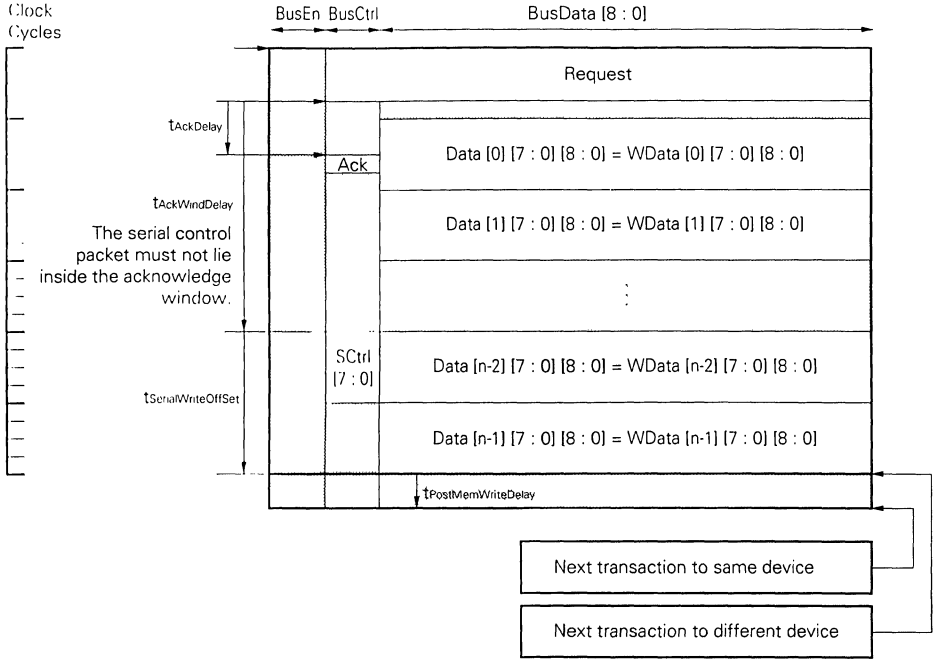
The following figure shows a memory space write transaction for a command which uses the serial control packet. This packet is used to terminate a transaction before the (CV+1) octbytes of write data have been transferred, where CV is the value of the Count[7:3] field when interpreted as an unsigned, five bit integer. In the example shown, the write data is terminated after (n) octbytes have been transferred.

The serial control packet is transmitted by the initiating device a time $t_{SerialWriteOffSet}$ clock cycles before the end of the last write data octbyte which is transmitted by the initiating device.

Note that this offset interval is measured at the initiating device or the responding device; it will be the same at either since the serial address packet and write data packet are moving in the same direction - from initiating device to responding device.

The serial control packet is also constrained to lie entirely outside the $t_{AckWinDelay}$ interval, as shown in the figure, in order to avoid interference with the acknowledge packet which is being returned by the responding device. Violation of this constraint will produce undefined (Undef) results.

**Figure 7-6. Write Transaction with Serial Control Packet**

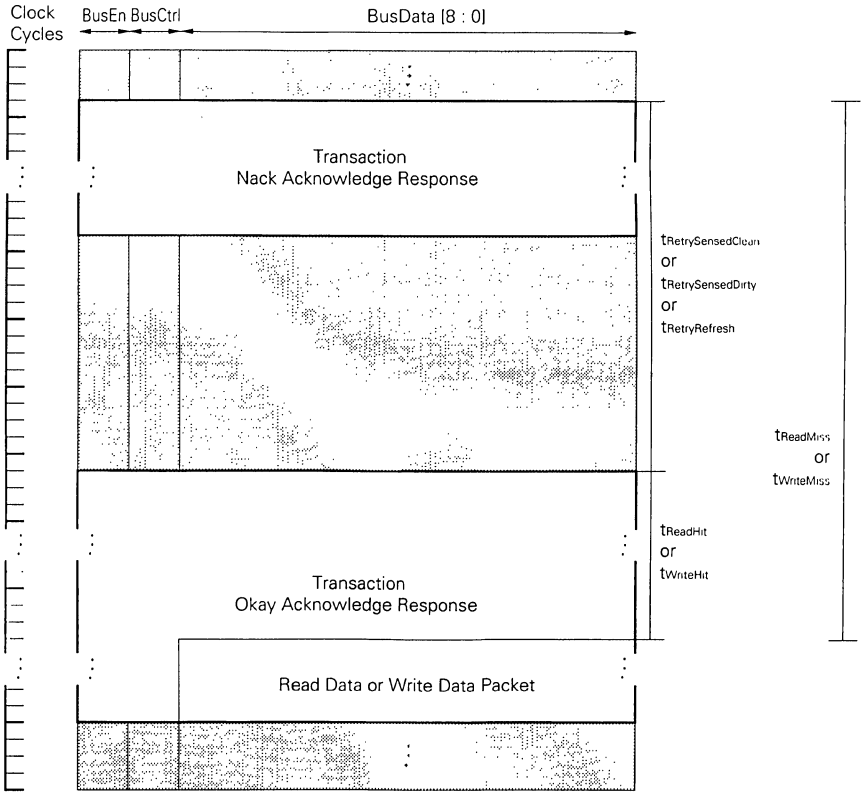


## 8. Nack Acknowledge Response

### 8.1 Retry and Miss Latency

If a responding device returns a Nack acknowledge packet, then no read or write data packet is transacted. The current transaction is complete at the end of the acknowledge window. It will be necessary to wait for an interval of time (called a tRETRY interval) before resubmitting the transaction. The following figure illustrates this case.

**Figure 8-1. Nack Acknowledge Response**




Once the tRETRY interval has elapsed, the transaction may be restarted by the initiating device, and the RDRAM will return an Okay acknowledge packet and the data packet will be transferred. An RDRAM will Nack any other transactions which are issued during the tRETRY interval.

Two miss latency parameters may be derived with the following equations:

$$t_{ReadMiss} = t_{RETRY} + t_{ReadHit} \quad \text{(Eq 8-1)}$$

$$t_{WriteMiss} = t_{RETRY} + t_{WriteHit} \quad \text{(Eq 8-2)}$$

where $t_{RETRY}$ = {$t_{RetrySensedClean}$, $t_{RetrySensedDirty}$, $t_{RetryRefresh}$}. The $t_{ReadMiss}$ and $t_{WriteMiss}$ parameters are the time from the beginning of the original (Nacked) request packet to the beginning of the data packet which is eventually transferred.

## 8.2 tRETRY Interval

### 8.2.1 Retry Due to RowMiss

If an initiating device requests a region of memory space in an RDRAM slave which is not currently held in the RowSenseAmpCache, the RDRAM will respond with a Nackacknowledge packet. The RDRAM will then begin a RowMiss operation to get the proper row into the RowSenseAmpCache. During the RowMiss, the RDRAM will Nack any request it is given. When the RowMiss is complete, the new row may be accessed.

Each bank has a Valid flag and a Dirty flag for its Row register. After reset, both are zero. After a RowMiss has caused a new row to be placed into the RowSenseAmpCache, the Row register contains its row address and the Valid flag is set to a one. If the RowSenseAmpCache contents are modified with a memory write transaction, the dirty flag will be set. These flags are not directly accessible to initiating devices.

A subsequent RowMiss will cause the old row to be written back to the bank (if it was dirty and an explicit restore was not forced with the Close bit in the request packet) and a new row to be placed into the RowSenseAmpCache. The time required for this is called the tRETRY time, and is added to the normal read and write hit latency times, as shown in the preceding figure. These times are given by the following equations. The component parameters are shown in a subsequent table. All of these tRETRY intervals correspond roughly to the cycle time parameter tRC of a conventional page mode DRAM. This is because RDRAMs use CAS-type accesses for all memory read and write transactions.

After a new row is sensed and placed into the RowSenseAmpCache, a final interval tRowImprestore is used to restore the row in core back to its original state. This is necessary because the DRAM sense operation is destructive. This interval is not in the critical timing path, and is performed in parallel with a subsequent data transfer. It can extend a subsequent retry operation.

There are two tRETRY equations for the 18M RDRAM:

$$t_{RetrySensedClean} = t_{RowOverHead} + t_{RowPrecharge} + t_{RowSense} \quad \text{(Eq 8-3)}$$

$$t_{RetrySensedDirty} = t_{RowOverHead} + t_{RowExprestore} + t_{RowPrecharge} + t_{RowSense} \quad \text{(Eq 8-4)}$$

The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

### 8.2.2 Retry Due to Pending Burst Refresh

In a 18M RDRAM, a refresh burst will also restore the currently accessed row if it is dirty. This requires a $t_{RowExprestore}$ interval. If the row is clean, this interval is not required. A burst of four rows are precharged/sensed/restored (using the $t_{RowImprestore}$ interval), and the current row is precharged/sensed so the RDRAM is left with its RowSenseAmpCache state unaltered (except the row's dirty flag will be cleared):

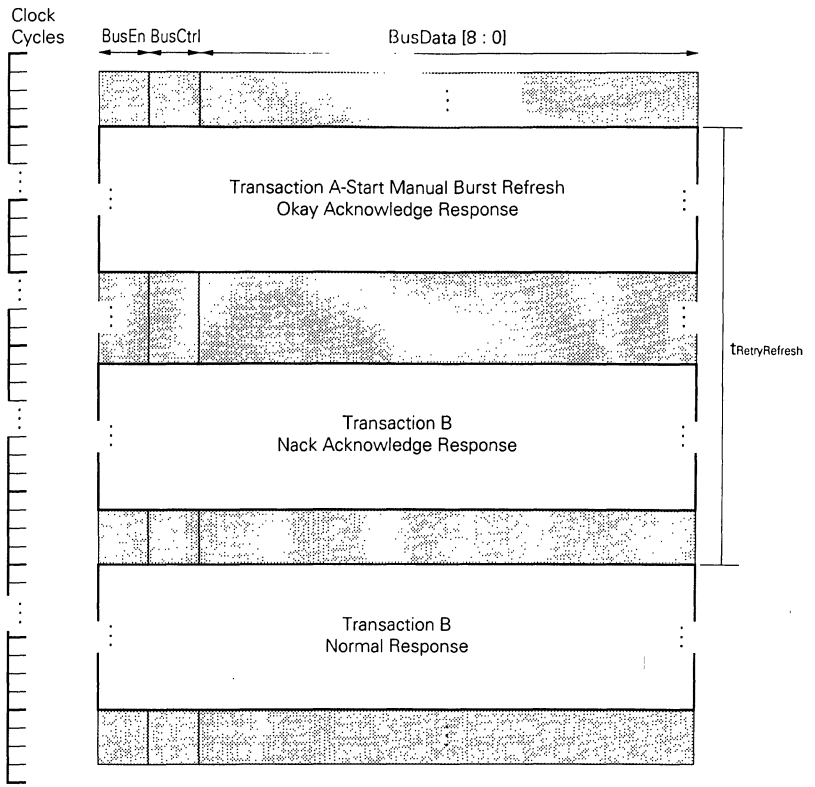
$$t_{RetryRefresh} = (t_{RowOverHead} + t_{RowExprestore}^{Note\ 1} + 2t_{RowPrecharge}^{Note\ 2} + t_{RowSense}) + 4\ (t_{RowOverHead} + t_{RowImprestore} + 2t_{RowPrecharge}^{Note\ 2} + t_{RowSense}) \quad \text{(Eq 8-5)}$$

When a transaction initiates a manual burst refresh in an RDRAM (transaction "A" in the figure below), the RDRAM will Nack all further transactions directed to in during the $t_{RetryRefresh}$ interval after. No information from these Nacked transactions will be retained after the $t_{RetryRefresh}$ interval. After the $t_{RetryRefresh}$ interval, transactions will be handled in a normal fashion. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Notes 1.** This term is not present if the current row is clean.

**2.** This term is present twice in each cycle because the $t_{RowPrecharge}$ interval is also used to ensure that a minimum delay between restore operations is met.

**Figure 8-2. Transaction Holdoff Due to Burst Refresh**



### 8.3 Retry Component Intervals

The $t_{RETRY}$ intervals are built from the $t_{RowOverHead}$, $t_{RowPrecharge}$, $t_{RowSense}$, $t_{RowImprestore}$, and $t_{RowExprestore}$ intervals. All five intervals are measured in $t_{CYCLE}$ units, and thus scale with the clock frequency.

The $t_{RowOverHead}$ interval consists of the RowMiss state machine overhead. The remaining four intervals represent the width of intervals used for timing core operations. These core operations have minimum times measured in nanosecond units (this is shown in the "core timing(ns)" columns in the table below). The four intervals are composed of a fixed part and a variable (programmable) part. If the clock frequency is reduced, the variable part may be reduced so the sum of the fixed and variable parts remain greater than or equal to the minimum core operation time (in nanoseconds).

**Table 8-1. Retry Components**

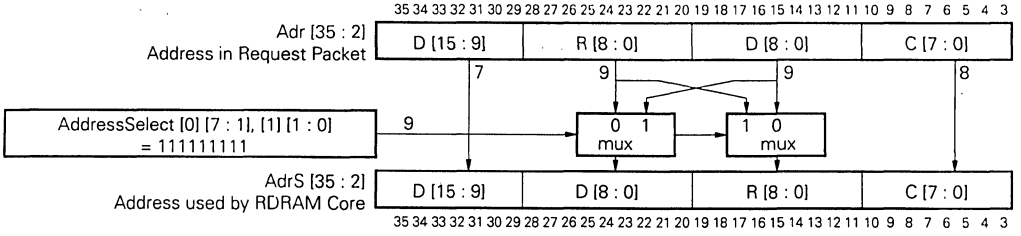
| Delay Parameter | Fixed Part (overhead) and Variable Part[Note] | 18M RDRAM | |
|---|---|---|---|
| | | $t_{CYCLE}$ Units (4 ns) | core timing (ns) with $t_{CYCLE}$ = 4ns |
| $t_{RowOverHead}$ | Row overhead | 6 | 24 |
| | – | n/a | |
| $t_{RowPrecharge}$ | RowPrecharge overhead | 4 | 20 |
| | RowPrecharge[4:0] | 1 | |
| $t_{RowSense}$ | RowSense overhead | 4 | 44 |
| | RowSense[4:0] | 7 | |
| $t_{RowImprestore}$ | RowImpRestore overhead | 4 | 56 |
| | RowImpRestore[4:0] | 10 | |
| $t_{RowExprestore}$ | RowExpRestore overhead | 4 | 32 |
| | RowExpRestore[4:0] | 4 | |

**Note** The variable part is programmed into the indicated field of the RasInterval register.

## 9. AddressMapping

The address space decoding logic contained in a 18M RDRAM is shown in the following figure. The initiating device places a 33 bit physical octbyte address Adr[35:3] on the Channel. This address is received by the RDRAM slave. The AddressSelect[1][1:0], [0][7:1] control register allows individual bits of the Adr[28:20] and Adr [19:11] fields to be swapped to produce the AdrS[28:20] and AdrS[19:11] fields. The Adr[35:29] and Adr[10:3] fields pass through unaltered to the AdrS[35:29] and AdrS[10:3] fields. The figure shows the case when AddressSelect[0][7:1],[1][1:0] = 111111111, and the two nine bit address fields are exchanged. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 9-1. AddressMapping Hardware**



## 10. Electrical Characteristics (Preliminary)

### Absolute Maximum Ratings

| Symbol | Parameter | MIN. | MAX. | Unit | Note |
|---|---|---|---|---|---|
| $V_{I,ABS}$ | Voltage applied to any RSL pin with respect to GND | −0.5 | $V_{DD}$+0.5 | V | |
| $V_{I,TTL,ABS}$ | Voltage applied to any TTL pin with respect to GND | −0.5 | $V_{DD}$+0.5 | V | |
| $V_{DD,ABS}$ | Voltage on $V_{DD}$ with respect to GND | −0.5 | $V_{DD,MAX}$+1.0 | V | |
| $T_{OPT}$ | Operation temperature | 0 | +70 | °C | 1 |
| $T_{STORE}$ | Storage temperature | −55 | +125 | °C | |

**Caution The following table represents stress ratings only, and functional operation at the maximums is not guaranteed. Extended exposure to the maximum ratings may affect device reliability. Furthermore, although devices contain protective circuitry to resist damage from static electric discharge, always take precautions to avoid high static voltages or electric fields.**

**Note 1** This parameter apply at the status of using 50% Rambus channel by Read or Write and a transverse air flow greater than 1.5m/s maintained.

### Thermal Parameters

| Symbol | Parameter | MIN. | MAX. | Unit |
|---|---|---|---|---|
| $T_J$ | Junction operating temperature | | 100 | °C |
| $\Theta_{JC}$ | Junction-to-Case thermal resistance | | 5 | °C/W |

### Capacitance

| Symbol | Parameter | MIN. | MAX. | Unit |
|---|---|---|---|---|
| $C_{I/O}$ | Low-swing input/output parasitic capacitance | | 2 | pF |
| $C_{I,TTL}$ | TTL input parasitic capacitance | | 8 | pF |

### Power Consumption

| Mode | Parameter | | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| $I_{CC1}$ | Active Current | -A45 | | 110 | mA |
| | | -A50 | | 125 | |
| $I_{CC2}$ | Standby Current | -A45 | | 330 | mA |
| | | -A50 | | 350 | |
| $I_{CC3}$ | Read Operation Current | -A45 | | 440 | mA |
| | (Burst Length = 256) | -A50 | | 480 | |
| $I_{CC4}$ | Write Operation Current | -A45 | | 435 | mA |
| | | -A50 | | 460 | |

**Caution These do not include the $I_{OL}$ current passing through the low-swing pins to ground.**

### Recommended Operating Conditions

| Symbol | Parameter | MIN. | MAX. | Unit |
|---|---|---|---|---|
| $V_{DD}$, $V_{DDA}$ | Supply voltage | 3.15 | 3.45 | V |
| $V_{REF}$ | Reference voltage | 1.95 | 2.15 | V |
| $V_{swing}$ | Input voltage range | 1.0 | 1.4 | V |
| $V_{IL}$ | Input low voltage | $V_{REF}$–0.7 | $V_{REF}$–0.5 | V |
| $V_{IH}$ | Input high voltage | $V_{REF}$+0.5 | $V_{REF}$+0.7 | V |
| $V_{IL, TTL}$ | TTL input low voltage | –0.5 | +0.8 | V |
| $V_{IH, TTL}$ | TTL input high voltage | 2.0 | $V_{DD}$+0.5 | V |

### DC Characteristics (Recommended operating conditions unless otherwise noted)

| Symbol | Parameter | Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| $I_{REF}$ | $V_{REF}$ current | $V_{REF}$=Maximum | –10 | +10 | μA |
| $I_{OH}$ | High level output current | $0 \le V_{OUT} \le V_{DD}$ | –10 | +10 | μA |
| $I_{OL}$ | Low level output current | $V_{OUT}$=1.6 V | | 25 | mA |
| $I_{I, TTL}$ | TTL input leakage current | $0 \le V_{I, TTL} \le V_{DD}$ | –10 | +10 | μA |
| $V_{OH, TTL}$ | High level TTL output voltage | $I_{OH, TTL}$=–0.25 mA | 2.4 | $V_{DD}$ | V |
| $V_{OL, TTL}$ | Low level TTL output voltage | $I_{OL, TTL}$=1.0 mA | 0 | 0.4 | V |

### Recommended Timing Conditions

| Symbol | Parameter | | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| $t_{PAUSE}$ | Pause time after Power On | | | 200 | μs |
| $t_{CR}$, $t_{CF}$ | TxClk and RxClk input rise and fall times | | 0.3 | 0.7 | ns |
| $t_{CYCLE}$ | TxClk and RxClk cycle times | -A45 | 4.45 | 6 | ns |
| | | -A50 | 4 | 6 | ns |
| $t_{TICK}$ | Transport time per bit per pin (this timing interval is synthesized by the RDRAM's internal clock generator) | | $t_{CYCLE}/2$ | $t_{CYCLE}/2$ | ns |
| $t_{CH}$, $t_{CL}$ | TxClk and RxClk high and low times | | 47% | 53% | $t_{CYCLE}$ |
| $t_{TR}$ | TxClk-RxClk differential | | 0.25 | 0.7 | ns |
| $t_{SD}$ | SIn-to-SOut propagation delay | | | 50 | ns |
| $t_Q$ | TxClk-to-Data/Control output time | | $t_{CYCLE}/8+0.05$ | $t_{CYCLE}3/8-0.05$ | ns |
| $t_S$ | Data/Control-to-RxClk setup time | | $t_{CYCLE}/4-0.05$ | | ns |
| $t_H$ | RxClk-to-Data/Control hold time | | $t_{CYCLE}/4-0.05$ | | ns |
| $t_{REF}$ | Refresh interval | | | 32 | ms |
| $t_{LOCK}$ | RDRAM internal clock generator lock time | | 500 | | $t_{CYCLE}$ |

### Transaction Timing Characteristics

| Symbol | Parameter | MIN. | Unit |
|---|---|---|---|
| $t_{PostRegWriteDelay}$ | Delay from the end of the current transaction to the beginning of the next transaction if the current transaction is a write to register space and the next transaction is made to the same device.<br>Use zero delay if the next transaction is to a different device. | 6 | $t_{CYCLE}$ |
| $t_{PostMemWriteDelay}$ | Delay from the end of the current transaction to the beginning of the next transaction if the current transaction is a write to memory space and the next transaction is made to the same device.<br>Use zero delay if the next transaction is to a different device. | 4 | $t_{CYCLE}$ |
| $t_{PostMemReadDelay}$ | Delay from the end of the current memory read transaction to the beginning of the next transaction. | 2 | $t_{CYCLE}$ |
| $t_{SerialReadOffSet}$ | Delay from the beginning of a serial address subpacket or serial control packet to the beginning of the read data subpacket (octbyte) with which it is associated. | 12 | $t_{CYCLE}$ |
| $t_{SerialWriteOffSet}$ | Delay from the beginning of a serial address subpacket or serial control packet to the beginning of the write data subpacket (octbyte) with which it is associated. | 8 | $t_{CYCLE}$ |

### Data and Transaction Latency Characteristics

| Symbol | Parameter | MIN. | Unit | Notes |
|---|---|---|---|---|
| $t_{ReadDelay}$ | Delay from the end of a read request packet to the beginning of the read data packet. | 7 | $t_{CYCLE}$ | 1 |
| $t_{WriteDelay}$ | Delay from the end of a write request packet to the beginning of the write data packet. | 1 | $t_{CYCLE}$ | 2 |

**Notes 1.** $t_{ReadDelay}$ is programmed to its minimum value.
**2.** $t_{WriteDelay}$ is programmed to its minimum value.

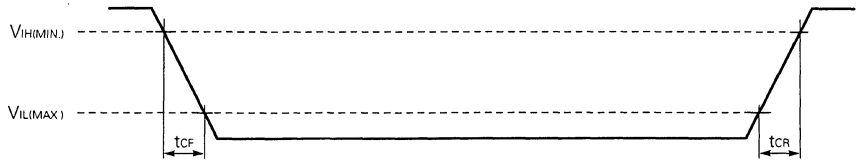
### Hit, Retry and Miss Delay Characteristics

| Symbol | Parameter | | MIN. | Unit | Notes |
|---|---|---|---|---|---|
| $t_{ReadHit}$ | Start of request packet to start of read data packet for row hit (Okay). | | 10 | $t_{CYCLE}$ | 1 |
| $t_{WriteHit}$ | Start of request packet to start of write data packet for row hit (Okay). | | 4 | $t_{CYCLE}$ | 1 |
| $t_{RetrySensedClean}$ | Start of request packet for row miss (Nack) to start of request packet for row hit (Okay). The previous row is unmodified. | | 22 | $t_{CYCLE}$ | 2 |
| $t_{RetrySensedDirty}$ | Start of request packet for row miss (Nack) to start of request packet for row hit (Okay). The previous row is modified. | | 30 | $t_{CYCLE}$ | 2 |
| $t_{RetryRefresh}$ | Start of request packet for row miss (Nack) to start of request packet for row hit (Okay). | Clean | 191 | $t_{CYCLE}$ | 2 |
| | | Dirty | 199 | | |
| $t_{ReadMiss}$ | Start of request packet for row miss (Nack) to start of Read Data packet for row hit (Okay). | | 32 | $t_{CYCLE}$ | 3 |
| $t_{WriteMiss}$ | Start of request packet for row miss (Nack) to start of Write Data packet for row hit (Okay). | | 26 | $t_{CYCLE}$ | 3 |

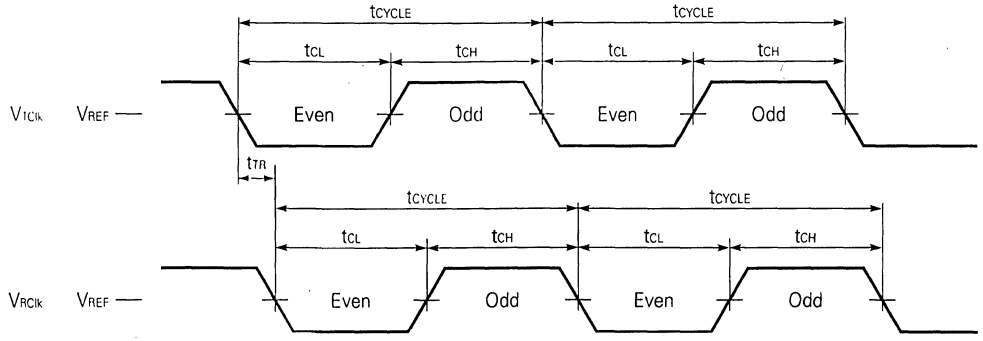
**Notes 1.** Programmable
**2.** $t_{RowExprestore}$, $t_{Precharge}$, and $t_{Sense}$ are programmed to their minimum value.
**3.** Calculated with $t_{RetrySensedClean(MIN)}$.

## Rise/Fall Timing Chart

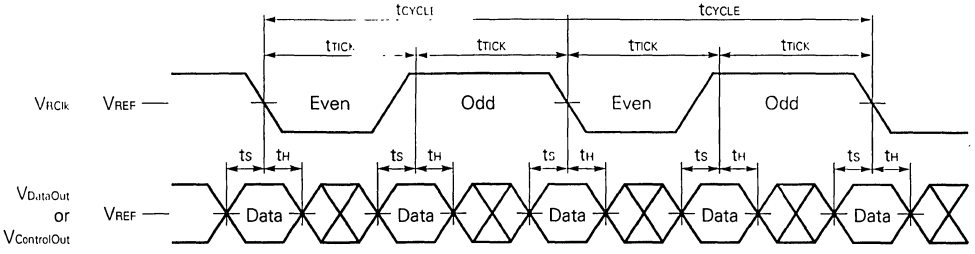
$V_{RxClk}$, $V_{TxClk}$



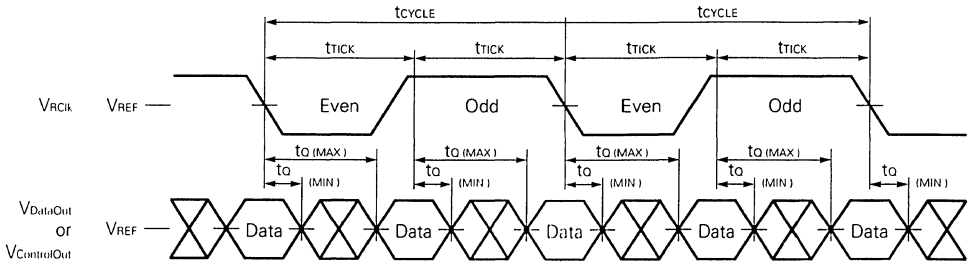

## Clock Timing Chart



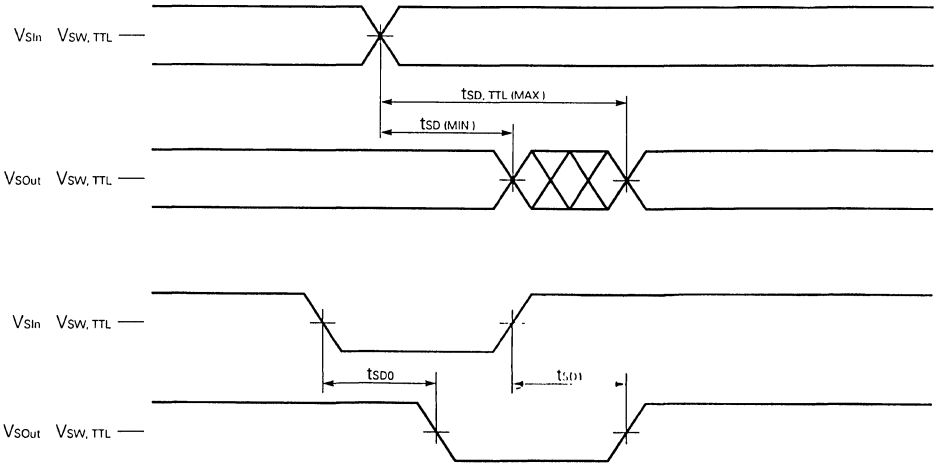

## Receive Data Timing Chart



### Transmit Data Timing Chart



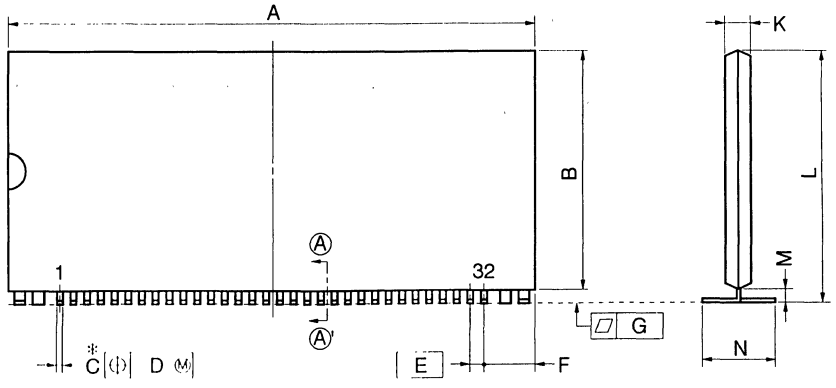

### Serial Configuration Pin Timing Chart

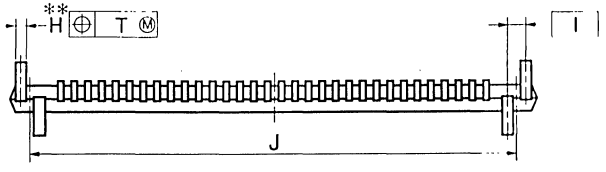



**Remark** $V_{SW,\,TTL}$ = 1.5 V

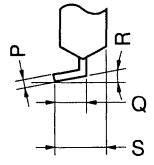
## 11. Package Drawings

## 32 PIN PLASTIC SVP (11×25)




Detail of Ⓐ - Ⓐ' part




NOTE

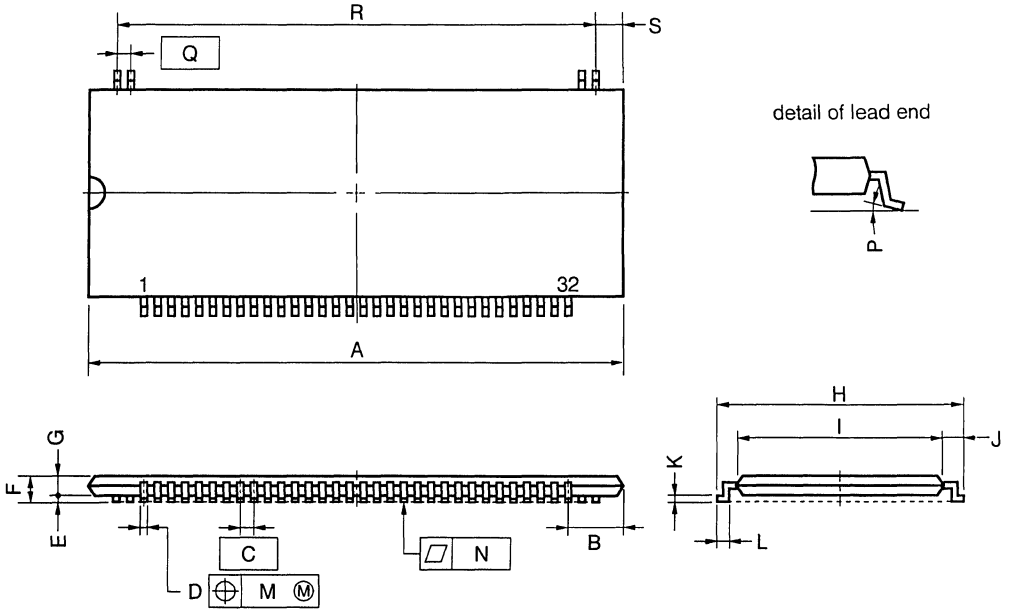
* Each I/O lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

** Each support lead centerline is located within 0.18 mm (0.007 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 25.30 MAX. | 0.996 MAX. |
| B | 11.0±0.1 | 0.433±0.004 |
| C | 0.24±0.06 | $0.009^{+0.003}_{-0.002}$ |
| D | 0.13 | 0.005 |
| E | 0.65 (T.P.) | 0.026 (T.P.) |
| F | 2.575 MAX. | 0.102 MAX. |
| G | 0.10 | 0.004 |
| H | 0.52±0.06 | 0.020±0.002 |
| I | 0.9 (T.P.) | 0.035 (T.P.) |
| J | 23.20 | 0.913 |
| K | 1.25 | 0.049 |
| L | 11.80 MAX. | 0.465 MAX. |
| M | 0.5±0.1 | $0.020^{+0.004}_{-0.005}$ |
| N | 3.70 MAX. | 0.146 MAX. |
| P | $0.17^{+0.025}_{-0.015}$ | 0.007±0.001 |
| Q | 0.9±0.25 | $0.035^{+0.011}_{-0.010}$ |
| R | $3°^{+7°}_{-3°}$ | $3°^{+7°}_{-3°}$ |
| S | 1.90 MAX. | 0.075 MAX. |
| T | 0.18 | 0.007 |

S32VN-65-9

## 72/36 PIN PLASTIC SSOP TYPE




**NOTE**

Each lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 25.30 MAX. | 0.996 MAX. |
| B | 2.575 MAX. | 0.102 MAX. |
| C | 0.65 (T.P.) | 0.026 (T.P.) |
| D | $0.24\pm0.06$ | $0.009^{+0.003}_{-0.002}$ |
| E | $0.25\pm0.05$ | $0.010^{+0.002}_{-0.003}$ |
| F | 1.6 MAX. | 0.063 MAX. |
| G | 1.25 | 0.049 |
| H | $13.0\pm0.2$ | $0.512\pm0.008$ |
| I | $11.0\pm0.1$ | $0.433\pm0.004$ |
| J | $1.0\pm0.2$ | $0.039^{+0.009}_{-0.008}$ |
| K | $0.17^{+0.025}_{-0.015}$ | $0.007\pm0.001$ |
| L | $0.5\pm0.1$ | $0.020^{+0.004}_{-0.005}$ |
| M | 0.13 | 0.005 |
| N | 0.10 | 0.004 |
| P | $3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$ | $3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$ |
| Q | 0.65 (T.P.) | 0.026 (T.P.) |
| R | 22.75 | 0.896 |
| S | 1.275 MAX. | 0.051 MAX. |

P32G6-65A

[MEMO]

NEC

MOS INTEGRATED CIRCUIT

# μPD488130L

## 16M-BIT Rambus DRAM
## 1M-WORD X 8-BIT X 2-BANK

### Description

The 16-Megabit Rambus™ DRAM (RDRAM™) is an extremely-high-speed CMOS DRAM organized as 2M words by 8 bits and capable of bursting up to 256 bytes of data at 2 ns per byte. The use of Rambus Signaling Logic (RSL) technology makes this 500 MHz transfer rate achievable while using conventional system and board design methodologies. Low latency is attained by using the RDRAM's large internal sense amplifier arrays as high speed caches.

RDRAMs are general purpose high-performance memory devices suitable for use in a broad range of applications including main memory, graphics, video, and any other application where high-performance and low cost are required.

Detailed information about product features and specifications can be found in the following document. Please make sure to read this document before starting design.

**Rambus DRAM user's manual (Reference Manual)** : IEU-1401

Rambus and RDRAM are trademarks of Rambus Inc.

### Features

- Rambus Interface
- 500 MB/sec peak transfer rate per RDRAM
- RSL interface
- Synchronous protocol for fast block-oriented transfers
- Direct connection to Rambus ASICs, MPUs, and Peripherals
- 40 ns from start of read request to first byte; 2 ns per byte thereafter
- Features for graphics include random-access mode, write-per-bit and mask-per-bit operations
- Dual 2K-Byte sense amplifiers act as caches for low latency accesses
- Multiple power-saving modes
- On-chip registers for flexible addressing and timing
- Low pincount-only 15 active signals
- Standardized pinout across multiple generations of RDRAMs
- 3.3 volt operation

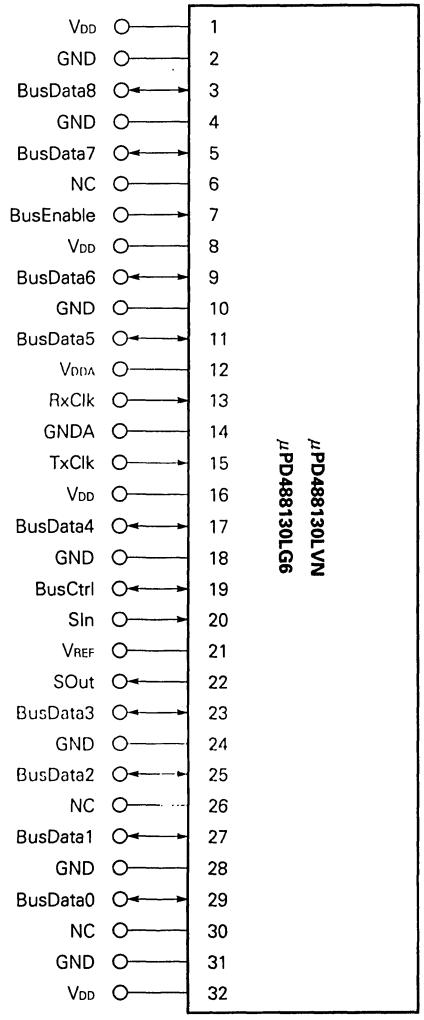
### Ordering Information

| Part Number | Clock Frequency | Operation Voltage | Package |
|---|---|---|---|
| μPD488130LVN-A50 | 250MHz | 3.3±0.15 V | 32-pin plastic SVP (11 × 25) |
| μPD488130LVN-A45 | 225MHz | 3.3±0.15 V | 32-pin plastic SVP (11 × 25) |
| μPD488130LG6-A50 | 250MHz | 3.3±0.15 V | 72/36-pin plastic SSOP type |
| μPD488130LG6-A45 | 225MHz | 3.3±0.15 V | 72/36-pin plastic SSOP type |

**The information in this document is subject to change without notice.**

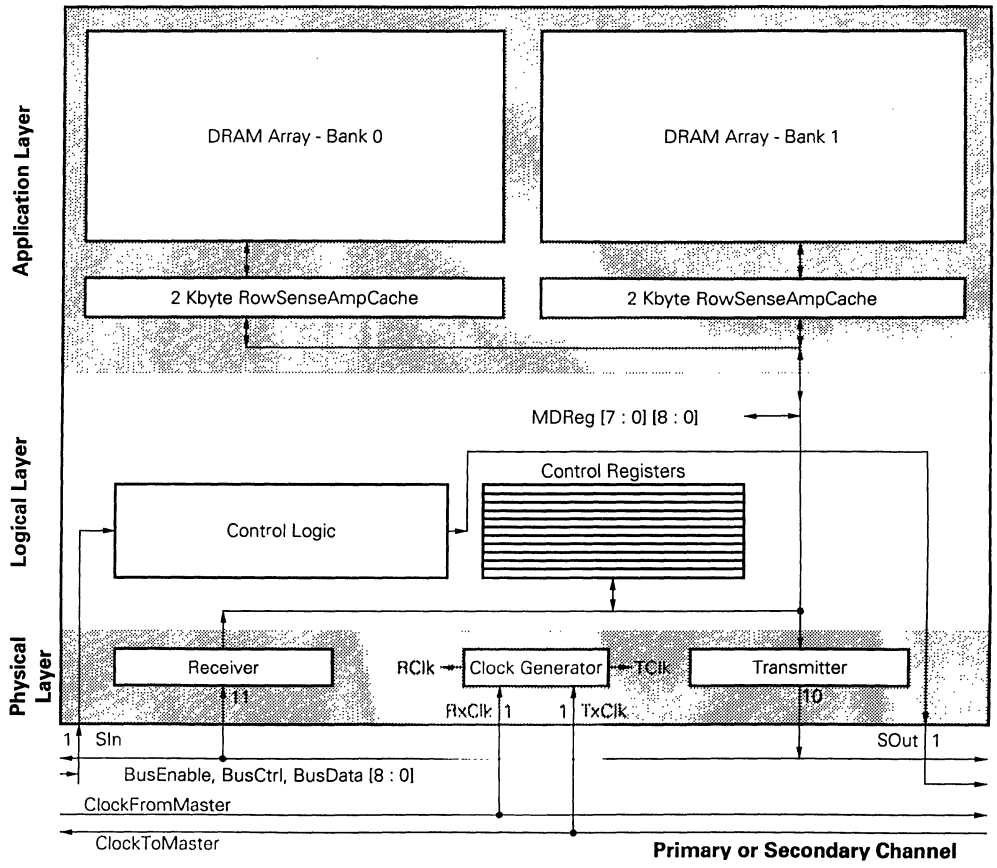
## Pin Configuration (Marking Side)

**32-pin plastic SVP (11 × 25)**
**72/36-pin plastic SSOP type**




| | |
|---|---|
| BusData 0 - BusData 8 | : Bus Data (Input/Output) |
| RxClk | : Receive Clock (Input) |
| TxClk | : Transmit Clock (Input) |
| $V_{REF}$ | : Logic Threshold Voltage (Input) |
| BusCtrl | : BusCtrl (Input/Output) |
| BusEnable | : BusEnable (Input) |
| $V_{DD}$, $V_{DDA}$ | : Power Supply |
| GND, GNDA | : Ground |
| SIn | : Serial Input (Input) |
| SOut | : Serial Output (Output) |
| NC | : No Connection |

## Block Diagram



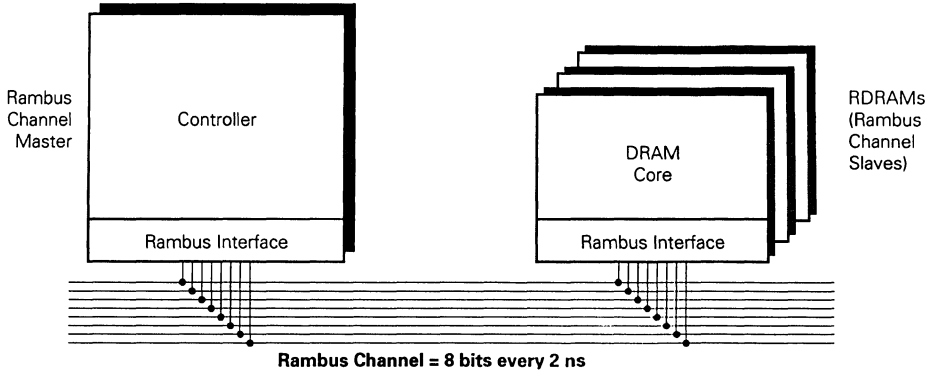
## 1. Pin Function

| Signal | I/O | Description |
|---|---|---|
| BusData [8:0] | I/O | Signal lines for request, write data, and read data packets. The request packet contains the address, operation codes, and the count of the bytes to be transferred. This is a low-swing, active-low signal referenced to $V_{REF}$.<br>BusData [8] is "Don't Care" in data packet of the accessing memory space. |
| RxClk | I | Receive clock. Incoming request and write data packets are aligned to this clock. This is a low-swing, active-low signal referenced to $V_{REF}$. |
| TxClk | I | Transmit clock. Outgoing acknowledge and read packets are aligned with this clock. This is a low-swing, active-low signal referenced to $V_{REF}$. |
| $V_{REF}$ | I | Logic threshold voltage for low swing signals. |
| BusCtrl | I/O | Control signal to frame packets, to transmit part of the operation code, and to acknowledge requests. Low-swing, active-low signal referenced to $V_{REF}$. |
| BusEnable | I | Control signal to enable the bus. Long assertions of this signal will reset all devices on the Channel. This is a low-swing, active-low signal referenced to $V_{REF}$. |
| $V_{DD}$, $V_{DDA}$ | | +3.3 V power supply. $V_{DDA}$ is a separate analog supply. |
| GND, GNDA | | Circuit ground. GNDA is a separate analog ground. |
| SIn | I | Initialization daisy chain input. TTL levels. Active high. |
| SOut | O | Initialization daisy chain output. TTL levels. Active high. |

## 2. Rambus System Overview

A typical Rambus memory system has three main elements: the Rambus Channel, the RDRAMs, and a Rambus Interface on a controller. The logical representation of this is shown in the following figure.

**Figure 2-1. Logical Representation**

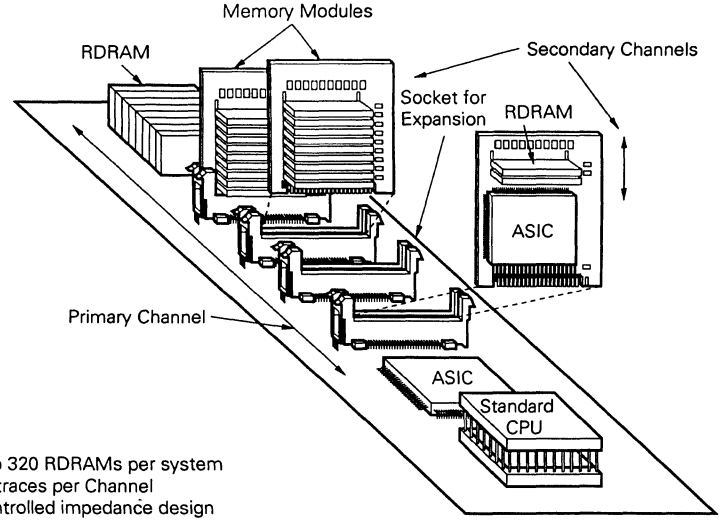



The Rambus Channel is a synchronous, high-speed, byte-wide bus that is used to directly connect Rambus devices together. Using only 13 high-speed signals, the Channel carries all address, data, and control information to and from devices through the use of a high level block-oriented protocol.

The Rambus Interface is implemented on both master and slave devices. Rambus masters are the only devices capable of generating transaction requests and can be ASIC devices, memory controllers, graphics engines, peripheral chips, or microprocessors. RDRAMs are slave devices and only respond to requests from master devices.

The following figure shows a typical physical implementation of a Rambus system. It includes a controller ASIC that acts as the Channel master and a base set of RDRAMs soldered directly to the board. An RSocket™ is included on the Channel for memory upgrade using RModule™ expansion cards.

**Figure 2-2. A Rambus System Example**

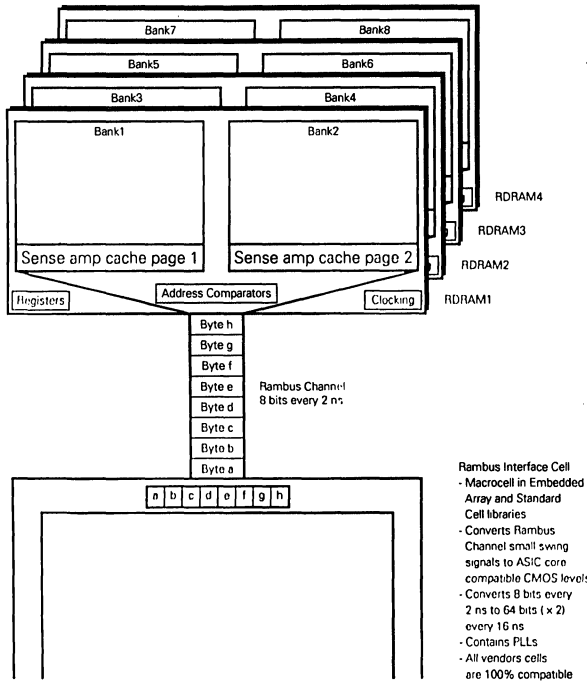


## 3. Rambus Signaling Logic

RSL technology is the key to attaining the high data rates available in Rambus systems. By employing high quality transmission lines, current-mode drivers, low capacitive loading, low-voltage signaling, and precise clocking, systems reliably transfer data at 2 nanosecond intervals on a Rambus Channel with signal quality that is superior to TTL or GTL-based interfaces.

All Rambus Interfaces incorporate special logic to convert signals from RSL to CMOS levels for internal use. In addition, these interfaces convert the Channel data rate of one byte every 2 nanoseconds to an internal data rate of 8 bytes every 16 nanoseconds as shown in the following figure. Although the bandwidth remains the same, the use of a wide internal bus eases internal timing requirements for chip designers.

**Figure 3-1. Converting the Channel Data Rate**



## 4. Register Space Map

The following table summarizes the registers included in all 16M RDRAMs.

**Table 4-1. Registers Space Map**

| Register Name | Adr[20:10] | Adr[9:2] | Register Number |
|---|---|---|---|
| Device Type[3:0][8:0] | xx...xx | 00000000 | 0 |
| DeviceId[3:0][8:0] | xx...xx | 00000001 | 1 |
| Delay[3:0][8:0] | xx...xx | 00000010 | 2 |
| Mode[3:0][8:0] | xx...xx | 00000011 | 3 |
| RefInterval[3:0][8:0] | xx...xx | 00000100 | 4 |
| RefRow[3:0][8:0] | xx...xx | 00000101 | 5 |
| RasInterval[3:0][8:0] | xx...xx | 00000110 | 6 |
| MinInterval[3:0][8:0] | xx...xx | 00000111 | 7 |
| AddressSelect[3:0][8:0] | xx...xx | 00001000 | 8 |
| DeviceManufacturer[3:0][8:0] | xx...xx | 00001001 | 9 |
| Undefined | xx...xx | 0000101x | 10-11 |
| Undefined | xx...xx | 000011xx | 12-15 |
| Undefined | xx...xx | 0001xxxx | 16-31 |
| Undefined | xx...xx | 001xxxxx | 32-63 |
| Undefined | xx...xx | 01xxxxxx | 64-127 |
| Row[3:0][8:0] | xx...xx | 10000000 | 128 |
| Undefined | xx...xx | 10000001 | 129 |
| Undefined | xx...xx | 1000001x | 130-131 |
| Undefined | xx...xx | 100001xx | 132-135 |
| Undefined | xx...xx | 10001xxx | 136-143 |
| Undefined | xx...xx | 1001xxxx | 144-159 |
| Undefined | xx...xx | 1010xxxx | 160-175 |
| Undefined | xx...xx | 1011xxxx | 176-191 |
| Undefined | xx...xx | 1100xxxx | 192-207 |
| Undefined | xx...xx | 1101xxxx | 208-223 |
| Undefined | xx...xx | 1110xxxx | 224-239 |
| Undefined | xx...xx | 1111xxxx | 240-255 |

**(1) Device Type Register**

This register specifies RDRAM configuration and size.

Device Type [0] [2]=0: This means that the RDRAM is 8-bit wide.

**(2) DeviceId Register**

This register specifies RDRAM base address.

**(3) Delay Register**

This register specifies RDRAM programmable CAS delay values.

**(4) Mode Register**

This register specifies RDRAM programmable output drive current.

**(5) RefInterval Register**

This register specifies RDRAM programmable refresh interval.

RefInterval Register is used to time the refresh interval for devices which require refresh.

**(6) RefRow Register**

This register specifies RDRAM refresh row and bank address.

The RefRow register contains read-write fields. It is used to keep track of the bank and row being refreshed. Normally this register is only read or written for testing purposes. The fields are aliased in the following way:

RowField[7:1] equals RefRow[0][7:1]
RowField[9:8] equals RefRow[2][1:0]
BankField[3] equals RefRow[1][3]

**(7) RasInterval Register**

This register specifies RDRAM programmable RAS delay values. The RasInterval Register contains four write-only fields. When a rowmiss occurs, or when a row is being refreshed during a burst refresh operation, it is necessary for the control logic of an RDRAM to count the appropriate number of clock cycles ($t_{CYCLE}$) for four intervals. This is done with a counter which is loaded successively with three values from the RasInterval Register. This counter is not available for read access and must be tested indirectly.

**(8) MinInterval Register**

This register specifies RDRAM refresh and powerdown control.

This register provides the minimum values for three time intervals for framing packets.

The time intervals are specified in clock cycle ($t_{CYCLE}$) units.

**(9) AddressSelect Register**

This register specifies RDRAM address mapping.

**(10) DeviceManufacturer Register**

This register specifies RDRAM manufacturer information.

This register specifies the manufacturer of the device. Additional bits are available for manufacturer specific information, e.g. stepping or revision numbers.

**(11) Row Register**

This register specifies RDRAM current sensed row in each bank.

The detailed functional description is provided in RDRAM Reference Manual.

## 5. Packet Formation

### 5.1 Packet Summary

The following table summarizes the transmit/receive functionality for the different packet classes.

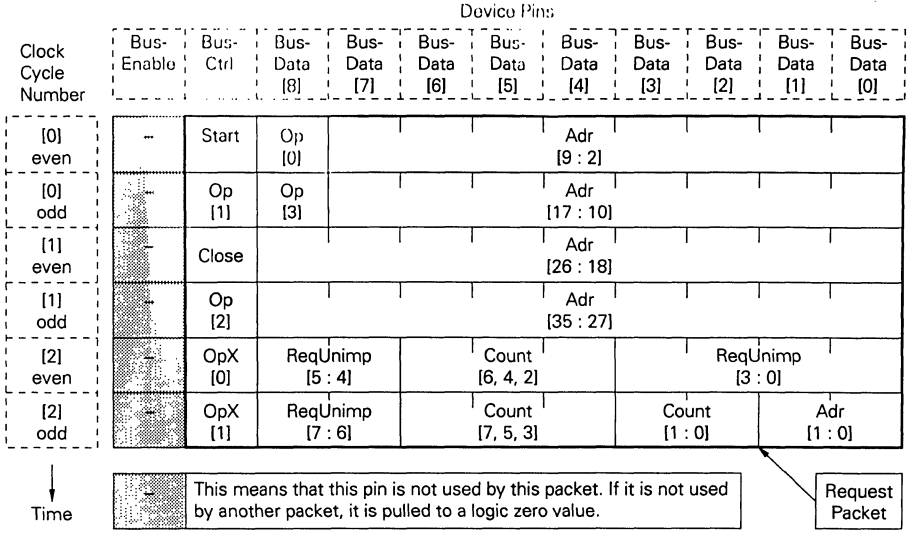
**Table 5-1. Transmitting/Receiving Devices for Packet Types**

| Packet Type | Initiating Devices | μPD488130L |
|---|---|---|
| Request Packet | Transmit | Receive |
| Acknowledge Packet | Receive | Transmit |
| Read Data Packet | Receive | Transmit |
| Write Data Packet | Transmit | Receive |
| Serial Address Packet | Transmit | Receive |
| Serial Control Packet | Transmit | Receive |
| Serial Mode Packet | Transmit | Receive |

### 5.2 Request Packet

The request packet format is shown in the following figure.

**Figure 5-1. Request Packet Format**




The vertical axis in all packet figures in the following sections shows time in units of clock cycles, with each clock cycle broken into even and odd bus ticks. The timing is relative, measured from the beginning of the packet.

### 5.2.1 Start Field

A device should start framing a request packet when it sees this bit asserted to a logical one and it is not looking for an acknowledge packet nor framing an earlier request packet.

### 5.2.2 Op[3:0], OpX[1:0] Fields

The command opcode also determines which packets (in addition to the request packet) will form the transaction. A detailed functional description of the actions that an RDRAM takes for each implemented command is provided in **"Rambus DRAM user's manual (Reference Manual)"**. The following table summarizes the functionality of each subcommand:

**Table 5-2. Subcommand Summary**

| SubCommand | Description |
|---|---|
| Rseq | Read sequential data from memory space. |
| Rnsq | Read non-sequential (random-access) data from memory space. |
| Wseq | Write sequential data to memory space. |
| Wnsq | Write non-sequential (random-access) data to memory space. |
| Wbns | Write non-sequential (random-access) data to memory space with non-contiguous byte masking. |
| Npb | Write data is from data packet. There is no bit mask. |
| Dpb | Write data is from data packet. The bit mask is in the MDReg. |
| Bpb | Write data is from data packet. The bit mask is also from the data packet. |
| Mpb | Write data is from MDReg. The bit mask is from the data packet. |
| Rreg | Read sequential data from register space. |
| Wreg | Write sequential data to register space. |
| WregB | Broadcast write with no Okay acknowledge permitted. |
| Alt | Alternate command (same function as the primary command - intended for use in future shared memory multiprocessor systems). |

The memory read commands are formed using the Rseq and Rnsq subcommands to select sequential or nonsequential (random) access. The Alt and "null" subcommands select between two equivalent command sets ("null" means no subcommand). The "Alt" subcommands are reserved for use in future shared memory multiprocessor systems.

- RrrrAaa  Rrrr = {Rseq, Rnsq}
  Aaa = {Alt, null}

The following table summarizes the available write commands and shows how they are formed from a 3×4 matrix of the Wwww and Bbb subcommands. The Alt and "null" subcommands have the same meaning as in the memory read commands.

- WwwwBbbAaa Wwww = {Wseq, Wnsq, Wbns}
  Bbb = {Npb, Dpb, Bpb, Mpb}
  Aaa = {Alt, null}

**Table 5-3. Write Commands**

| Bbb subcommand | Wwww subcommands | | |
|---|---|---|---|
| | Wseq (seqential-access with contiguous byte masking) | Wnsq (non-sequential-access) | Wbns (non-sequential-access with non-contiguous-byte-masking) |
| Npb | WseqNpb | WnsqNpb | WbnsNpb |
| Dpb | WseqDpb | WnsqDpb | WbnsDpb |
| Mpb | WseqMpb | WnsqMpb | WbnsMpb |
| Bpb | WseqBpb | WnsqBpb | Not implemented |

There are three Wwww subcommands. They control the accessing pattern and the use of non-contiguous byte masking.

- Wseq - octbyte blocks in the RDRAM core are accessed in sequential (ascending little-endien) address order. Contiguous byte masking is controlled with the Adr[2:0] and Count[2:0] fields of the request packet.

- Wnsq - octbyte blocks in the RDRAM core are accessed in non-sequential address order. The addresses for the octbyte blocks within the sensed row come from serial address packets which are received on the BusEnable pin.
  The address order is arbitrary.

- Wbns - octbyte blocks in the RDRAM core are accessed in non-sequential address order, as in the Wnsq subcommand. In addition, byte masks are transmitted with the write data, permitting arbitrary non-contiguous byte masking of this write data. The bytemask octbytes are not included in the total octbyte transfer count ; i.e. a Count[7:3] field of 31 implies 4 bitmask octbytes and 32 write data octbytes, for a data packet size of 36 octbytes.

There are four Bbb subcommands. They select the type of bit masking to be applied to the write data.

- Npb (no-per-bit) - There is no bit mask applied to the write data. The MDReg is not used or modified.

- Dpb (data-per-bit) - The MDReg is used as a bit mask, the write data comes from the data packet. The same bit mask is used for each octbyte. This is also called persistent bit masking. The MDReg is not modified.

- Mpb (mask-per-bit) - The bit mask comes from the data packet, the write data comes from the MDReg. The same data is used for each octbyte. This is also called color masking. The MDReg is not modified.

- Bpb (both-per-bit) - The bit mask and the write data come from the data packet. The MDReg is not used, but is modified as a side effect (the WwwwBpbAaa commands are used to load the MDReg for the WwwwDpbAaa and WwwwMpbAaa commands). This is also called non-persistent bit masking.
  The bitmask octbytes are included in the total octbyte transfer count ; i.e. a Count[7:3] field of 31 implies 16 bitmask octbytes and 16 write data octbytes.

#### 5.2.2.1 Op[3:0] and OpX[1:0] Fields for 16M RDRAM

The Op and OpX fields are summarized in the following table.

**Table 5-4. Op[3:0] and OpX[1:0] Fields - Command Encodings**

| Op[3:0] | OpX[1:0] = 00 | OpX[1:0] = 01 | OpX[1:0] = 10 | OpX[1:0] = 11 |
|---|---|---|---|---|
| 0000 | Rseq | Rnsq | Rsrv | Rsrv |
| 0001 | RseqAlt | RnsqAlt | Rsrv | Rsrv |
| 0010 | Rsrv | Rsrv | Rsrv | Rsrv |
| 0011 | Rsrv | Rsrv | Rsrv | Rsrv |
| 0100 | WseqNpb | WseqDpb | WseqBpb | WseqMpb |
| 0101 | WseqNpbAlt | WseqDpbAlt | WseqBpbAlt | WseqMpbAlt |
| 0110 | Rreg | Rsrv | Rsrv | Rsrv |
| 0111 | Wreg | Rsrv | Rsrv | Rsrv |
| 1000 | WnsqNpb | WnsqDpb | WnsqBpb | WnsqMpb |
| 1001 | WnsqNpbAlt | WnsqDpbAlt | WnsqBpbAlt | WnsqMpbAlt |
| 1010 | Rsrv | Rsrv | Rsrv | Rsrv |
| 1011 | Rsrv | Rsrv | Rsrv | Rsrv |
| 1100 | WbnsNpb | WbnsDpb | Rsrv | WbnsMpb |
| 1101 | WbnsNpbAlt | WbnsDpbAlt | Rsrv | WbnsMpbAlt |
| 1110 | Rsrv | Rsrv | Rsrv | Rsrv |
| 1111 | WregB | Rsrv | Rsrv | Rsrv |

The command opcode determines whether the other fields of the request packet are implemented (Imp) or unimplemented (Unimp). This is summarized in the table below.

**Table 5-5. 16M RDRAM Request Packet Fields - Imp or Unimp**

| Command | Adr[35:3] | Adr[2] | Adr[1:0] | Count[7:3] | Count[2] | Count[1:0] | Close | ReqUnimp[7:0] |
|---|---|---|---|---|---|---|---|---|
| Rsrv | Unimp(0.0) | Unimp(0) | Unimp(00) | Unimp(0.0) | Unimp(1) | Unimp(11) | Unimp(0) | Unimp(0.0) |
| Rseq | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| Rnsq | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| RseqAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| RnsqAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqNpb | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqDpb | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqBpb | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqMpb | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqNpbAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqDpbAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqBpbAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| WseqMpbAlt | Imp | Imp | Imp | Imp | Imp | Imp | Imp | Unimp(0.0) |
| Rreg | Imp | Imp | Unimp(00) | Unimp(0.0) | Unimp(1) | Unimp(11) | Unimp(0) | Unimp(0.0) |
| Wreg | Imp | Imp | Unimp(00) | Unimp(0.0) | Unimp(1) | Unimp(11) | Unimp(0) | Unimp(0.0) |
| WnsqNpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqDpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqBpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqMpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqNpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqDpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqBpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WnsqMpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsNpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsDpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsMpb | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsNpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsDpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WbnsMpbAlt | Imp | Unimp(0) | Unimp(00) | Imp | Unimp(1) | Unimp(11) | Imp | Unimp(0.0) |
| WregB | Imp | Imp | Unimp(00) | Unimp(0.0) | Unimp(1) | Unimp(11) | Unimp(0) | Unimp(0.0) |

### 5.2.3 Adr[35:0] Field

The Adr field is used as either a memory or register space address depending upon the OP[3:0] and OpX[1:0] fields. Devices extract a portion of the Adr field to match against their DeviceId register (IdMatch), thus selecting the device to which the request is directed. The remainder of the Adr field accesses the desired region of the device's memory or register space. The memory read and write commands and the Rreg and Wreg commands will only take place if there is an IdMatch. The IdMatch criteria is ignored for the WRegB commands, with all responding devices performing the required actions.

The Rambus protocol uses quadbyte resolution in the data packet for register space read and write commands; i.e. one quadbyte is the smallest data item that may be transferred, and all transfers are an integral number of quadbytes. The Adr[35:2] field is the quadbyte address. The Adr[1:0] field is Unimp for these commands, and should be driven with "00" by initiating devices.

The Rambus protocol uses octbyte resolution in the data packet for memory space read and write commands; i.e. one octbyte is the smallest data item that may be transferred, and all transfers are an integral number of octbytes. The Adr[35:3] field is the octbyte address.

Some commands use the Adr[2:0] field to specify contiguous byte masking. Refer to **"Rambus DRAM user's manual (Reference Manual)"**.

### 5.2.4 Count[7:0] Field

The following table summarizes the transfer count ranges for 16M RDRAMs:

**Table 5-6. Transfer Count Summary**

| Count Range | μPD488130L |
|---|---|
| Maximum count for memory space | 32 octbytes |
| Minimum count for memory space | 1 octbyte |
| Maximum count for register space | 1 quadbyte |
| Minimum count for register space | 1 quadbyte |

Register space read and write commands use a transfer count of one quadbyte, regardless of the Count[7:0] field value.

Memory space read and write commands specify the number of octbytes to be transferred with the Count[7:3] field. An offset-by-one-encoding is used so that "00000" specifies one octbyte, "00001" specifies two octbytes, and so on up to "11111" which specifies thirty-two octbytes. The transfer count does include the octbytes containing bitmasks (for commands using the Bpb subcommand). The transfer count does not include the octbytes containing non-contiguous ByteMasks (for commands using the Wbns subcommand).

Some commands use the Count[2:0] field to specify contiguous byte masking. Refer to **"Rambus DRAM user's manual (Reference Manual)"**.

Memory space transactions to RDRAMs are not allowed to cross internal row address boundaries within the device. Attempts to do so have Undef (undefined) results. These row boundaries are at 2kbyte intervals for 16M RDRAMs.

### 5.2.5 Adr[2:0] and Count[2:0] Fields for Contiguous Byte Masking

An initiating device wishing to transfer an arbitrary number of contiguous bytes to a starting address on an arbitrary byte boundary may do so with the Adr[2:0] and Count[2:0] fields for some of the commands. These commands include:

- RrrrAaa
- WseqBbbAaa

The transfer count and starting address are given by:

- MasterCount[7:0] specifies the number of bytes which the master device wishes to transfer.
- Adr[35:0] specifies the starting byte address (this is the same as the Adr[35:0] field in the request packet)

Where the convention used by the initiating device for the count is that Master-Count[7:0] = "00000000" means one byte, MasterCount[7:0] = "00000001" means two bytes and MasterCount[7:0] = "11111111" means 256 bytes (an offset-by-one encoding; the data block count is equal to MasterCount[7:0]+1).

The initiating device converts this internal count value into a value for the request packet with the following formula. Little-endien byte addressing is used for specifying bytes within octbytes.

Count[7:0] = Adr[2:0] + MasterCount[7:0] (Eq 5-1)

Where "+" denotes unsigned integer addition of two bit fields (short fields are zero-extended on the left). If the value of Adr[2:0] + MasterCount[7:0] is greater than 255 (it may be as much as 262), then the initiating device must break the request into two transactions.

The Adr[2:0] and Count[2:0] field generate masks for individual bytes within an octbyte. The Adr[35:3] and Count[7:3] field have the octbyte resolution previously described. The following tables show how the byte masks are generated. In the case of memory read transactions, the byte masks that are generated do not affect the data that is returned by the RDRAM; all data bytes in the first and last octbytes are returned in the read data packet.

In the case of memory write transactions, ByteMaskLS[7:0] applies to the first octbyte at Mem[AV][7:0][8:0]. Byte MaskMS[7:0] applies to the last octbyte at Mem[AV+CV][7:0][8:0]. All intermediate octbytes use a byte mask of 11111111 (a one means the byte is written, a zero means it is not). Here AV is the value of the Adr[35:3] field when interpreted as an unsigned, 33 bit integer, and CV is the value of the Count[7:3] field when interpreted as an unsigned, 5 bit interger. If the Count[7:3] is "00000" (one octbyte), the ByteMaskLS[7:0] and ByteMaskMS[7:0] masks are logically 'anded' together to give the effective byte mask.:

**Table 5-7. Adr[2:0] to ByteMaskLS[7:0] Encoding**

| Adr[2:0] | ByteMaskLS[7:0] | Adr[2:0] | ByteMaskLS[7:0] |
|---|---|---|---|
| 000 | 11111111 | 100 | 11110000 |
| 001 | 11111110 | 101 | 11100000 |
| 010 | 11111100 | 110 | 11000000 |
| 011 | 11111000 | 111 | 10000000 |

**Table 5-8. Count[2:0] to ByteMaskMS[7:0] Encoding**

| Count[2:0] | ByteMaskMS[7:0] | Count[2:0] | ByteMaskMS[7:0] |
|---|---|---|---|
| 000 | 00000001 | 100 | 00011111 |
| 001 | 00000011 | 101 | 00111111 |
| 010 | 00000111 | 110 | 01111111 |
| 011 | 00001111 | 111 | 11111111 |

The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)".**

### 5.2.6 Close Field

The Close field causes a currently accessed row to be explicitly restored (written back to the core if its Dirty flag for that bank is set) after the current access has completed. This reduces the latency of a subsequent access to a different row of memory space in the same bank of that device; i.e. the $t_{ReterSensedClean}$ Nack timing is used rather than the $t_{RetrySensedDirty}$ timing.

**Table 5-9. Close Field Encodings**

| Close Field | Description |
|---|---|
| 0 | Don't restore the currently accessed row of memory. It may be left sensed and either clean or dirty. |
| 1 | Restore the currently accessed row of memory if its Dirty flag is set. It will be left sensed and clean. |

### 5.2.7 ReqUnimp[7:0] Fields

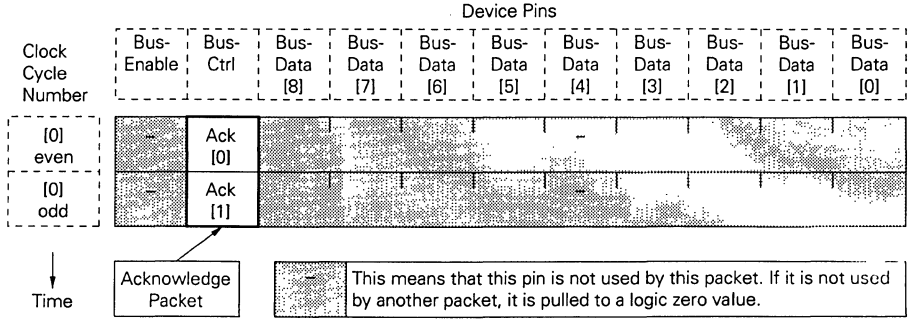
These fields are unimplemented (Unimp) in the request packet. They should be driven as zeroes by initiating devices which satisfy this Version of the Rambus protocol.

Responding devices which satisfy this Version of the Rambus protocol should ignore these fields and process the request according to the content of the Start, Op[3:0], OpX[1:0], Adr[35:0], Count[7:0], and Close fields. The protocol Version number of a device is held in the DeviceType register.

## 5.3 Acknowledge Packet

The Ack[1:0] field carries the acknowledge encoding from the responding device(s) to the initiating device and any other listening devices. The following figure shows the format of the acknowledge packet.

**Figure 5-2. Acknowledge Packet Format**




The following table summarizes the four combinations of the Ack[1:0] field. The Ack3 combination is Undef. The Okay combination indicates that the read or write access to the specified space will take place.

When a responding device acknowledges a request with a Nack, then there will be no immediate change in the state of the device's memory space or register space. The responding device will take the appropriate steps to make the requested region of memory or register space accessible when the initiating device makes a subsequent request. The initiating device will need to wait some device-dependent length of time until the requested region is available.

There are three possible reasons for an RDRAM to respond with Nack. They are summarized below. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

- $t_{PostMemWriteDelay}$ or $t_{PostRegWriteDelay}$ violation
- RowMiss (this causes a delay of $t_{RetrySensedClean}$ or $t_{RetrySensedDirty}$)
- ongoing refresh (this causes a delay of up to $t_{RetryRefresh}$)

**Table 5-10. Ack[1:0] Encodings**

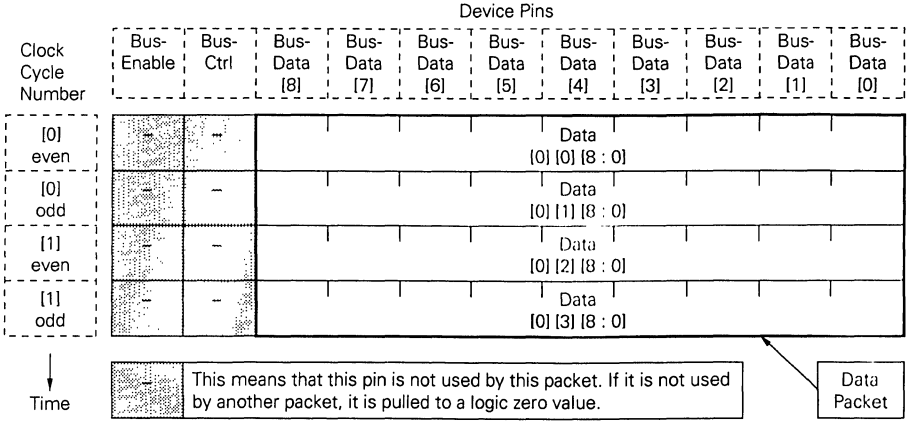
| Commands allowed to use the Ack Combination | Ack [1:0] | Name | Description | Spec Undef |
|---|---|---|---|---|
| All commands | 00 | Nonexistent | Indicates passive acceptance of the request (WregB), or indicates that the addressed device did not respond (all other commands). | Spec |
| All commands but WregB | 01 | Okay | Indicates that the request was accepted by the addressed by the addressed (responding) device. | Spec |
| All commands | 10 | Nack | Indicates that the request could not be accepted because the state of the responding device prevented an access at the fixed timing slot. | Spec |
| All commands but WregB | 11 | Ack3 | This should not be returned by this responding device. Initiating devices will, when presented with this combination, have an undefined response. | Undef |

## 5.4 Data Packet

The following figure shows the format of a data packet for register space read and write commands. It consists of 1 quadbyte driven on the BusData[8:0] wires for RDRAMs.

Other responding devices may support data packet lengths longer than one quadbyte.

**Figure 5-3. Data Packet Format (Register Space)**



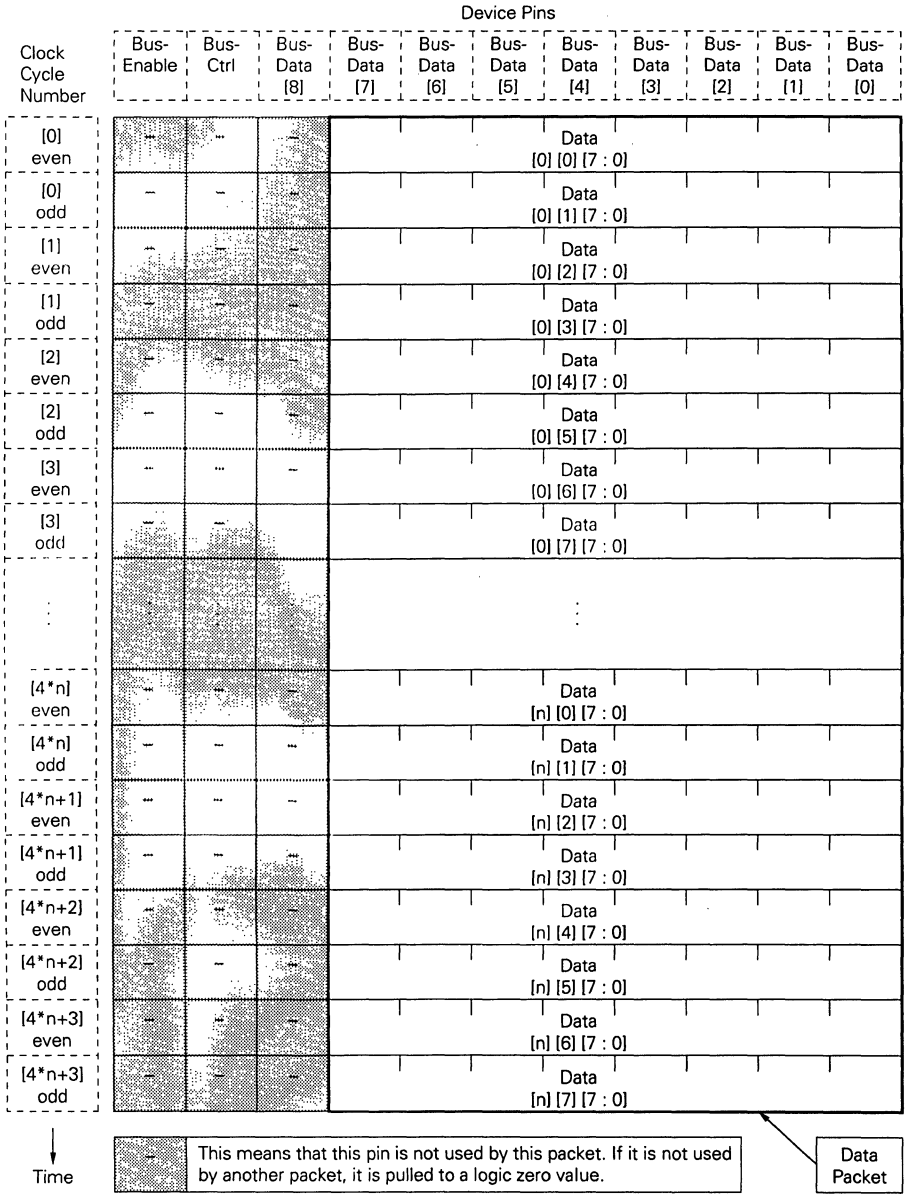

The following figure shows the format of a data packet for memory space read and write commands. For most of these commands, it consists of 1 to 32 octbytes driven on the BusData[7:0] wires. BusData [8] is not used by this packet. In the figure, "n" is either the CV value (if the transaction is allowed to complete) or the last count value (if the transaction is terminated prematurely by the serial control packet). "CV" is the value of the Count[7:3] field when interpreted as an unsigned, 5 bit integer.

The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

### 5.4.1 MD Reg [7:0] [7:0]=U

This register holds the write data or mask for the persistent per-bit operations (Dpb & Mpb). The MDreg need not implement the ninth bits when the RDRAM is 8-bit wide.

**Figure 5-4. Data Packet Format (Memory Space)**



### 5.5 Serial Address Packet Format

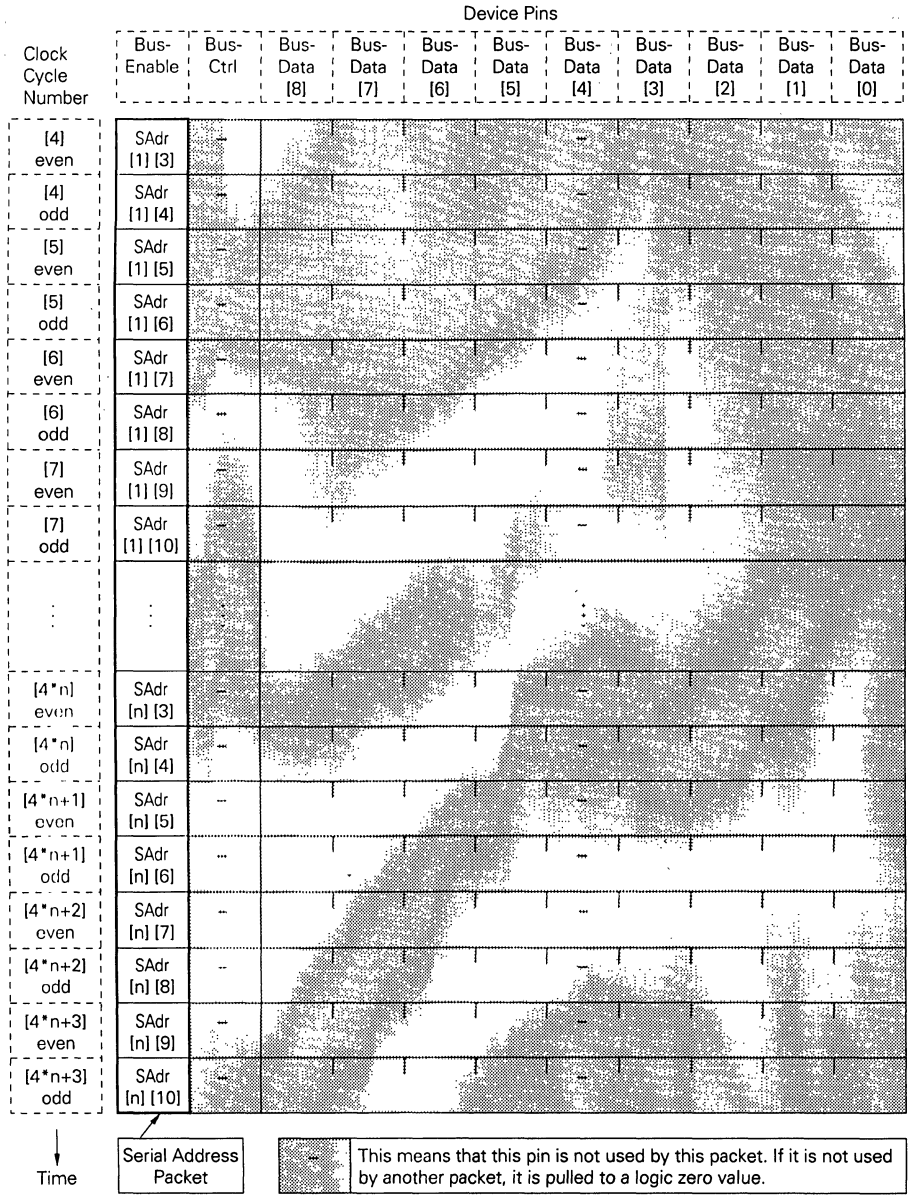
The serial address packet is transmitted by the initiating device and received by the responding devices. It provides eight low-order address bits for each octbyte which is accessed in memory space (a non-sequential or random-access transfer). These eight address bits are transferred serially on the BusEnable pin of the RDRAM, and are thus called a serial address. Each eight bit serial address accesses an octbyte of data within the RowSenseAmpCache of one of the two banks of the RDRAM. The complete set of serial addresses transmitted by the initiating device during the transaction are referred to as a serial address packet. The commands which use this packet are the RnsqAaa, WnsqBbbAaa, and WbnsBbbAaa classes of commands.

The high order bits for each octbyte are provided by the Adr[35:11] address bits from the request packet. The low-order address bits for the first octbyte are Adr[10:3], also from the request packet. The low-order address bits for octbytes [n:1] are provided by the serial address packet. As before, "n" is either the CV value (if the transaction is allowed to complete) or the last count value (if the transaction is terminated prematurely by the serial control packet). "CV" is the value of the Count[7:3] field when interpreted as an unsigned, 5 bit integer. The detailed functional description is provided in **"Rembus DRAM user's manual (Reference Manual)"**.

**Table 5-11. Serial Address Fields (i = n:1)**

| Serial Address Field | Description | Unimp Imp |
|---|---|---|
| SAdr[i][10:3] | Low-order address bits for each octbyte. | Imp |

**Figure 5-5. Serial Address Packet Format**



### 5.5.1 Serial Control Packet Format

The serial control packet is transmitted by the initiating device and received by the responding devices. It provides for the early termination of a memory space read or write transaction (before the specified data count in the Count[7:3] field has elapsed). It consists of eight bits transferred serially on the BusCtrl pin of the device, thus it is referred to as a serial control packet. The eight bits have the same timing alignment as the serial address packet. The commands which use this packet are all of those which access memory space. The register read and write commands do not use the serial control packet. The 16M RDRAM implements this packet.

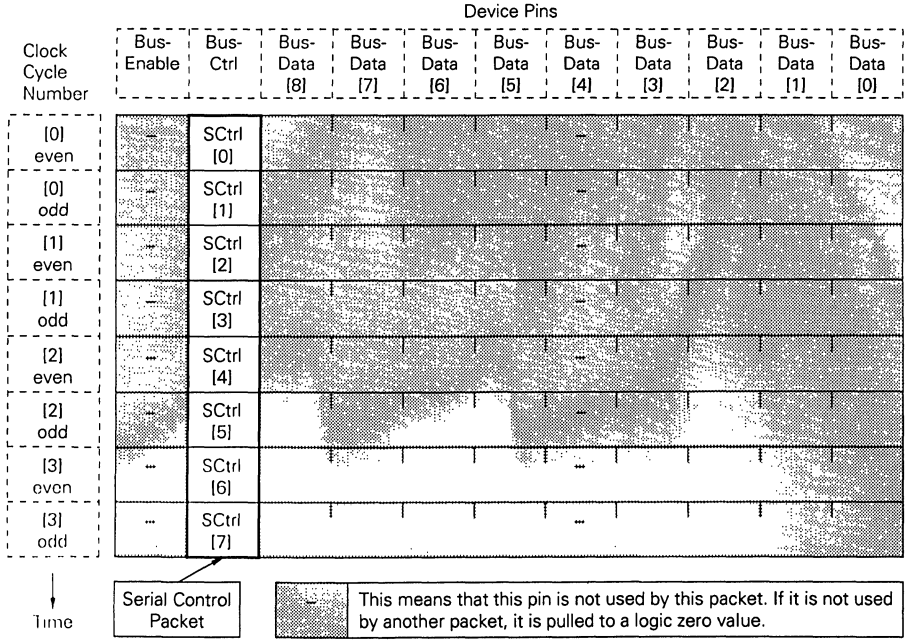
The termination occurs on octbyte data packet boundaries. The next figure shows the format of the serial control packet. The following table summarizes the function of the bits within the serial control packet. Note that the bits in the even bus ticks must be zero in order for framing to work properly (otherwise, one of these bits would be interpreted as the Start bit of a new request packet). The SCtrl[5] bit is used to control termination, and the other three odd bus tick bits are unimplemented.

**Table 5-12. Serial Control Fields**

| Serial Control Fields | Description | Unimp Imp |
|---|---|---|
| SCtrl[0] | This bit must be a zero due to framing requirements. | Imp |
| SCtrl[1] | unimplemented | Unimp(0) |
| SCtrl[2] | This bit must be a zero due to framing requirements. | Imp |
| SCtrl[3] | unimplemented | Unimp(0) |
| SCtrl[4] | This bit must be a zero due to framing requirements. | Imp |
| SCtrl[5] | 0 means don't terminate the current access.<br>1 means terminate the current access. | Imp |
| SCtrl[6] | This bit must be a zero due to framing requirements. | Imp |
| SCtrl[7] | unimplemented | Unimp(0) |

If a memory read transaction (RrrrAaa) is terminated by asserting the SCtrl[5] bit to a logical one, the data octbyte with which it is associated is not transmitted by the responding device. The initiating device may start a new transaction once the transmission of the read data packet has ceased. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

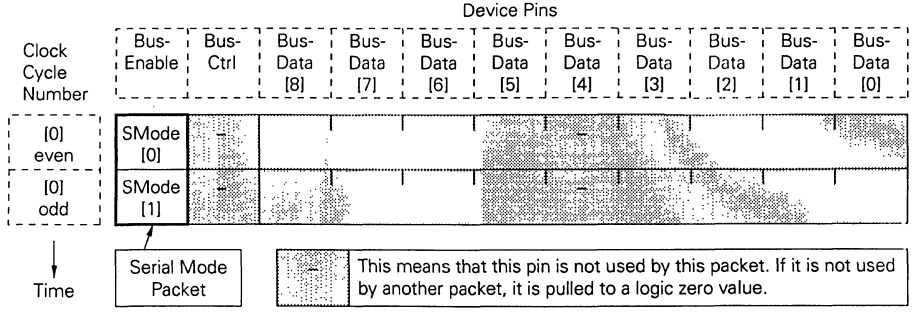
**Figure 5-6. Serial Control Packet Format**



### 5.5.2 Serial Mode Packet Format

The serial mode packet transmitted by initiating devices, and received by responding device. Its format is shown in the following figure.

**Figure 5-7. Serial Mode Packet Format**




The serial mode packet modifies the state of the Count00[7:0] and Count11[7:0] counters.

These counters cause operating mode transitions when they reach special values. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

A serial mode packet with the SMode[1:0] field set to 00 is the default. Most transitions are caused by blocks of sequential serial mode packets, each with the SMode[1:0] field set to 11. The serial mode packets should never set SMode[1:0] field to 01 or 10. This is because in some of the operating modes, the clock generator is unlocked (the frequency is correct but not the phase). When this happens, the BusEnable receiver is unable to discriminate anything other than long pulses of zeros or ones. Because the frequency of the clock generator is correct, it can count the length of these pulses with moderate accuracy.

**Table 5-13. Serial Mode Fields**

| SMode[1:0] | Description | Spec/Rsrv/ Undef |
|---|---|---|
| 00 | Increments Count00[3:0], clears Count11[7:0]. | Spec |
| 01 | – | Undef |
| 10 | – | Undef |
| 11 | Increments Count11[7:0], clears Count00[3:0] | Spec |

## 6. State Diagram

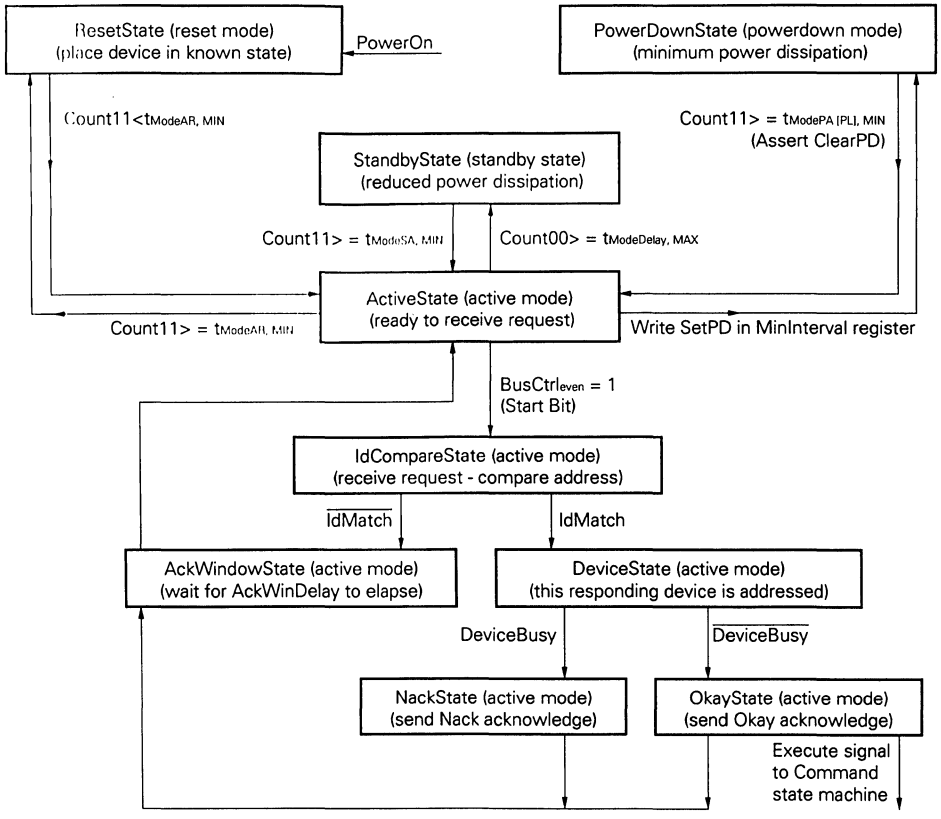
The following figure is a state diagram of the Frame state machine. The operating mode of the device depends upon which of the nine states it is in:

- reset mode - ResetState
- powerdown mode - PowerDownState
- standby mode - StandbyState
- active mode - ActiveState, IdCompareState, DeviceState, OkayState, NackState, AckWindowState

This section will only discuss the first three states (ResetState, StandbyState, ActiveState). The remaining five states which are shown shaded in the state diagram (IdCompareState, DeviceState, OkayState, NackState, AckWindowState) will be dealt with in the **"Rambus DRAM user's manual (Reference Manual)"**.

The device will enter ResetState when power is initially applied (PowerOn). In ResetState, the device will be in the reset operating mode, in which all control registers assume a known state. If power has just been applied, the device will pass through ActiveState and settle in StandbyState, and remain there until serial mode packets are received from an initiating device.

**Figure 6-1. Frame State Machine - State Diagram**



ActiveState is the state in which all decisions are made to transition to the states for the other operating modes. From here, the device will also enter the transaction-framing states. Refer to **"Rambus DRAM user's manual (Reference Manual)"**.

After poweron, the device will re-enter ResetState when the value of the Count11[7:0] counter is greater than or equal to $t_{ModeAR,MIN}$. The device will leave ResetState when the value of the Count11[7:0] counter is less than $t_{ModeSA,MIN}$. This will happen when an SMode[1:0] field of 00 is received, causing the Count11[7:0] counter to clear.

The device will enter StandbyState when the value of the Count00[3:0] counter is greater than or equal to $t_{ModeDelay,MAX}$. The device will leave StandbyState when the value of the Count11[7:0] counter is greater than or equal to $t_{ModeSA,MIN}$.

## 6.1 Parameters for Operating Mode Transitions

The following table summarizes the parameter values associated with operating mode transitions of a responding device. A minimum and maximum value are given for the parameters to account for implementation differences. In all cases, the SMode[1:0] field of the consecutive serial mode packets must have the value 11 to cause an operating mode transition (with the exception of the $t_{ModeDelay,MAX}$ as mentioned in the previous section). Initiating devices must use values within the minimum and maximum SMode packet count requirements shown above to control operating mode transitions.

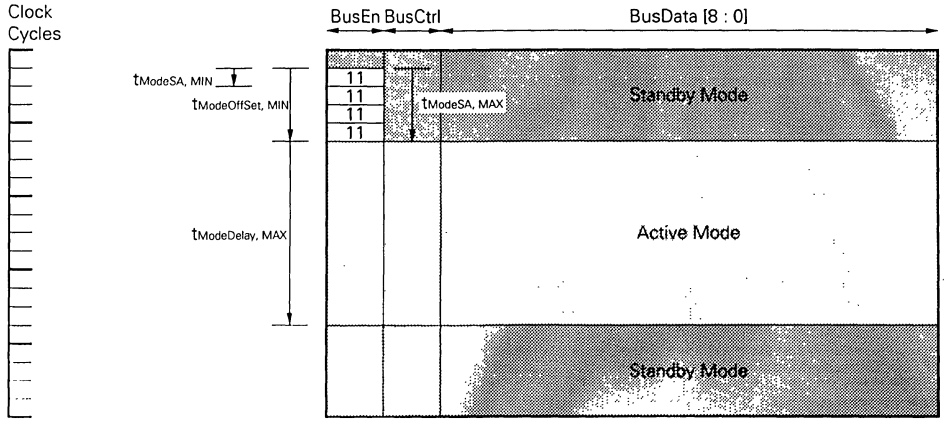
**Table 6-1. Responding Device Parameters for Operating Mode Transitions**

| Count Parameter Name | Minimum (clock cycles) | Maximum (clock cycles) | Description |
|---|---|---|---|
| $t_{ModeSA}$ | 1 | 4 | Number of SMode packets to cause a transition from Standby-Mode to ActiveMode |
| Rsrv | 5 | 9 | Reserved for future functionality |
| Undef | 10 | 15 | Undefined |
| Rsrv | 21 | 189 | Reserved for future functionality |
| Undef | 190 | 207 | Undefined |
| Rsrv | 225 | 253 | Reserved for future functionality |
| Undef | 254 | 271 | Undefined |
| $t_{ModeAR}$ | 272 | – | Number of SMode packets to cause a transition from Active-Mode to ResetMode |
| $t_{ModeOffSet}$ | 4 | – | Offset from beginning of SMode packet to request packet for standby to active transition |
| $t_{ModeDelay}$ | – | 10 | Delay from end of SMode packet to request packet for standby to active transition |

## 6.2 Standby Mode and Active Mode

The following figure shows the basic transitions between active and standby modes in response to serial mode packets

**Figure 6-2. Basic ActiveMode/StandbyMode Transitions**




This is a timing diagram, with time increasing in the downward direction. The time scale is in clock cycles, as shown on the left scale. The value of each of the eleven low-swing signal pins of the responding device is shown with the assumption that $t_{TR}$ is zero (the responding device is located at the master end of the Channel).

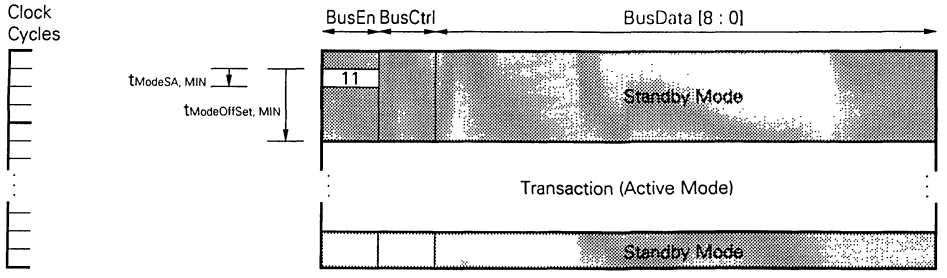
Serial mode packets with an SMode[1:0] field are shown as a box with a "11" label in the BusEn column. The BusEnable defaults to a logical zero value. The initiating device has transmitted $t_{ModeSA,MAX}$ serial mode packets with SMode[1:0] equal to 11 (this is the longest sequence permitted for invoking a standby to active transition). After the first $t_{ModeSA,MIN}$ serial mode packets, the device begins the transition to active mode. It reaches active mode after $t_{ModeOffSet,MIN}$ clock cycles after the start of the first serial mode packet. It remains there for $t_{ModeOffSet,MAX}$ clock cycles after the last serial mode packet.

The responding device is in active mode when it begins framing the request packet. A transaction may begin in any of the clock cycles with the light shading above (labeled "Active Mode").

If the serial mode packet(s) causing a standby to active mode transition are not followed by a transaction with $t_{ModeOffSet,MAX}$ clock cycles after the last serial mode packet, then the responding device will return to standby mode.

The next figure shows the case in which a transaction is started as early as possible after a serial mode packet which causes a standby to active mode transition.

**Figure 6-3. ActiveMode/StandbyMode Transition - Early Transaction**

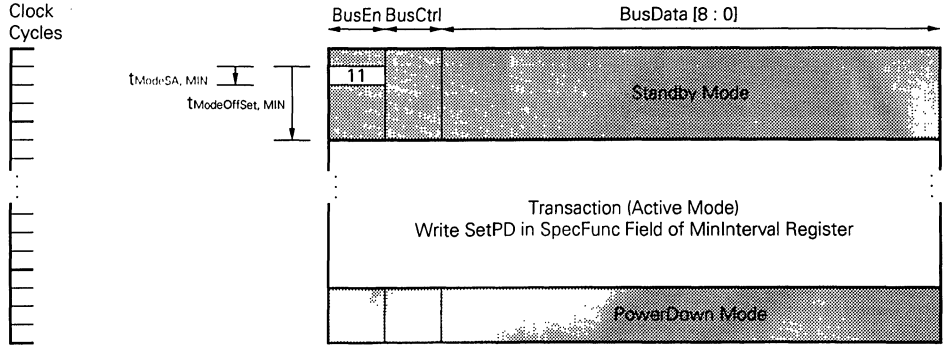



A transaction is composed of packet types other than serial mode packets, and will be defined in the next chapter. These other packet types lie entirely inside the heavy black box in the above figure. When a transaction has completed, the device returns to standby mode. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

### 6.3 PowerDown Mode

When responding devices watch the bus, their interface circuits consume power. Standby mode reduces this power by trading a small latency for reduced power. While in standby mode, the device's receivers are inactive.

**Figure 6-4. ActiveMode to PowerDownMode Transition**




Power may be greatly reduced using the powerdown mode at the expense of moderate latency. In powerdown mode, the device's receivers and clock circuitry are inactive. The device may deactivate other power-hungry circuits also. Power-up enables all these circuits and makes the device available for transactions once the clock circuitry resynchronizes and stabilizes.

Powerdown mode is controlled by the PD bit. This bit is not directly accessible in the register space. Instead, the "SetPD" combination is written to the SpecFunc field in the MinInterval register, as indicated in the above figure. When this is done, the RDRAM performs the following operations:

- Restore and precharge the RowSenseAmpCache for both banks
- Disable the clock generator
- Disable all DC current sources except for a special BusEnable receiver

When these operations have completed, the RDRAM is in powerdown mode. It will consume power for refresh (which is performed with the SIn/SOut TTL pins), and it will consume a small amount of power watching the BusEnable pin waiting for a serial mode packet of the proper length. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

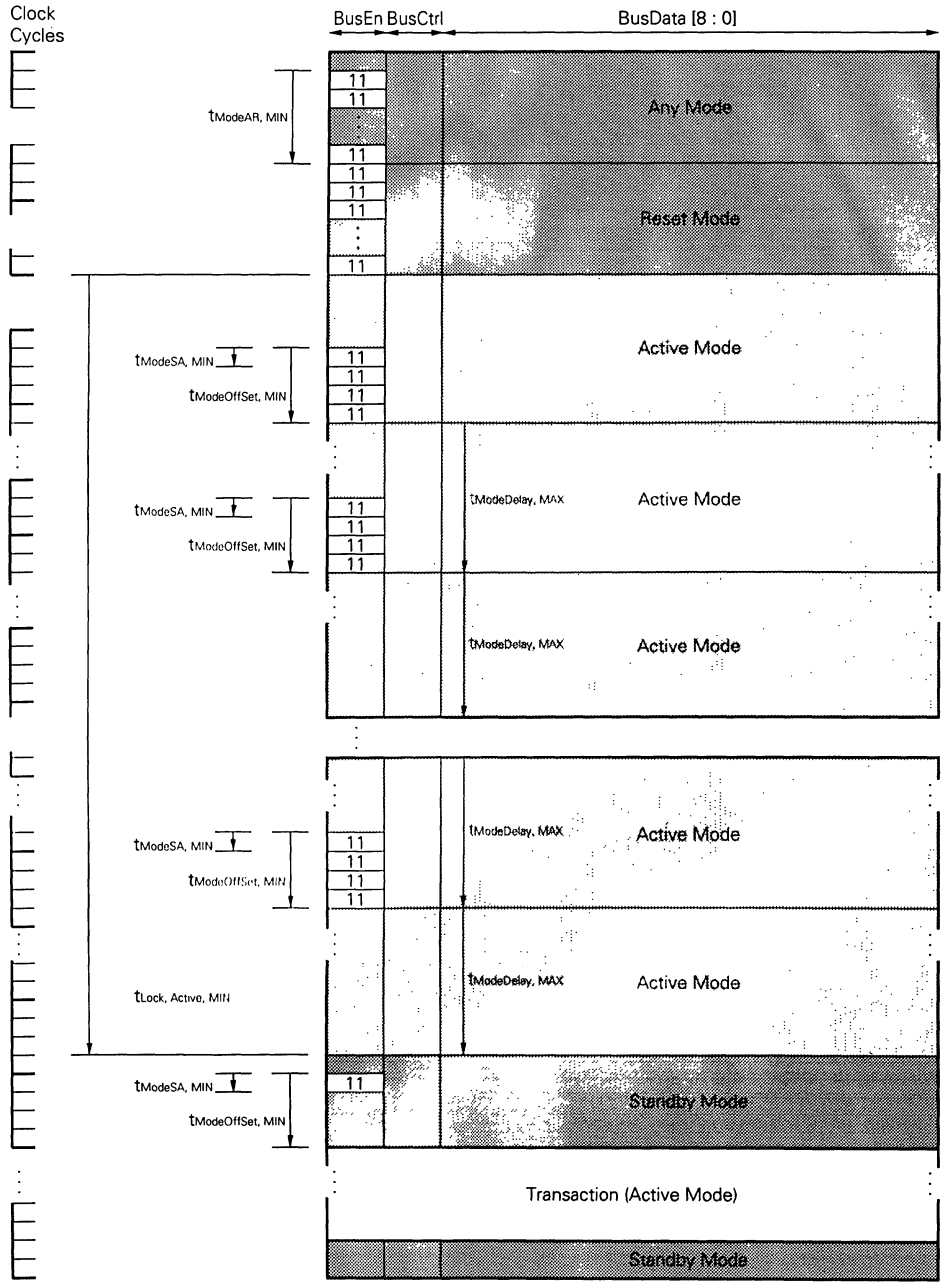
### 6.4 ResetMode

Reset mode is entered when a consecutive sequence of $t_{ModeAR,MIN}$ serial mode packets with a value of 11 are seen by a responding device (shown in the following figure). In reset mode, all devices enter a known state from which they may be Initialized. The device remains in reset mode for as long as serial mode packets with 11 value are received. When one or more serial mode packets with a value of 00 are seen, the responding device enters the active mode state.

Although devices enter the active mode state immediately, their clock circuitry requires a time $t_{Lock,MIN}$ to resynchronize. Initiating devices must wait this long after the transition out of reset mode before starting any transactions.

Note that in order to keep the devices in active mode during this synchronization process, it is necessary to provide a burst of serial mode packets every $t_{ModeDelay,MAX}$ clock cycles. This burst is shown as $t_{ModeSA,MAX}$ in length, but may, of course, be as short as $t_{ModeSA,MIN}$. If the device is not kept in active mode during synchronization, then the synchronization process requires $t_{Lock,MIN}$ clock cycles.

**Figure 6-5. ResetMode to ActiveMode Transition**



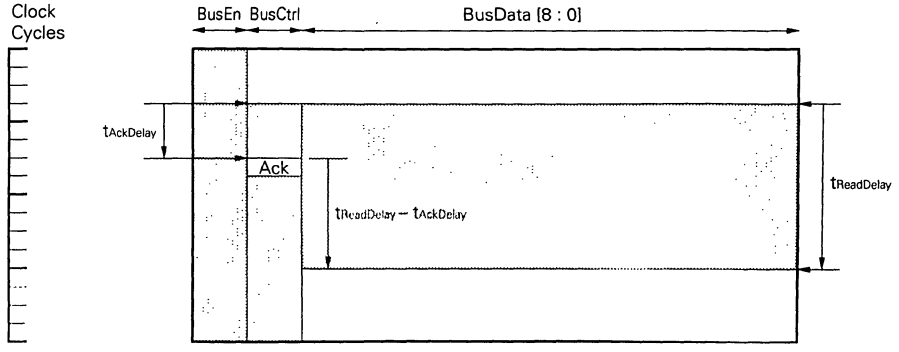
## 7. Transactions

### 7.1 Read Transactions

The following figure shows the basic form of a memory space or register space read transaction. There are request and acknowledge packets, with the same $t_{AckDelay}$ and $t_{AckWinDelay}$ timing constraints as already discussed ($t_{AckWinDelay}$ will not be shown explicitly on any further transaction diagrams in this document.

When the responding device transmits an Okay acknowledge packet to the initiating device, it will also transmit a data packet with read data. This packet is sent a time $t_{ReadDelay}$ after the end of the request packet. The $t_{ReadDelay}$ value is in $t_{CYCLE}$ units and is programmed into the ReadDelay field of the Delay register of each responding device. It is not required to be the same for all devices within a Rambus system, but the difference ($t_{ReadDelay}$ - $t_{AckDelay}$) is required to be the same. This allows initiating devices to use the acknowledge packet to determine when the read data packet begins. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 7-1. Read Transaction**



## 7.2 Write Transactions

The following figure shows the basic form of a memory space or register space write transaction. There are request and acknowledge packets, with the same $t_{AckDelay}$ and $t_{AckWinDelay}$ timing constraints as already discussed.

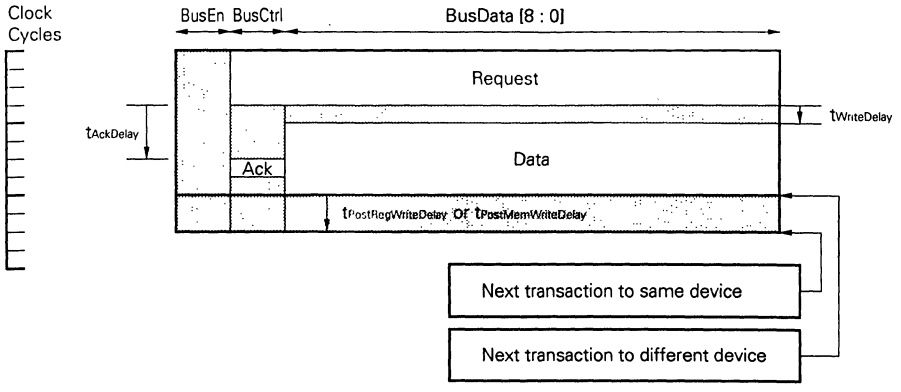
When the initiating device transmits a request packet to the responding devices, it will also transmit a data packet with write data. This packet is sent a time $t_{WriteDelay}$ cycles after the end of the request packet. The $t_{WriteDelay}$ is in $t_{CYCLE}$ units and is programmed into the WriteDelay field of the Delay register of each responding device. It is required to be the same for all devices within a Rambus system. A responding device will see the same $t_{WriteDelay}$ interval between the request and write data packets whether the device is on the Primary Channel or on a Secondary Channel.

If the responding device returns an Okay acknowledge packet, then the transaction is complete at the end of the acknowledge window or at the end of the write data packet, whichever is later. The next request packet can be transmitted in the following clock cycle except for the case in which a register or memory space write to a device is followed by any other transaction to that device. In that case, one of the following two intervals must be inserted between the two transactions, where the memory or register case depends upon the first transaction.

- $t_{PostRegWriteDelay}$ if the current transaction is a register space access
- $t_{PostMemWriteDelay}$ if the current transaction is a memory space access

If the responding device returns a Nack or Nonexistent acknowledge packet for a write command, then no write data packet is required by the responding device. The current transaction is complete at the end of the acknowledge window, or when the initiating device stops transmitting the write data packet, whichever is later. The next request packet can be transmitted in the following clock cycle. For the case of a Nack or Nonexistent, the initiating device must terminate the write data packet before another initiating device is given control of the Rambus Channel for a transaction. This is part of the arbitration mechanism used by the initiating devices. The arbitration mechanism is not specified in this document because it does not use the Rambus Channel. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 7-2. Write Transaction**



### 7.3 Read Transactions with Serial Address Packet

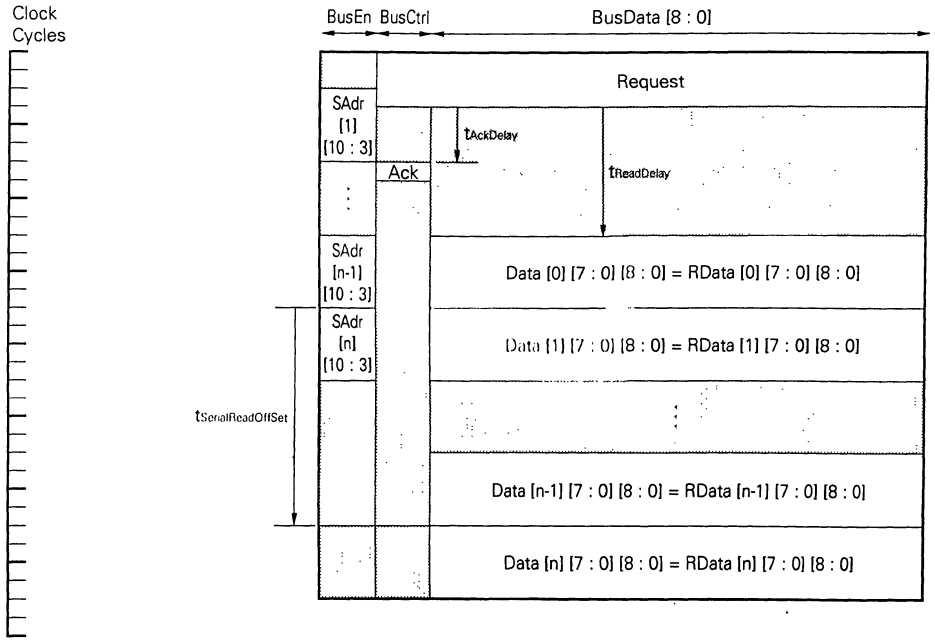
The following figure shows a memory space read transaction for a command which uses the serial address packet. For a transaction which moves (n+1) octbytes of read data, the serial address packet will be (4 × n) clock cycles in length (recall that the low-order address bits for the first octbyte of read data come from the request packet).

Each serial address subpacket (each SAdr[i][10:3] field) is transmitted by the initiating device a time $t_{SerialReadOffSet}$ clock cycles before the octbyte of read data to which it corresponds. This means that the serial address packet will move with the read data packet, with a constant offset.

- $t_{SerialReadOffSet}$ is the delay from the beginning of a serial address subpacket to the beginning of the read data subpacket (octbyte) with which it is associated.

The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 7-3. Read Transaction with Serial Address Packet**



## 7.4 Write Transactions with Serial Address Packet

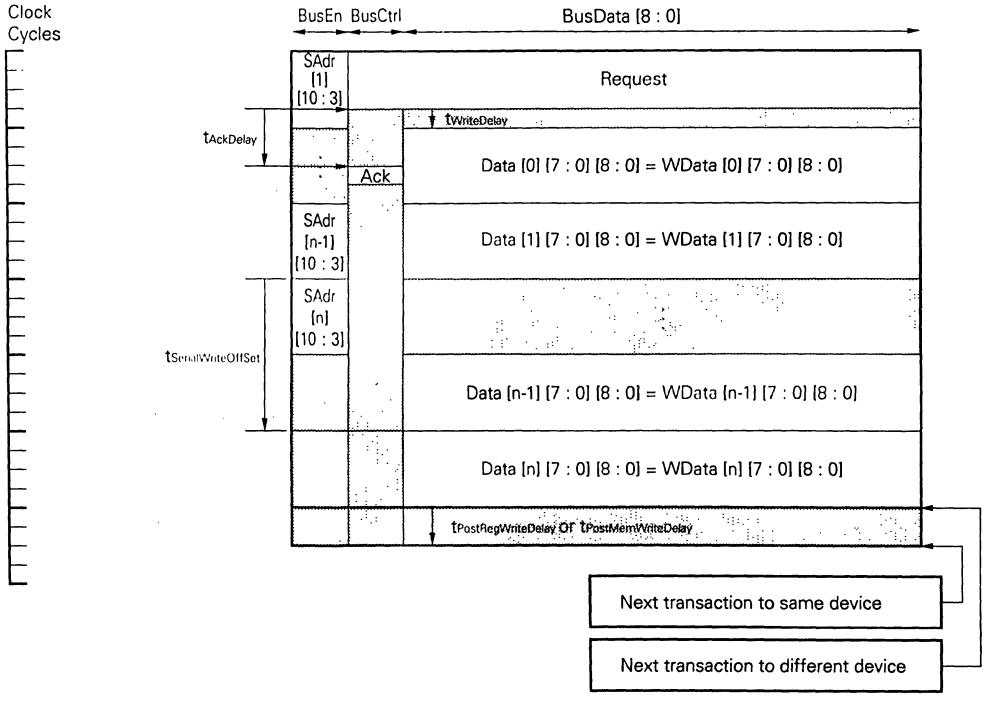
The following figure shows a memory space write transaction for a command which uses the serial address packet. For a transaction which moves (n+1) octbytes of write data, the serial address packet will be (4 × n) clock cycles in length (recall that the low-order address bits for the first octbyte of write data come from the request packet).

Each serial address subpacket (each SAdr[i][10:3] field) is transmitted by the initiating device a time $t_{SerialWriteOffSet}$ clock cycles before the octbyte of write data to which it corresponds. This means that the serial address packet will move with the write data packet, with a constant offset.

- $t_{SerialWriteOffSet}$ is the delay from the beginning of a serial address subpacket to the beginning of the write data subpacket (octbyte) with which it is associated.

Note that this offset interval is measured at the initiating device or the responding device; it will be the same at either point since the serial address packet and write data packet are moving in the same direction - from initiating device to responding device.

**Figure 7-4. Write Transaction with Serial Address Packet**



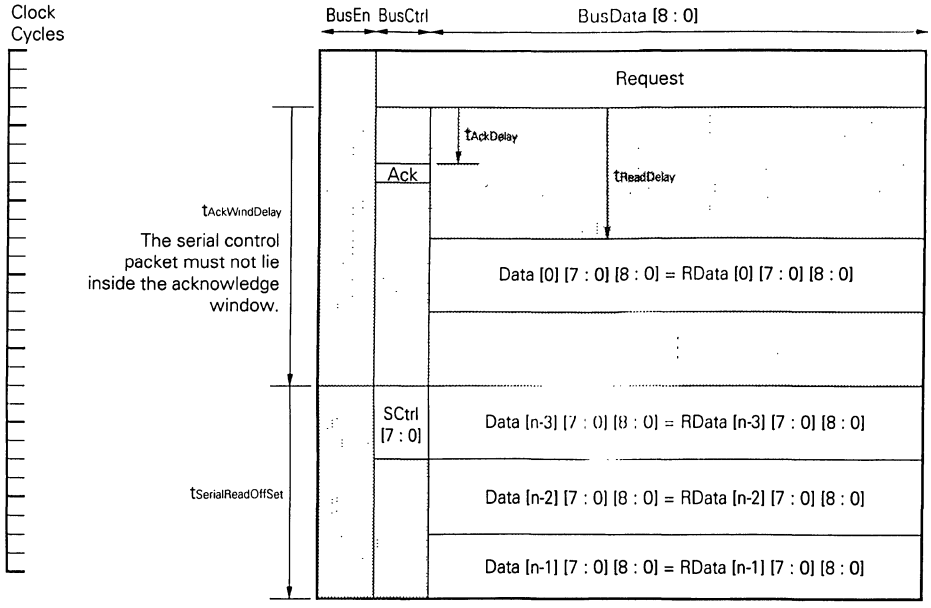
## 7.5 Read Transactions with Serial Control Packet

The following figure shows a memory space read transaction for a command which uses the serial control packet. This packet is used to terminate a transaction before the (CV+1) octbytes of read data have been transferred, where CV is the value of the Count[7:3] Field when interpreted as an unsigned, five bit integer. In the example shown, the read data is terminated after (n) octbytes have been transferred.

The serial control packet is transmitted by the initiating device a time $t_{SerialReadOffSet}$ clock cycles before the end of the last read data octbyte which is transmitted by the responding device.

The serial control packet is also constrained to lie entirely outside the $t_{AckWinDelay}$ interval, as shown in the figure, in order to avoid interference with the acknowledge packet which is being returned by the responding device. Violation of this constraint will produce undefined (Undef) results. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 7-5. Read Transaction with Serial Control Packet**



### 7.6 Write Transactions with Serial Control Packet

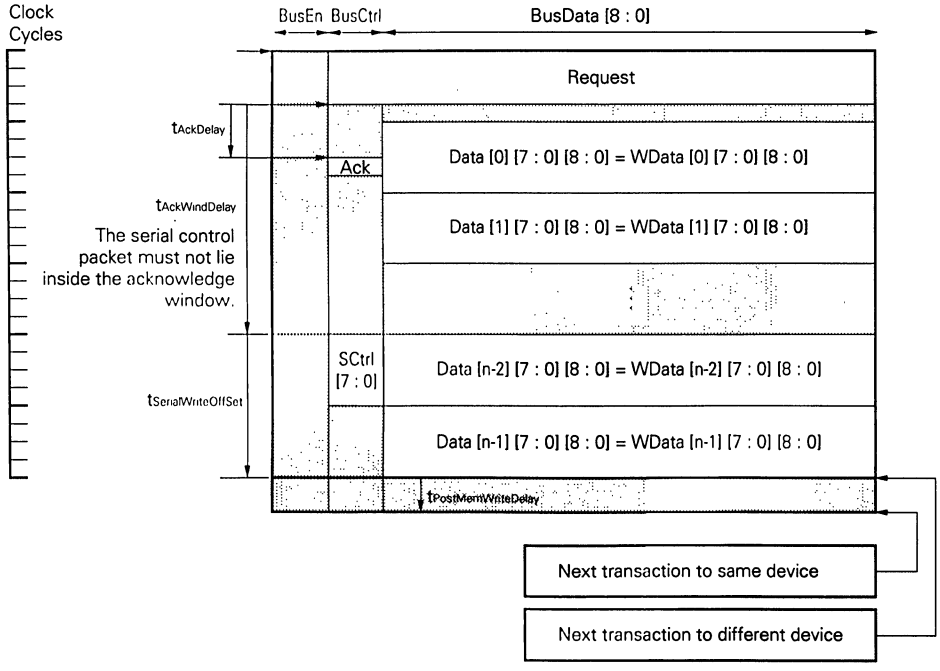
The following figure shows a memory space write transaction for a command which uses the serial control packet. This packet is used to terminate a transaction before the (CV+1) octbytes of write data have been transferred, where CV is the value of the Count[7:3] field when interpreted as an unsigned, five bit integer. In the example shown, the write data is terminated after (n) octbytes have been transferred.

The serial control packet is transmitted by the initiating device a time $t_{SerialWriteOffSet}$ clock cycles before the end of the last write data octbyte which is transmitted by the initiating device.

Note that this offset interval is measured at the initiating device or the responding device; it will be the same at either since the serial address packet and write data packet are moving in the same direction - from initiating device to responding device.

The serial control packet is also constrained to lie entirely outside the $t_{AckWinDelay}$ interval, as shown in the figure, in order to avoid interference with the acknowledge packet which is being returned by the responding device. Violation of this constraint will produce undefined (Undef) results.

**Figure 7-6. Write Transaction with Serial Control Packet**

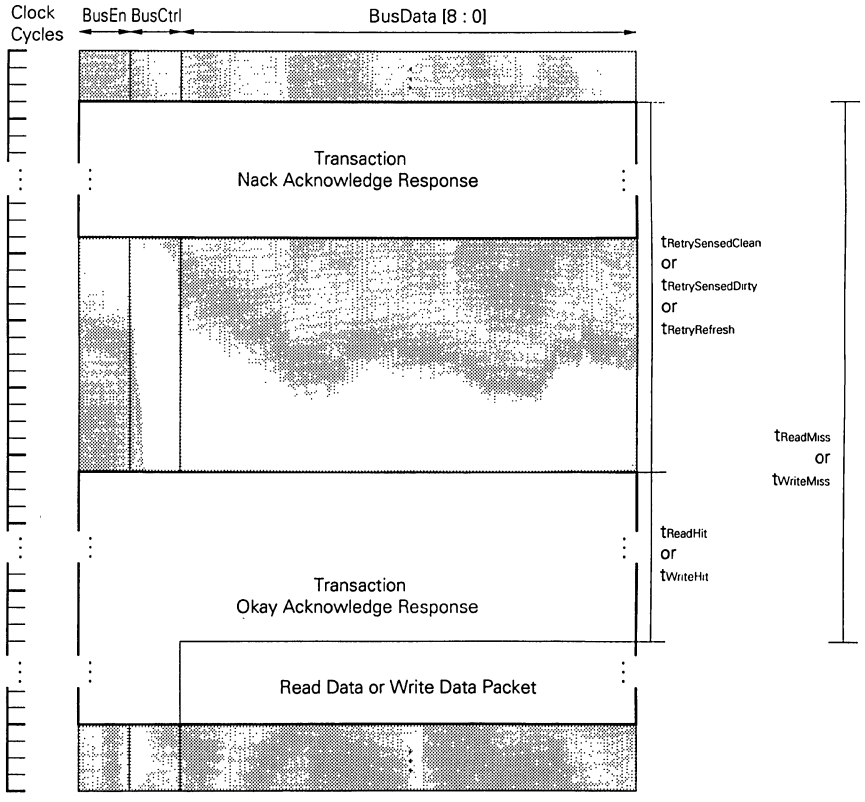


## 8. Nack Acknowledge Response

### 8.1 Retry and Miss Latency

If a responding device returns a Nack acknowledge packet, then no read or write data packet is transacted. The current transaction is complete at the end of the acknowledge window. It will be necessary to wait for an interval of time (called a $t_{RETRY}$ interval) before resubmitting the transaction. The following figure illustrates this case.

**Figure 8-1. Nack Acknowledge Response**




Once the $t_{RETRY}$ interval has elapsed, the transaction may be restarted by the initiating device, and the RDRAM will return an Okay acknowledge packet and the data packet will be transferred. An RDRAM will Nack any other transactions which are issued during the $t_{RETRY}$ interval.

Two miss latency parameters may be derived with the following equations:

$$t_{ReadMiss} = t_{RETRY} + t_{ReadHit} \quad \text{(Eq 8-1)}$$

$$t_{WriteMiss} = t_{RETRY} + t_{WriteHit} \quad \text{(Eq 8-2)}$$

where $t_{RETRY} = \{t_{RetrySensedClean}, t_{RetrySensedDirty}, t_{RetryRefresh}\}$. The $t_{ReadMiss}$ and $t_{WriteMiss}$ parameters are the time from the beginning of the original (Nacked) request packet to the beginning of the data packet which is eventually transferred.

## 8.2 $t_{RETRY}$ Interval

### 8.2.1 Retry Due to RowMiss

If an initiating device requests a region of memory space in an RDRAM slave which is not currently held in the RowSenseAmpCache, the RDRAM will respond with a Nackacknowledge packet. The RDRAM will then begin a RowMiss operation to get the proper row into the RowSenseAmpCache. During the RowMiss, the RDRAM will Nack any request it is given. When the RowMiss is complete, the new row may be accessed.

Each bank has a Valid flag and a Dirty flag for its Row register. After reset, both are zero. After a RowMiss has caused a new row to be placed into the RowSenseAmpCache, the Row register contains its row address and the Valid flag is set to a one. If the RowSenseAmpCache contents are modified with a memory write transaction, the dirty flag will be set. These flags are not directly accessible to initiating devices.

A subsequent RowMiss will cause the old row to be written back to the bank (if it was dirty and an explicit restore was not forced with the Close bit in the request packet) and a new row to be placed into the RowSenseAmpCache. The time required for this is called the $t_{RETRY}$ time, and is added to the normal read and write hit latency times, as shown in the preceding figure. These times are given by the following equations. The component parameters are shown in a subsequent table. All of these $t_{RETRY}$ intervals correspond roughly to the cycle time parameter $t_{RC}$ of a conventional page mode DRAM. This is because RDRAMs use CAS-type accesses for all memory read and write transactions.

After a new row is sensed and placed into the RowSenseAmpCache, a final interval $t_{RowImprestore}$ is used to restore the row in core back to its original state. This is necessary because the DRAM sense operation is destructive. This interval is not in the critical timing path, and is performed in parallel with a subsequent data transfer. It can extend a subsequent retry operation.

There are two $t_{RETRY}$ equations for the 16M RDRAM:

$$t_{RetrySensedClean} = t_{RowOverHead} + t_{RowPrecharge} + t_{RowSense} \quad \text{(Eq 8-3)}$$

$$t_{RetrySensedDirty} = t_{RowOverHead} + t_{RowExprestore} + t_{RowPrecharge} + t_{RowSense} \quad \text{(Eq 8-4)}$$

The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

### 8.2.2 Retry Due to Pending Burst Refresh

In a 16M RDRAM, a refresh burst will also restore the currently accessed row if it is dirty. This requires a $t_{RowExprestore}$ interval. If the row is clean, this interval is not required. A burst of four rows are precharged/sensed/restored (using the $t_{RowImprestore}$ interval), and the current row is precharged/sensed so the RDRAM is left with its RowSenseAmpCache state unaltered (except the row's dirty flag will be cleared):

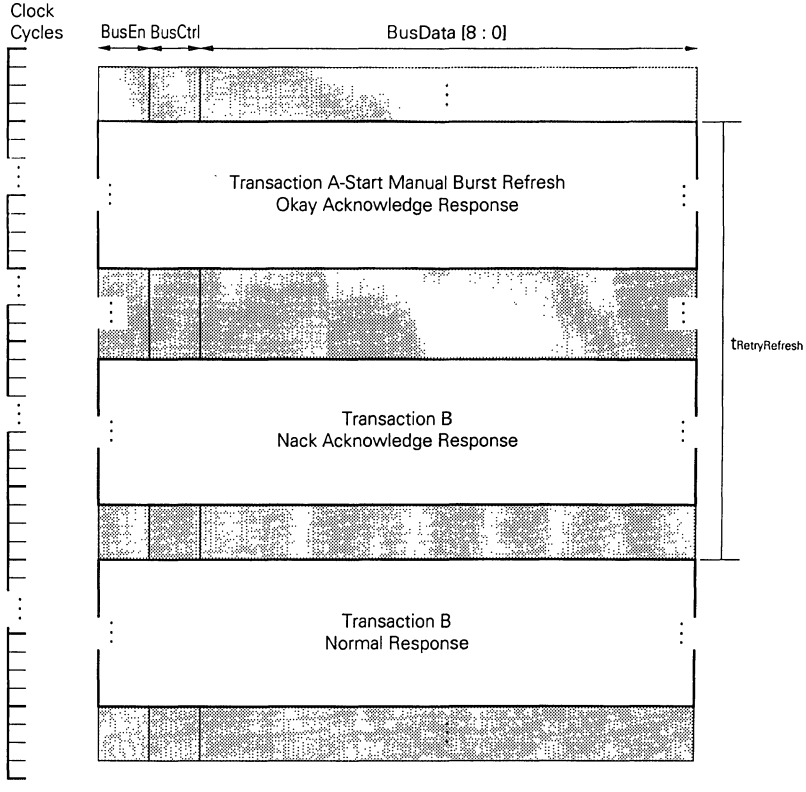
$$t_{RetryRefresh} = (t_{RowOverHead} + t_{RowExprestore}^{\text{Note 1}} + 2t_{RowPrecharge}^{\text{Note 2}} + t_{RowSense}) + 4\,(t_{RowOverHead} + t_{RowImprestore} + 2t_{RowPrecharge}^{\text{Note 2}} + t_{RowSense}) \quad \text{(Eq 8-5)}$$

When a transaction initiates a manual burst refresh in an RDRAM (transaction "A" in the figure below), the RDRAM will Nack all further transactions directed to in during the $t_{RetryRefresh}$ interval after. No information from these Nacked transactions will be retained after the $t_{RetryRefresh}$ interval. After the $t_{RetryRefresh}$ interval, transactions will be handled in a normal fashion. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Notes 1.** This term is not present if the current row is clean.

**2.** This term is present twice in each cycle because the $t_{RowPrecharge}$ interval is also used to ensure that a minimum delay between restore operations is met.

**Figure 8-2. Transaction Holdoff Due to Burst Refresh**



## 8.3 Retry Component Intervals

The $t_{RETRY}$ intervals are built from the $t_{RowOverHead}$, $t_{RowPrecharge}$, $t_{RowSense}$, $t_{RowImprestore}$, and $t_{RowExprestore}$ intervals. All five intervals are measured in $t_{CYCLE}$ units, and thus scale with the clock frequency.

The $t_{RowOverHead}$ interval consists of the RowMiss state machine overhead. The remaining four intervals represent the width of intervals used for timing core operations. These core operations have minimum times measured in nanosecond units (this is shown in the "core timing(ns)" columns in the table below). The four intervals are composed of a fixed part and a variable (programmable) part. If the clock frequency is reduced, the variable part may be reduced so the sum of the fixed and variable parts remain greater than or equal to the minimum core operation time (in nanoseconds).

**Table 8-1. Retry Components**

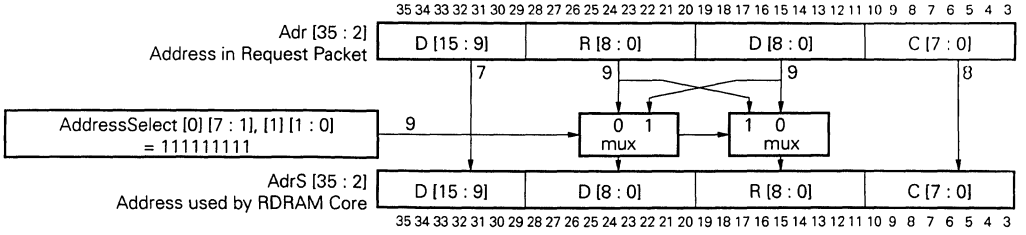
| Delay Parameter | Fixed Part (overhead) and Variable Part[Note] | 16M RDRAM | |
|---|---|---|---|
| | | $t_{CYCLE}$ Units (4 ns) | core timing (ns) with $t_{CYCLE}$ = 4ns |
| $t_{RowOverHead}$ | Row overhead | 6 | 24 |
| | – | n/a | |
| $t_{RowPrecharge}$ | RowPrecharge overhead | 4 | 20 |
| | RowPrecharge[4:0] | 1 | |
| $t_{RowSense}$ | RowSense overhead | 4 | 44 |
| | RowSense[4:0] | 7 | |
| $t_{RowImprestore}$ | RowImpRestore overhead | 4 | 56 |
| | RowImpRestore[4:0] | 10 | |
| $t_{RowExprestore}$ | RowExpRestore overhead | 4 | 32 |
| | RowExpRestore[4:0] | 4 | |

**Note** The variable part is programmed into the indicated field of the RasInterval register.

## 9. AddressMapping

The address space decoding logic contained in a 16M RDRAM is shown in the following figure. The initiating device places a 33 bit physical octbyte address Adr[35:3] on the Channel. This address is received by the RDRAM slave. The AddressSelect[1][1:0], [0][7:1] control register allows individual bits of the Adr[28:20] and Adr [19:11] fields to be swapped to produce the AdrS[28:20] and AdrS[19:11] fields. The Adr[35:29] and Adr[10:3] fields pass through unaltered to the AdrS[35:29] and AdrS[10:3] fields. The figure shows the case when AddressSelect[0][7:1],[1][1:0] = 111111111, and the two nine bit address fields are exchanged. The detailed functional description is provided in **"Rambus DRAM user's manual (Reference Manual)"**.

**Figure 9-1. AddressMapping Hardware**



## 10. Electrical Characteristics (Preliminary)

### Absolute Maximum Ratings

| Symbol | Parameter | MIN. | MAX. | Unit | Note |
|---|---|---|---|---|---|
| $V_{I,ABS}$ | Voltage applied to any RSL pin with respect to GND | −0.5 | $V_{DD}$+0.5 | V | |
| $V_{I,TTL,ABS}$ | Voltage applied to any TTL pin with respect to GND | −0.5 | $V_{DD}$+0.5 | V | |
| $V_{DD,ABS}$ | Voltage on $V_{DD}$ with respect to GND | −0.5 | $V_{DD,MAX}$+1.0 | V | |
| $T_{OPT}$ | Operation temperature | 0 | +70 | °C | 1 |
| $T_{STORE}$ | Storage temperature | −55 | +125 | °C | |

**Caution The following table represents stress ratings only, and functional operation at the maximums is not guaranteed. Extended exposure to the maximum ratings may affect device reliability. Furthermore, although devices contain protective circuitry to resist damage from static electric discharge, always take precautions to avoid high static voltages or electric fields.**

**Note 1** This parameter apply at the status of using 50% Rambus channel by Read or Write and a transverse air flow greater than 1.5m/s maintained.

### Thermal Parameters

| Symbol | Parameter | MIN. | MAX. | Unit |
|---|---|---|---|---|
| $T_J$ | Junction operating temperature | | 100 | °C |
| $\Theta_{JC}$ | Junction-to-Case thermal resistance | | 5 | °C/W |

### Capacitance

| Symbol | Parameter | MIN. | MAX. | Unit |
|---|---|---|---|---|
| $C_{I/O}$ | Low-swing input/output parasitic capacitance | | 2 | pF |
| $C_{I,TTL}$ | TTL input parasitic capacitance | | 8 | pF |

### Power Consumption

| Mode | Parameter | | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| $I_{CC1}$ | Active Current | -A45 | | 110 | mA |
| | | -A50 | | 125 | |
| $I_{CC2}$ | Standby Current | -A45 | | 330 | mA |
| | | -A50 | | 350 | |
| $I_{CC3}$ | Read Operation Current | -A45 | | 440 | mA |
| | (Burst Length = 256) | -A50 | | 480 | |
| $I_{CC4}$ | Write Operation Current | -A45 | | 435 | mA |
| | | -A50 | | 460 | |
| $I_{CC5}$ | Power Down Current | | | T.B.D | mA |

**Caution These do not include the $I_{OL}$ current passing through the low-swing pins to ground.**

### Recommended Operating Conditions

| Symbol | Parameter | MIN. | MAX. | Unit |
|---|---|---|---|---|
| $V_{DD}$, $V_{DDA}$ | Supply voltage | 3.15 | 3.45 | V |
| $V_{REF}$ | Reference voltage | 1.95 | 2.15 | V |
| $V_{swing}$ | Input voltage range | 1.0 | 1.4 | V |
| $V_{IL}$ | Input low voltage | $V_{REF}$-0.7 | $V_{REF}$-0.5 | V |
| $V_{IH}$ | Input high voltage | $V_{REF}$+0.5 | $V_{REF}$+0.7 | V |
| $V_{IL, TTL}$ | TTL input low voltage | -0.5 | +0.8 | V |
| $V_{IH, TTL}$ | TTL input high voltage | 2.0 | $V_{DD}$+0.5 | V |

### DC Characteristics (Recommended operating conditions unless otherwise noted)

| Symbol | Parameter | Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| $I_{REF}$ | $V_{REF}$ current | $V_{REF}$=Maximum | -10 | +10 | μA |
| $I_{OH}$ | High level output current | $0 \le V_{OUT} \le V_{DD}$ | -10 | +10 | μA |
| $I_{OL}$ | Low level output current | $V_{OUT}$=1.6 V | | 25 | mA |
| $I_{I, TTL}$ | TTL input leakage current | $0 \le V_{I, TTL} \le V_{DD}$ | -10 | +10 | μA |
| $V_{OH, TTL}$ | High level TTL output voltage | $I_{OH, TTL}$=-0.25 mA | 2.4 | $V_{DD}$ | V |
| $V_{OL, TTL}$ | Low level TTL output voltage | $I_{OL, TTL}$=1.0 mA | 0 | 0.4 | V |

### Recommended Timing Conditions

| Symbol | Parameter | | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| $t_{PAUSE}$ | Pause time after Power On | | | 200 | μs |
| $t_{CR}$, $t_{CF}$ | TxClk and RxClk input rise and fall times | | 0.3 | 0.7 | ns |
| $t_{CYCLE}$ | TxClk and RxClk cycle times | -A45 | 4.45 | 6 | ns |
| | | -A50 | 4 | 6 | ns |
| $t_{TICK}$ | Transport time per bit per pin (this timing interval is synthesized by the RDRAM's internal clock generator) | | $t_{CYCLE}/2$ | $t_{CYCLE}/2$ | ns |
| $t_{CH}$, $t_{CL}$ | TxClk and RxClk high and low times | | 47% | 53% | $t_{CYCLE}$ |
| $t_{TR}$ | TxClk-RxClk differential | | 0.25 | 0.7 | ns |
| $t_{SD}$ | SIn-to-SOut propagation delay | | | 50 | ns |
| $t_Q$ | TxClk-to-Data/Control output time | | $t_{CYCLE}/8+0.05$ | $t_{CYCLE}3/8-0.05$ | ns |
| $t_S$ | Data/Control-to-RxClk setup time | | $t_{CYCLE}/4-0.05$ | | ns |
| $t_H$ | RxClk-to-Data/Control hold time | | $t_{CYCLE}/4-0.05$ | | ns |
| $t_{REF}$ | Refresh interval | | | 32 | ms |
| $t_{LOCK}$ | RDRAM internal clock generator lock time | | 500 | | $t_{CYCLE}$ |

### Transaction Timing Characteristics

| Symbol | Parameter | MIN. | Unit |
|---|---|---|---|
| $t_{PostRegWriteDelay}$ | Delay from the end of the current transaction to the beginning of the next transaction if the current transaction is a write to register space and the next transaction is made to the same device. Use zero delay if the next transaction is to a different device. | 6 | $t_{CYCLE}$ |
| $t_{PostMemWriteDelay}$ | Delay from the end of the current transaction to the beginning of the next transaction if the current transaction is a write to memory space and the next transaction is made to the same device. Use zero delay if the next transaction is to a different device. | 4 | $t_{CYCLE}$ |
| $t_{PostMemReadDelay}$ | Delay from the end of the current memory read transaction to the beginning of the next transaction. | 2 | $t_{CYCLE}$ |
| $t_{SerialReadOffSet}$ | Delay from the beginning of a serial address subpacket or serial control packet to the beginning of the read data subpacket (octbyte) with which it is associated. | 12 | $t_{CYCLE}$ |
| $t_{SerialWriteOffSet}$ | Delay from the beginning of a serial address subpacket or serial control packet to the beginning of the write data subpacket (octbyte) with which it is associated. | 8 | $t_{CYCLE}$ |

### Data and Transaction Latency Characteristics

| Symbol | Parameter | MIN. | Unit | Notes |
|---|---|---|---|---|
| $t_{ReadDelay}$ | Delay from the end of a read request packet to the beginning of the read data packet. | 7 | $t_{CYCLE}$ | 1 |
| $t_{WriteDelay}$ | Delay from the end of a write request packet to the beginning of the write data packet. | 1 | $t_{CYCLE}$ | 2 |

**Notes 1.** $t_{ReadDelay}$ is programmed to its minimum value.

**2.** $t_{WriteDelay}$ is programmed to its minimum value.

### Hit, Retry and Miss Delay Characteristics

| Symbol | Parameter | | MIN. | Unit | Notes |
|---|---|---|---|---|---|
| $t_{ReadHit}$ | Start of request packet to start of read data packet for row hit (Okay). | | 10 | $t_{CYCLE}$ | 1 |
| $t_{WriteHit}$ | Start of request packet to start of write data packet for row hit (Okay). | | 4 | $t_{CYCLE}$ | 1 |
| $t_{RetrySensedClean}$ | Start of request packet for row miss (Nack) to start of request packet for row hit (Okay). The previous row is unmodified. | | 22 | $t_{CYCLE}$ | 2 |
| $t_{RetrySensedDirty}$ | Start of request packet for row miss (Nack) to start of request packet for row hit (Okay). The previous row is modified. | | 30 | $t_{CYCLE}$ | 2 |
| $t_{RetryRefresh}$ | Start of request packet for row miss (Nack) to start of request packet for row hit (Okay). | Clean | 191 | $t_{CYCLE}$ | 2 |
| | | Dirty | 199 | | |
| $t_{ReadMiss}$ | Start of request packet for row miss (Nack) to start of Read Data packet for row hit (Okay). | | 32 | $t_{CYCLE}$ | 3 |
| $t_{WriteMiss}$ | Start of request packet for row miss (Nack) to start of Write Data packet for row hit (Okay). | | 26 | $t_{CYCLE}$ | 3 |

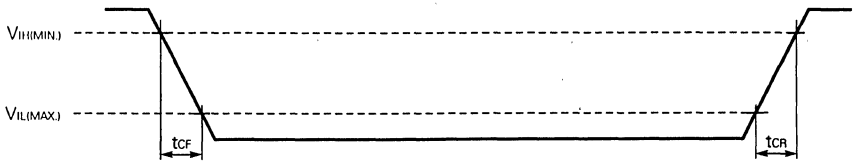
**Notes 1.** Programmable

**2.** $t_{RowExprestore}$, $t_{Precharge}$, and $t_{Sense}$ are programmed to there minimum value.

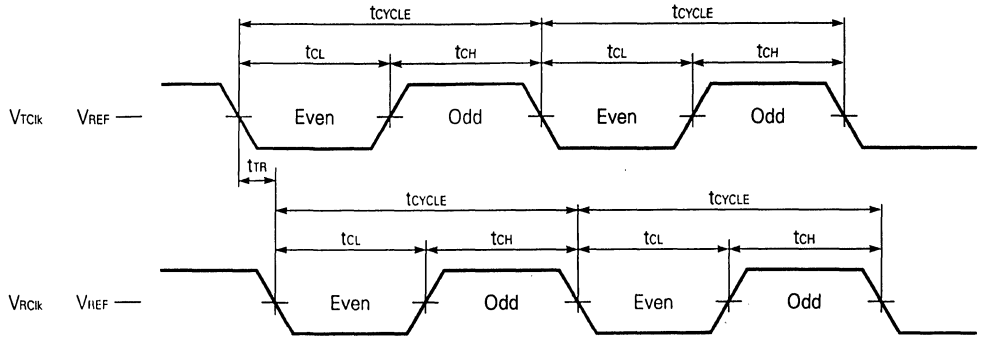
**3.** Calculated with $t_{RetrySensedClean(MIN)}$.

## Rise/Fall Timing Chart

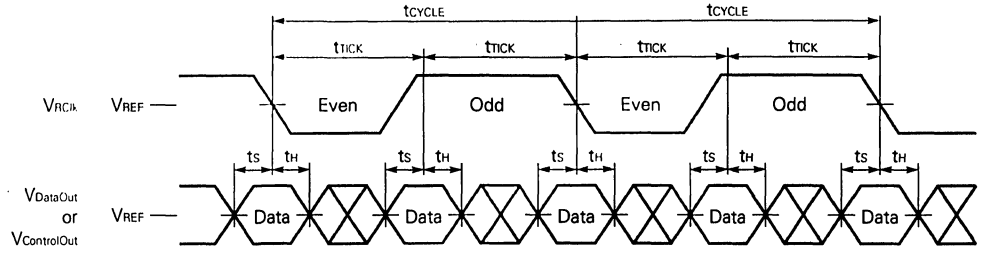
$V_{RxClk}$, $V_{TxClk}$



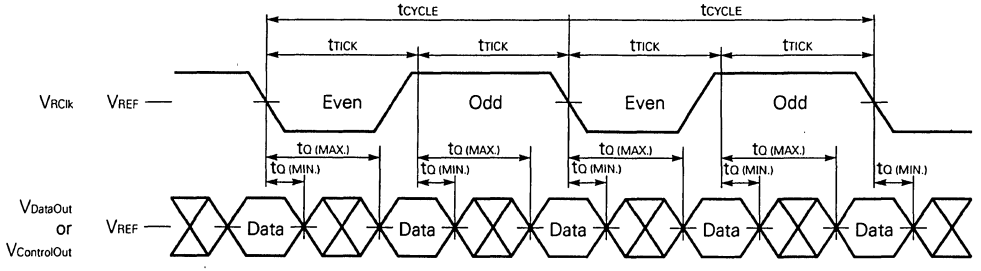

## Clock Timing Chart



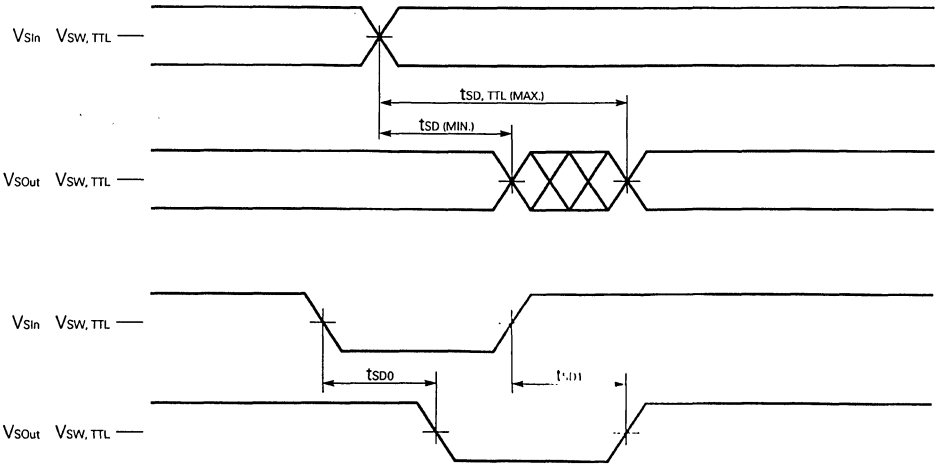

## Receive Data Timing Chart



### Transmit Data Timing Chart



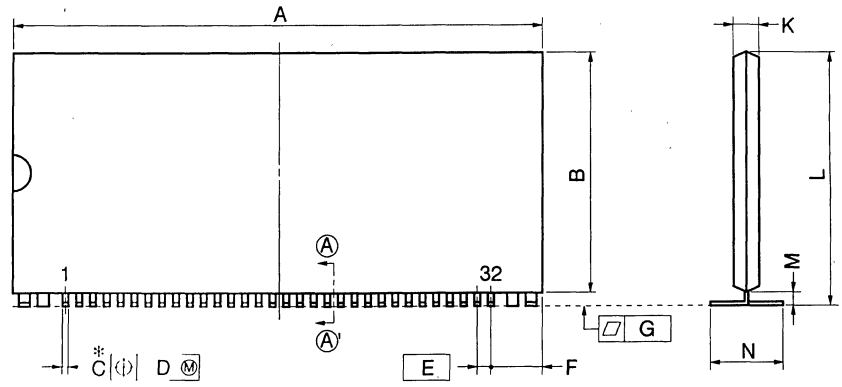

### Serial Configuration Pin Timing Chart

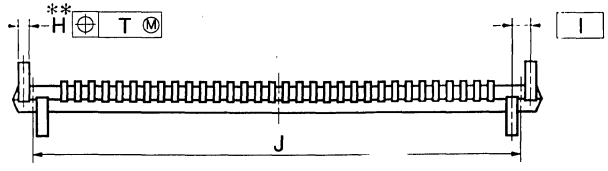



**Remark** $V_{SW,\ TTL}$ = 1.5 V

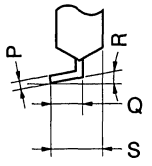
## 11. Package Drawings

## 32 PIN PLASTIC SVP (11×25)




Detail of Ⓐ - Ⓐ' part




**NOTE**

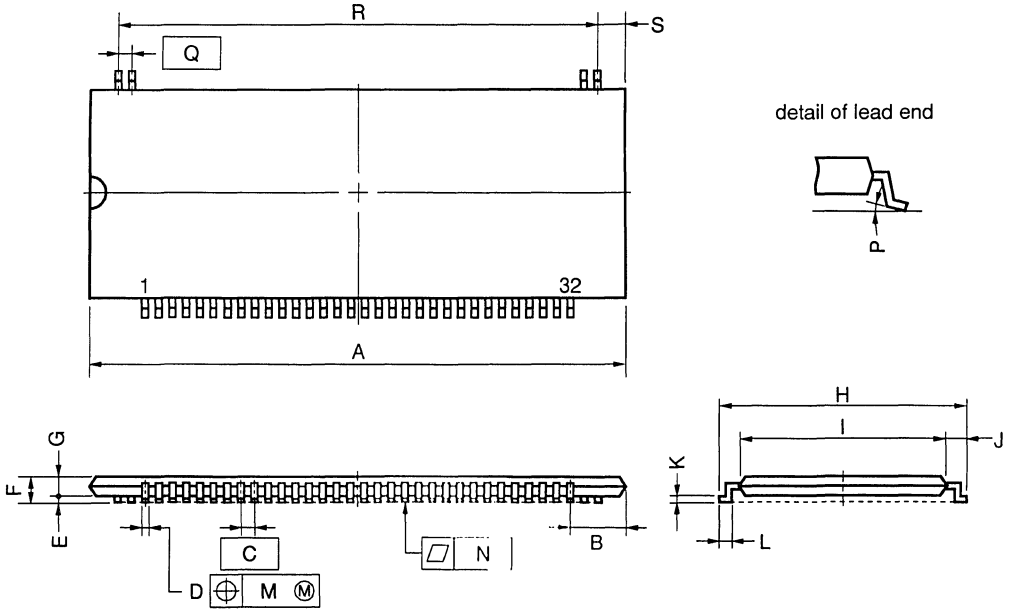
* Each I/O lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

** Each support lead centerline is located within 0.18 mm (0.007 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 25.30 MAX. | 0.996 MAX. |
| B | 11.0±0.1 | 0.433±0.004 |
| C | 0.24±0.06 | $0.009^{+0.003}_{-0.002}$ |
| D | 0.13 | 0.005 |
| E | 0.65 (T.P.) | 0.026 (T.P.) |
| F | 2.575 MAX. | 0.102 MAX. |
| G | 0.10 | 0.004 |
| H | 0.52±0.06 | 0.020±0.002 |
| I | 0.9 (T.P.) | 0.035 (T.P.) |
| J | 23.20 | 0.913 |
| K | 1.25 | 0.049 |
| L | 11.80 MAX. | 0.465 MAX. |
| M | 0.5±0.1 | $0.020^{+0.004}_{-0.005}$ |
| N | 3.70 MAX. | 0.146 MAX. |
| P | $0.17^{+0.025}_{-0.015}$ | 0.007±0.001 |
| Q | 0.9±0.25 | $0.035^{+0.011}_{-0.010}$ |
| R | $3°^{+7°}_{-3°}$ | $3°^{+7°}_{-3°}$ |
| S | 1.90 MAX. | 0.075 MAX. |
| T | 0.18 | 0.007 |

S32VN-65-9

## 72/36 PIN PLASTIC SSOP TYPE




**NOTE**

Each lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 25.30 MAX. | 0.996 MAX. |
| B | 2.575 MAX. | 0.102 MAX. |
| C | 0.65 (T.P.) | 0.026 (T.P.) |
| D | $0.24\pm0.06$ | $0.009^{+0.003}_{-0.002}$ |
| E | $0.25\pm0.05$ | $0.010^{+0.002}_{-0.003}$ |
| F | 1.6 MAX. | 0.063 MAX. |
| G | 1.25 | 0.049 |
| H | $13.0\pm0.2$ | $0.512\pm0.008$ |
| I | $11.0\pm0.1$ | $0.433\pm0.004$ |
| J | $1.0\pm0.2$ | $0.039^{+0.009}_{-0.008}$ |
| K | $0.17^{+0.025}_{-0.015}$ | $0.007\pm0.001$ |
| L | $0.5\pm0.1$ | $0.020^{+0.004}_{-0.005}$ |
| M | 0.13 | 0.005 |
| N | 0.10 | 0.004 |
| P | $3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$ | $3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$ |
| Q | $0.65\pm0.1$ | $0.026\pm0.004$ |
| R | 22.75 | 0.896 |
| S | 1.275 MAX. | 0.051 MAX. |

P32G6-65A

[MEMO]

# Synchronous GRAM

NEC

MOS INTEGRATED CIRCUIT

# μPD481850

## 8M-bit Synchronous GRAM

### Description

The μPD481850 is a synchronous graphics memory (SGRAM) organized as 128 K words × 32 bits × 2 banks random access port.

This device can operate up to 100 MHz by using synchronous interface. Also, it has 8-column Block Write function to improve capability in graphics system.

This product is packaged in 100-pin plastic QFP (14 × 20 mm).

### Features

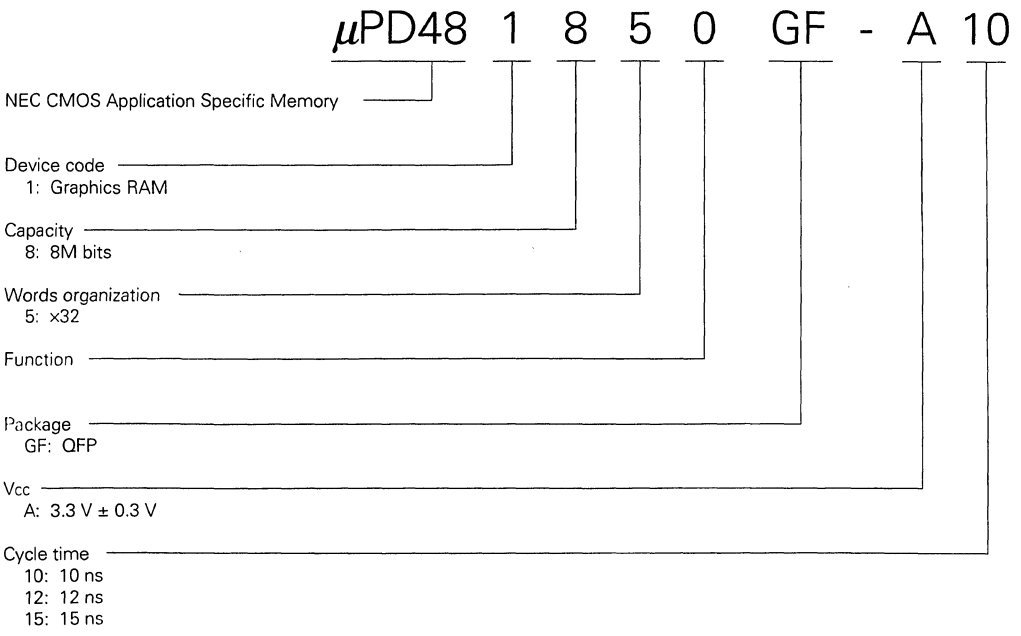
- 131,072 words × 32 bits × 2 banks memory
- Synchronous interface (Fully synchronous DRAM with all input signals are latched at rising edge of clock)
  - : Pulsed interface
  - : Automatic precharge and controlled precharge commands
  - : Ping-pong operation between the two internal memory banks
  - : Up to 100 MHz operation frequency
- Possible to assert random column address in every cycle
- Dual internal banks controlled by A9 (Bank Address: BA)
- Byte control using DQM0 to DQM3 signals both in read and write cycle
- 8-column Block Write (BW) function
- Persistent write per bit (WPB) function
- Programmable wrap sequence (Sequential/Interleave)
- Programmable burst length (1, 2, 4, 8 and full page)
- Programmable $\overline{CAS}$ latency (1, 2, and 3)
- Power Down operation and Clock Suspend operation
- Auto refresh (CBR refresh) or self refresh capability
- Single 3.3 V ± 0.3 V power supply
- LVTTL compatible inputs and outputs
- 100-pin Plastic QFP (14 × 20 mm)
- 1,024 refresh cycles/16 ms
- Burst termination by Precharge command
- Burst termination by Burst stop command (in case of full-page burst)

### Ordering Information

| Part number | Cycle time ns (MIN.) | Clock frequency MHz (MAX.) | Package |
|---|---|---|---|
| μPD481850GF-A10-JBT | 10 | 100 | 100-pin Plastic QFP (14 × 20 mm) |
| μPD481850GF-A12-JBT | 12 | 83 | |
| μPD481850GF-A15-JBT | 15 | 66 | |

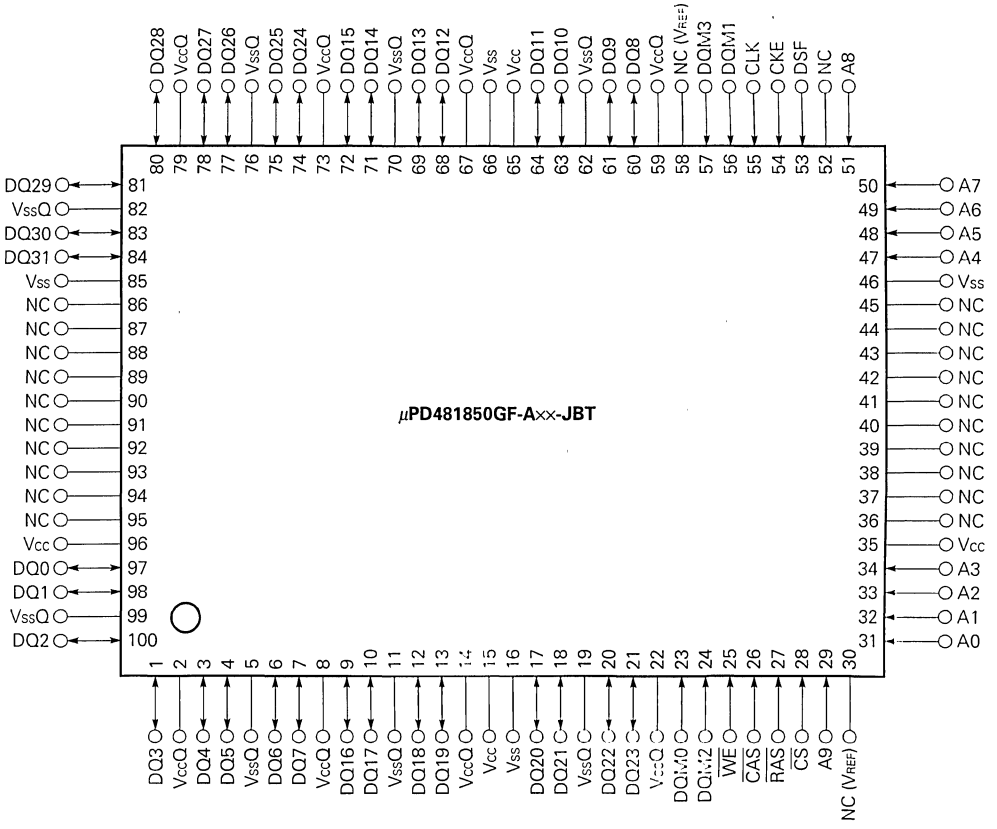
## Part Number

### Synchronous GRAM



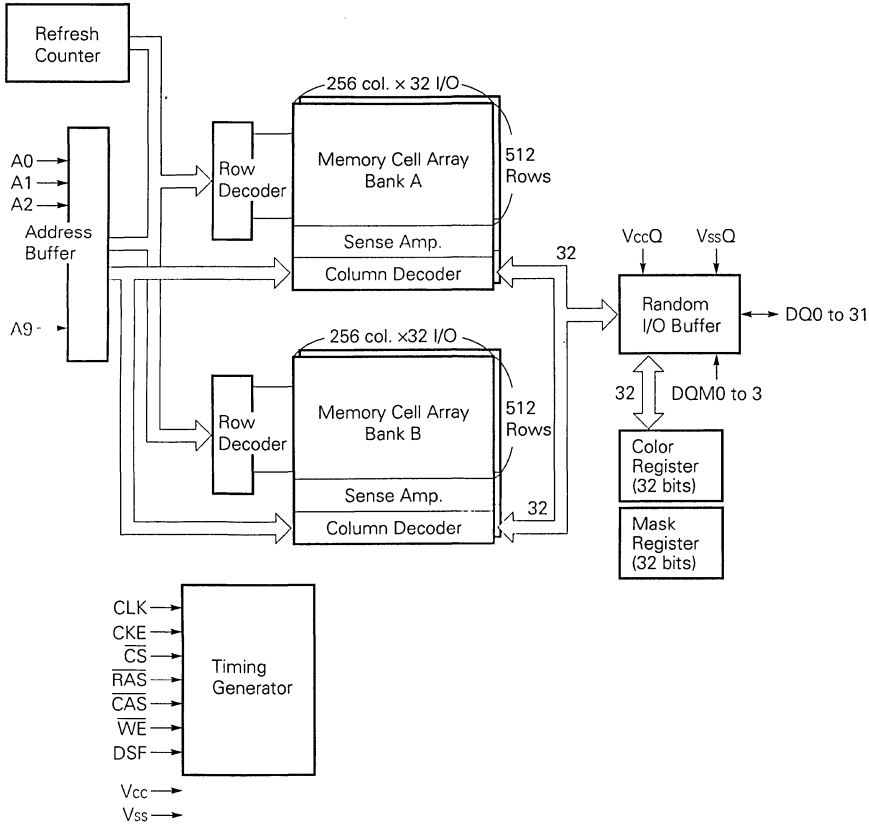
## Pin Configuration (Marking Side)

**100-pin Plastic QFP (14 × 20 mm)**




| | | |
|---|---|---|
| A0 - A9 | : | Address inputs |
| A0 - A8 | : | Row address inputs |
| A0 - A7 | : | Column address inputs |
| A9 | : | Bank address |
| DQ0 - DQ31 | : | Data inputs/outputs |
| $\overline{CS}$ | : | Chip select |
| $\overline{RAS}$ | : | Row address strobe |
| $\overline{CAS}$ | : | Column address strobe |
| $\overline{WE}$ | : | Write enable |
| DQM0 - DQM3 | : | DQ mask enable |
| DSF | : | Special function enable |
| CKE | : | Clock enable |
| CLK | : | System clock input |
| $V_{CC}$ | : | Supply voltage |
| $V_{SS}$ | : | Ground |
| $V_{CC}Q$ | : | Supply voltage for DQ |
| $V_{SS}Q$ | : | Ground for DQ |
| NC | : | No connection |

## Block Diagram



CONTENTS


1. **Input/Output Pin Function** ..... **365**

2. **Commands** ..... **366**

3. **Simplified State Diagram** ..... **370**

4. **Truth Table** ..... **371**
   - 4.1 Command Truth Table ..... 371
   - 4.2 DQM Truth Table ..... 371
   - 4.3 CKE Truth Table ..... 372
   - 4.4 Operative Command Table ..... 373
   - 4.5 Command Truth Table for CKE ..... 380
   - 4.6 Command Truth Table for Two Banks Operation ..... 381

5. **Initialization** ..... 382

6. **Programming the Mode Register** ..... 383

7. **Mode Register** ..... **384**
   - 7.1 Burst Length and Sequence ..... 385

8. **Programming the Special Register** ..... **386**

9. **Address Bits of Bank-Select and Precharge** ..... **387**

10. **Precharge** ..... **388**

11. **Auto Precharge** ..... **389**
    - 11.1 Read with Auto Precharge ..... 389
    - 11.2 Write with Auto Precharge ..... 390
    - 11.3 Block Write with Auto Precharge ..... 391

12. **Write/Block Write with Write Per Bit** ..... **392**

13. **Block Write** ..... **392**

14. **Read/Write Command Interval** ..... **394**
    - 14.1 Read to Read Command Interval ..... 394
    - 14.2 Write to Write Command Interval ..... 394
    - 14.3 Write to Read Command Interval ..... 395
    - 14.4 Block Write to Write or Write/Block Write Command Interval ..... 396
    - 14.5 Block Write to Read Command Interval ..... 396
    - 14.6 Read to Write/Block Write Command Interval ..... 397

**15. Burst Termination ..... 399**
15.1 Burst Stop Command in Full Page ..... 399
15.2 Precharge Termination ..... 400
15.2.1 Precharge Termination in READ Cycle ..... 400
15.2.2 Precharge Termination in WRITE Cycle ..... 401

**16. Electrical Specifications (Preliminary) ..... 403**
16.1 AC Parameters for Read/Write Cycles ..... 408
16.2 Relationship between Frequency and Latency ..... 410
16.3 $\overline{CS}$ Function ..... 411
16.4 Basic Cycles ..... 412
16.4.1 Initialization ..... 412
16.4.2 Mode Register Set ..... 413
16.4.3 Refresh Cycle ..... 414
16.4.4 Cycle with Auto Precharge ..... 416
16.4.5 Full Page Mode Cycle ..... 422
16.4.6 Precharge Termination Cycle ..... 428
16.4.7 Clock Suspension ..... 431
16.4.8 Power Down Mode ..... 437
16.4.9 Other Cycles ..... 438
16.5 Graphics Cycles ..... 440
16.6 Application Cycles ..... 447
16.6.1 Page Cycles with Same Bank ..... 447
16.6.2 Cycles with Pingpong Banks ..... 453
16.6.3 READ and WRITE Cycles ..... 460
16.6.4 Interleaved Cycles ..... 463
16.6.5 Full Page Random Cycles ..... 469

**17. Package Drawing ..... 471**

**18. Recommended Soldering Conditions ..... 472**

## 1. Input/Output Pin Function

| Pin name | Input/Output | Function |
|---|---|---|
| CLK | Input | CLK is the master clock input. Other inputs signals are referenced to the CLK rising edge. |
| CKE | Input | CKE determine validity of the next CLK (clock). If CKE is high, the next CLK rising edge is valid; otherwise it is invalid. If the CLK rising edge is invalid, the internal clock is not asserted and the μPD481850 suspends operation.<br>When the μPD481850 is not in burst mode and CKE is negated, the device enters power down mode. During power down mode, CKE must remain low.<br>In Self refresh mode, low level on this pin is also used as part of the input command to specify Self refresh. |
| $\overline{CS}$ | Input | $\overline{CS}$ low starts the command input cycle. When $\overline{CS}$ is high, commands are ignored but operations continue. |
| $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ | Input | $\overline{RAS}$, $\overline{CAS}$ and $\overline{WE}$ have the same symbols on conventional DRAM but different functions. For details, refer to the command table. |
| DSF | Input | DSF is part of the inputs of graphics command of the μPD481850.<br>If DSF is inactive (Low level), μPD481850 operates as same as SDRAM. |
| A0 - A8 | Input | Row Address is determined by A0 - A8 at the CLK (clock) rising edge in the activate command cycle.<br>Column Address is determined by A0 - A7 at the CLK rising edge in the read or write command cycle.<br>A8 defines the precharge mode. When A8 is high in the precharge command cycle, both banks are precharged; when A8 is low, only the bank selected by A9 is precharged.<br>When A8 high in read or write command cycle, the precharge start automatically after the burst access. |
| A9 | | A9 is the bank select signal (BA). In command cycle, A9 low selects bank A and A9 high selects bank B. |
| DQM0 - DQM3 | Input | DQM controls I/O buffers. DQM0 corresponds to the lowest byte (DQ0 to DQ7), DQM1 corresponds to DQ8 to DQ15, DQM2 corresponds to DQ16 to DQ23. DQM3 corresponds to DQ24 to DQ31.<br>In read mode, DQM controls the output buffers like a conventional $\overline{OE}$ pin. DQM high and DQM low turn the output buffers off and on, respectively. The DQM latency for the read is two clocks.<br>In write mode, DQM controls the word mask. Input data is written to the memory cell if DQM is low but not if DQM is high. The DQM latency for the write is zero. |
| DQ0 - DQ31 | Input/Output | DQ pins have the same function as I/O pins on a conventional DRAM. These are normally 32-bit data bus and are used for inputting and outputting data.<br>· Function as the mask data input pins in the special register set command. Write operations can be performed after Active command with WPB (old mask data).<br>· Functions as the column selection data input pin in the block write cycle. |
| $V_{CC}$<br>$V_{SS}$<br>$V_{CC}Q$<br>$V_{SS}Q$ | (Power supply) | $V_{CC}$ and $V_{SS}$ are power supply pins for internal circuits. $V_{CC}Q$ and $V_{SS}Q$ are power supply pins for the output buffers. |

## 2. Commands

### Mode register set command

($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, DSF = Low)

The μPD481850 has a mode register that defines how the device operates. In this command, A0 through A9 are the data input pins. After power on, the mode register set command must be executed to initialize the device.

The mode register can be set only when both banks are in idle state.

During 20 ns ($t_{RSC}$) following this command, the μPD481850 cannot accept any other commands.

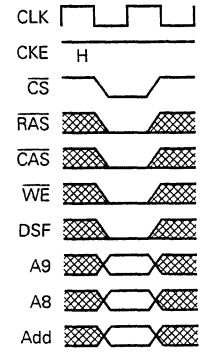



**Fig. 1 Mode register set command**

### Bank activate command

($\overline{CS}$, $\overline{RAS}$, DSF = Low, $\overline{CAS}$, $\overline{WE}$ = High)

The μPD481850 has two banks, each with 512 rows.

This command activates the bank selected by A9 (BA) and a row address selected by A0 through A8.

This command corresponds to a conventional DRAM's $\overline{RAS}$ falling.

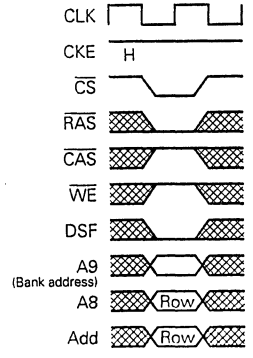



**Fig. 2 Row address strobe and bank active command**

### Bank activate command with WPB enable

($\overline{CS}$, $\overline{RAS}$ = Low, $\overline{CAS}$, $\overline{WE}$, DSF = High)

This command is same as Bank activate command. After this command, write per bit function is available. Mask register's data is used as write mask data.

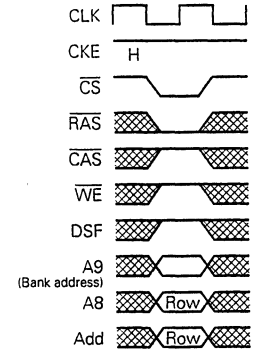



**Fig. 3 Row address strobe and bank active command with WPB enable**

**Precharge command**

($\overline{CS}$, $\overline{RAS}$, $\overline{WE}$, DSF = Low, $\overline{CAS}$ = High)

This command begins precharge operation of the bank selected by A9 (BA) and A8. When A8 is High, both banks are precharged, regardless of A9. When A8 is Low, only the bank selected by A9 is precharged. A9 low selects bank A and A9 high selects bank B.

After this command, the μPD481850 can't accept the activate command to the precharging bank during $t_{RP}$ (precharge to activate command period). This command can terminate the current burst operation (2, 4, 8, full page burst length).

This command corresponds to a conventional DRAM's $\overline{RAS}$ rising.

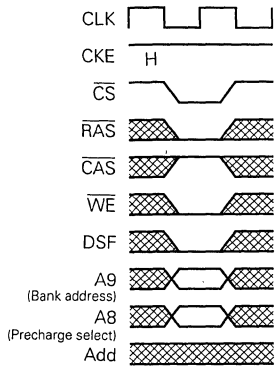



**Fig. 4 Precharge command**

**Write command**

($\overline{CS}$, $\overline{CAS}$, $\overline{WE}$, DSF = Low, $\overline{RAS}$ = High)

If the mode register is in the burst write mode, this command sets the burst start address given by the column address to begin the burst write operation. The first write data must be input with this write operation. The first write data in burst mode can input with this command with subsequent data on following clocks.

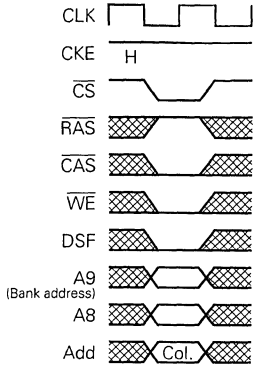



**Fig. 5 Column address and write command**

**Read command**

($\overline{CS}$, $\overline{CAS}$, DSF = Low, $\overline{RAS}$, $\overline{WE}$ = High)

This command sets the burst start address given by the column address.

Read data is available after $\overline{CAS}$ latency requirements have been met.

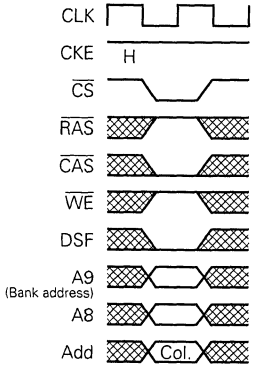



**Fig. 6 Column address and read command**

**CBR (auto) refresh command**

($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, DSF = Low, $\overline{WE}$, CKE = High)

This command is a request to begin the CBR refresh operation. The refresh address is generated internally.

Before executing CBR refresh, both banks must be precharged.

After this cycle, both banks will be in the idle (precharged) state and ready for a bank activate command.

During $t_{RC}$ period (from refresh command to refresh or activate command), the μPD481850 cannot accept any other command.

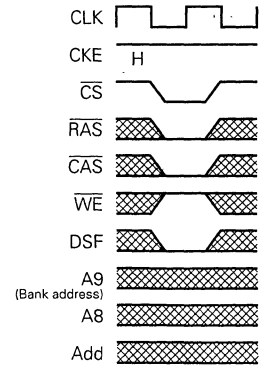



**Fig. 7 Auto refresh command**

**Self refresh entry command**

($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, DSF, CKE = Low, $\overline{WE}$ = High)

After the command execution, self refresh operation continues while CKE remains low. When CKE goes high, the μPD481850 exits the self refresh mode.

During self refresh mode, refresh interval and refresh operation are performed internally, so there is no need for external control.

Before executing self refresh, both banks must be precharged.

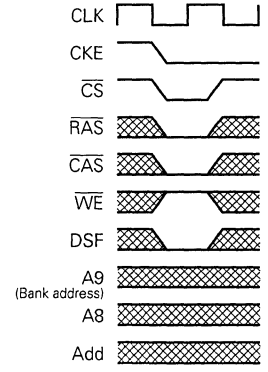



**Fig. 8 Self refresh entry command**

**Burst stop command in full page**

($\overline{CS}$, $\overline{WE}$, DSF = Low, $\overline{RAS}$, $\overline{CAS}$ = High)

This command can stop the current full page burst (BL = 256) operation. If BL is set to 2, 4, 8, to execute this command is Nop.

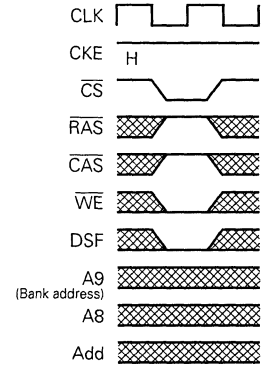



**Fig. 9 Burst stop command in Full Page mode**

**No operation**

($\overline{CS}$, DSF = Low, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ = High)

This command is not a execution command. No operations begin or terminate by this command.

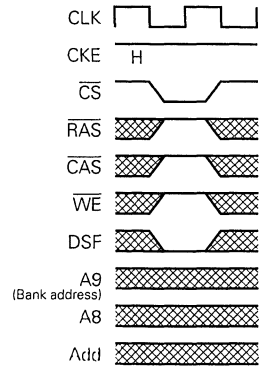



**Fig. 10 No operation**

**Special register set command**

($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ = Low, DSF = High)

The μPD481850 has two special registers for graphics commands. One is color register and the other is mask register. In this command, A0 through A9 are the data input pins for the register select (color or mask register). DQ0 through DQ31 are the data input pins for the Color data or the WPB data.

During 20 ns ($t_{RSC}$) following this command, the μPD481850 can not accept any other commands.

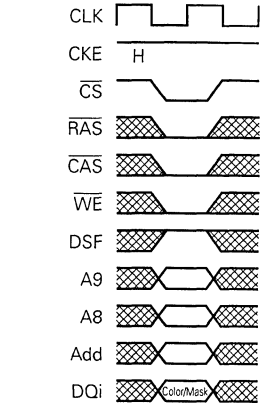



**Fig. 11 Mode register set command**

**Masked block write command**

($\overline{CS}$, $\overline{CAS}$, $\overline{WE}$ = Low, $\overline{RAS}$, DSF = High)

This command activates 8-column block write function. This command assumes as burst length = 1. Write data comes from color register, column address mask data is input from DQi in this command.

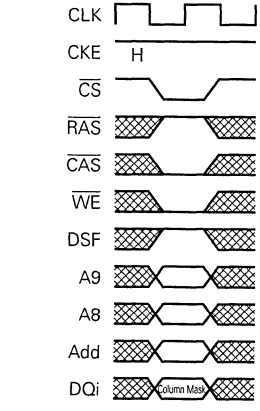
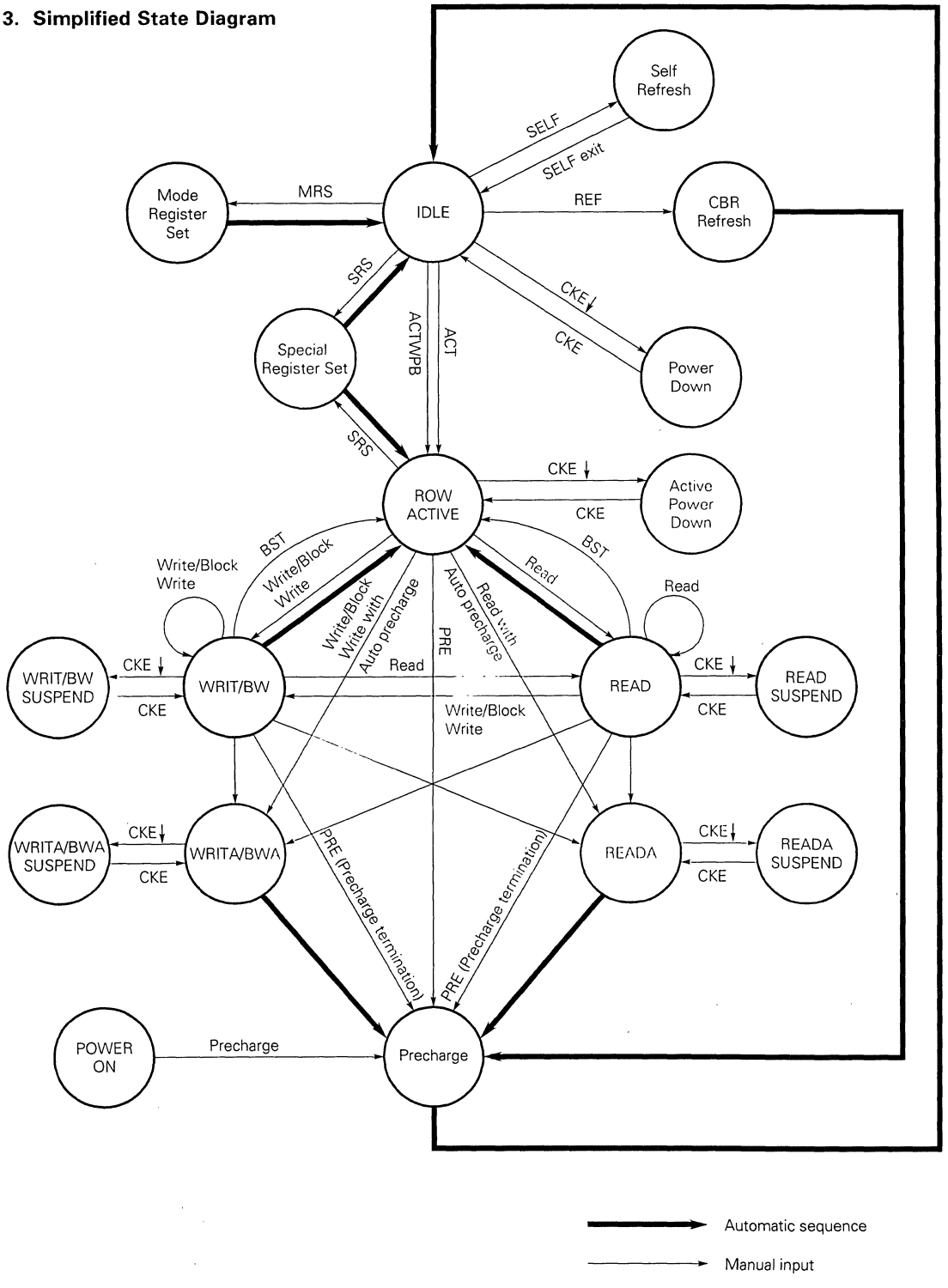



**Fig. 12 Mode register set command**

### 3. Simplified State Diagram



## 4. Truth Table

### 4.1 Command Truth Table

| Function | Symbol | CKE | | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | n–1 | n | | | | | | A9 | A8 | A7 - A0 |
| Device deselect | DESL | H | × | H | × | × | × | × | × | × | × |
| No operation | NOP | H | × | L | H | H | H | L | × | × | × |
| Burst stop in full page | BST | H | × | L | H | H | L | L | × | × | × |
| Read | READ | H | × | L | H | L | H | L | BA | L | CA |
| Read with auto precharge | READA | H | × | L | H | L | H | L | BA | H | CA |
| Write | WRIT | H | × | L | H | L | L | L | BA | L | CA |
| Write with auto precharge | WRITA | H | × | L | H | L | L | L | BA | H | CA |
| Masked block write | BW | H | × | L | H | L | L | H | BA | L | CA |
| Masked block write with auto precharge | BWA | H | × | L | H | L | L | H | BA | H | CA |
| Bank activate | ACT | H | × | L | L | H | H | L | BA | RA | |
| Bank activate with WPB enable | ACTWPB | H | × | L | L | H | H | H | BA | RA | |
| Precharge select bank | PRE | H | × | L | L | H | L | L | BA | L | × |
| Precharge all banks | PALL | H | × | L | L | H | L | L | × | H | × |
| Mode register set | MRS | H | × | L | L | L | L | L | OP. CODE | | |
| Special register set | SRS | H | × | L | L | L | L | H | OP. CODE | | |

**Remark** Legend:
H = High level, L = Low level, × = High or Low level (Don't care), BA = Bank address (A9), RA = Row address, CA = Column address

### 4.2 DQM Truth Table

| Function | Symbol | CKE | | DQMi |
|---|---|---|---|---|
| | | n–1 | n | |
| Data write/output enable | ENBi | H | × | L |
| Data mask/output disable | MASKi | H | × | H |

**Remark** Legend:
H = High level, L = Low level, × = High or Low level (Don't care), i = 0, 1, 2, 3

### 4.3 CKE Truth Table

| Current state | Function | Symbol | CKE | | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | n-1 | n | | | | | | |
| Activating | Clock suspend mode entry | | H | L | × | × | × | × | × | × |
| Any | Clock suspend | | L | L | × | × | × | × | × | × |
| Clock suspend | Clock suspend mode exit | | L | H | × | × | × | × | × | × |
| Idle | CBR refresh command | REF | H | H | L | L | L | H | L | × |
| Idle | Self refresh entry | SELF | H | L | L | L | L | H | L | × |
| Self refresh | Self refresh exit | | L | H | L | H | H | H | × | × |
| | | | L | H | H | × | × | × | × | × |
| Idle | Power down entry | | H | L | × | × | × | × | × | × |
| Power down | Power down exit | | L | H | × | × | × | × | × | × |

**Remark** Legend:

H = High Level, L = Low level, × = High or Low level (Don't care)

### 4.4 Operative Command Table[Note 1]

(1/7)

| Current state | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | Command | Action | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Idle | H | × | × | × | × | × | DESL | Nop or Power down | 2 |
| | L | H | H | H | × | × | NOP | Nop or Power down | 2 |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | 3 |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | 3 |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | 3 |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | 3 |
| | L | L | H | H | H | BA, RA | ACTWPB | Bank active with WPB: Latch RA | |
| | L | L | H | H | L | BA, RA | ACT | Bank active: Latch RA | |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | Nop | 11 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | CBR refresh/Self refresh | 4, 12 |
| | L | L | L | L | H | Op-Code | SRS | Special register access | |
| | L | L | L | L | L | Op-Code | MRS | Mode register access | 12 |
| Bank active | H | × | × | × | × | × | DESL | Nop | |
| | L | H | H | H | × | × | NOP | Nop | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | 3 |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | Begin read; Latch CA: Determine AP | 5 |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | Begin block write: Latch CA: Determine AP | 5 |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | Begin write; Latch CA: Determine AP | 5 |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | Precharge | 6 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | Special register access | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |

(2/7)

| Current state | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | Command | Action | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Read | H | × | × | × | × | × | DESL | Continue burst to end → Bank active | |
| | L | H | H | H | × | × | NOP | Continue burst to end → Bank active | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | 1, 2, 4, 8 burst length; Nop (Continue burst to end → Bank active)<br>Full page burst; Burst stop → Bank active | |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | Term burst, new read: Determine AP | 7 |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | Term burst, Start block write: Determine AP | 7, 8 |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | Term burst, start write: Determine AP | 7, 8 |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | Term burst, precharge timing for reads | |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | ILLEGAL | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |
| Write/Block write | H | × | × | × | × | × | DESL | Continue burst to end → Write recovering | |
| | L | H | H | H | × | × | NOP | Continue burst to end → Write recovering | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | 1, 2, 4, 8 burst length; Nop (Continue burst to end → Bank active)<br>Full page burst; Burst stop → Bank active | |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | Term burst, start read: Determine AP | 7, 8 |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | Term burst, new block write: Determine AP | 7 |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | Term burst, new write: Determine AP | 7 |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | Term burst, precharge timing for writes | 3, 9 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | ILLEGAL | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |

(3/7)

| Current state | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | Command | Action | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Read with auto precharge | H | × | × | × | × | × | DESL | Continue burst to end → precharging | |
| | L | H | H | H | × | × | NOP | Continue burst to end → precharging | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | ILLEGAL | 3 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | ILLEGAL | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |
| Write/Block write with auto precharge | H | × | × | × | × | × | DESL | Continue burst to end → Write recovering with auto precharge | |
| | L | H | H | H | × | × | NOP | Continue burst to end → Write recovering with auto precharge | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | ILLEGAL | 3 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | ILLEGAL | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |

(4/7)

| Current state | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | Command | Action | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Precharging | H | × | × | × | × | × | DESL | Nop → Enter idle after $t_{RP}$ | |
| | L | H | H | H | × | × | NOP | Nop → Enter idle after $t_{RP}$ | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | 3 |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | 3 |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | 3 |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | 3 |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | Nop → Enter idle after $t_{RP}$ | 11 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | Special register access | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |
| Bank activating ($t_{RCD}$) | H | × | × | × | × | × | DESL | Nop → Enter bank active after $t_{RCD}$ | |
| | L | H | H | H | × | × | NOP | Nop → Enter bank active after $t_{RCD}$ | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | 3 |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | 3 |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | 3 |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | 3 |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3, 10 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3, 10 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | ILLEGAL | 3 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | Special register access | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |

(5/7)

| Current state | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | Command | Action | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Write recovering ($t_{DPL}$) | H | × | × | × | × | × | DESL | Nop → Enter bank active after $t_{DPL}$ | |
| | L | H | H | H | × | × | NOP | Nop → Enter bank active after $t_{DPL}$ | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | 3 |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | Begin read; Latch CA; Determine AP | 8 |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | Begin block write; Latch CA; Determine AP | |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | Begin write; Latch CA; Determine AP | |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | ILLEGAL | 3 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | Special register access | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |
| Write recovering with auto precharge | H | × | × | × | × | × | DESL | Nop → Enter precharge after $t_{DPL}$ | |
| | L | H | H | H | × | × | NOP | Nop → Enter precharge after $t_{DPL}$ | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | 3, 8 |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | 3 |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | 3 |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | 3 |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | 3 |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | ILLEGAL | 3 |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | Special register access | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |

(6/7)

| Current state | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | Command | Action | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Refreshing | H | × | × | × | × | × | DESL | Nop → Enter idle after $t_{RC}$ | |
| | L | H | H | H | × | × | NOP | Nop → Enter idle after $t_{RC}$ | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | ILLEGAL | |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | ILLEGAL | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |
| Mode register accessing | H | × | × | × | × | × | DESL | Nop → Enter idle after $t_{RSC}$ | |
| | L | H | H | H | × | × | NOP | Nop → Enter idle after $t_{RSC}$ | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | ILLEGAL | |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | ILLEGAL | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |

(7/7)

| Current state | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | Command | Action | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Special mode register accessing | H | × | × | × | × | × | DESL | Nop → Enter previous state after $t_{RSC}$ | |
| | L | H | H | H | × | × | NOP | Nop → Enter previous state after $t_{RSC}$ | |
| | L | H | H | L | H | × | Undefined | ILLEGAL | |
| | L | H | H | L | L | × | BST | ILLEGAL | |
| | L | H | L | H | H | × | Undefined | ILLEGAL | |
| | L | H | L | H | L | BA, CA, A8 | READ/READA | ILLEGAL | |
| | L | H | L | L | H | BA, CA, A8 | BW/BWA | ILLEGAL | |
| | L | H | L | L | L | BA, CA, A8 | WRIT/WRITA | ILLEGAL | |
| | L | L | H | H | H | BA, RA | ACTWPB | ILLEGAL | |
| | L | L | H | H | L | BA, RA | ACT | ILLEGAL | |
| | L | L | H | L | H | × | Undefined | ILLEGAL | |
| | L | L | H | L | L | BA, A8 | PRE/PALL | ILLEGAL | |
| | L | L | L | H | H | × | Undefined | ILLEGAL | |
| | L | L | L | H | L | × | REF/SELF | ILLEGAL | |
| | L | L | L | L | H | Op-Code | SRS | ILLEGAL | |
| | L | L | L | L | L | Op-Code | MRS | ILLEGAL | |

**Notes** 1. All entries assume that CKE was active (High level) during the preceding clock cycle.

2. If both banks are idle, and CKE is inactive (Low level), μPD481850 will enter Power down mode. All input buffers except CKE will be disabled.
3. Illegal to bank in specified states; Function may be legal in the bank indicated by Bank Address (BA), depending on the state of that bank.
4. If both banks are idle, and CKE is inactive (Low level), μPD481850 will enter Self refresh. All input buffers except CKE will be disabled.
5. Illegal if $t_{RCD}$ is not satisfied.
6. Illegal if $t_{RAS}$ is not satisfied.
7. Must satisfy burst interrupt condition.
8. Must satisfy bus contention, bus turn around, and/or write recovery requirements.
9. Must mask preceding data which don't satisfy $t_{DPL}$.
10. Illegal if $t_{RRD}$ is not satisfied.
11. Nop to bank precharging or in idle state. May precharge bank(s) indicated by BA (and A8).
12. Illegal if any bank is not idle.

**Remark** Legend:

H = High level, L = Low level, × = High or Low level (Don't care), V = Valid Data input,
BA = Bank address (A9), A8 = Precharge select, RA = Row address, CA = Column address,
Term = Terminate, AP = Auto precharge, NOP = No operation,
ILLEGAL = Device operation and/or data-integrity are not guaranteed

### 4.5 Command Truth Table for CKE

| Current state | CKE | | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | Address | Action | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| | n–1 | n | | | | | | | | |
| Self refresh (S.R.) | H | × | × | × | × | × | × | × | INVALID, CLK(n-1) would exit S.R. | |
| | H | H | H | × | × | × | × | × | S.R. Recovery | 1 |
| | L | H | L | H | H | × | × | × | S.R. Recovery | 1 |
| | L | H | L | H | L | × | × | × | ILLEGAL | 1 |
| | L | H | L | L | × | × | × | × | ILLEGAL | 1 |
| | L | L | × | × | × | × | × | × | Maintain S.R. | |
| Self refresh recovery | H | H | H | × | × | × | × | × | Idle after $t_{RC}$ | |
| | H | H | L | H | H | H | × | × | Idle after $t_{RC}$ | |
| | H | H | L | H | H | L | × | × | ILLEGAL | |
| | H | H | L | H | L | × | × | × | ILLEGAL | |
| | H | H | L | L | × | × | × | × | ILLEGAL | |
| | H | L | H | × | × | × | × | × | Begin clock suspend next cycle | |
| | H | L | L | H | H | H | × | × | Begin clock suspend next cycle | |
| | H | L | L | H | H | L | × | × | ILLEGAL | |
| | H | L | L | H | L | × | × | × | ILLEGAL | |
| | H | L | L | L | × | × | × | × | ILLEGAL | |
| | L | H | × | × | × | × | × | × | Exit clock suspend next cycle | 1 |
| | L | L | × | × | × | × | × | × | Maintain clock suspend | |
| Power down (P.D.) | H | × | × | × | × | × | × | | INVALID, CLK(n-1) would exit P.D. | |
| | L | H | × | × | × | × | × | × | EXIT P.D. → Idle | 1 |
| | L | L | × | × | × | × | × | × | Maintain power down mode | |
| Both banks idle | H | H | H | × | × | × | × | × | Refer to operations in Operative Command Table | |
| | H | H | L | H | × | × | × | × | Refer to operations in Operative Command Table | |
| | H | H | L | L | H | × | × | × | Refer to operations in Operative Command Table | |
| | H | H | L | L | L | H | L | × | Refresh | |
| | H | H | L | L | L | L | × | Op-Code | Refer to operations in Operative Command Table | |
| | H | L | H | × | × | × | × | × | Refer to operations in Operative Command Table | |
| | H | L | L | H | × | × | × | × | Refer to operations in Operative Command Table | |
| | H | L | L | L | H | × | × | × | Refer to operations in Operative Command Table | |
| | H | L | L | L | L | H | L | × | Self refresh | 2 |
| | H | L | L | L | L | L | × | Op-Code | Refer to operations in Operative Command Table | |
| | L | × | × | × | × | × | × | × | Power down | 2 |
| Any state other than listed above | H | H | × | × | × | × | × | × | Refer to operations in Operative Command Table | |
| | H | L | × | × | × | × | × | × | Begin clock suspend next cycle | 3 |
| | L | H | × | × | × | × | × | × | Exit clock suspend next cycle | |
| | L | L | × | × | × | × | × | × | Maintain clock suspend | |

**Notes 1.** CKE Low to High transition will re-enable CLK and other inputs asynchronously. A minimum setup time must be satisfied before any command other than EXIT.

**2.** Power down and Self refresh can be entered only from the both banks idle state.

**3.** Must be legal command as defined in Operative Command Table.

**Remark** Legend:

H = High level, L = Low level, × = High or Low level (Don't care)

### 4.6 Command Truth Table for Two Banks Operation

| $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | DSF | A9 (BA) | A8 | A7 - A0 | Action | "FROM" State[Note 1] | "TO" State[Note 2] |
|---|---|---|---|---|---|---|---|---|---|---|
| H | × | × | × | × | × | × | × | NOP | Any | Any |
| L | H | H | H | L | × | × | × | NOP | Any | Any |
| L | H | H | L | L | × | × | × | BST | (R/W/A)0(I/A)1 | A0(I/A)1 |
| | | | | | | | | | I0(I/A)1 | I0(I/A)1 |
| | | | | | | | | | (R/W/A)1(I/A)0 | A1(I/A)0 |
| | | | | | | | | | I1(I/A)0 | I1(I/A)0 |
| L | H | L | H | L | H | H | CA | Read | (R/W/A)1(I/A)0 | RP1(I/A)0 |
| | | | | | H | H | CA | | A1(R/W)0 | RP1A0 |
| | | | | | H | L | CA | | (R/W/A)1(I/A)0 | R1(I/A)0 |
| | | | | | H | L | CA | | A1(R/W)0 | R1A0 |
| | | | | | L | H | CA | | (R/W/A)0(I/A)1 | RP0(I/A)1 |
| | | | | | L | H | CA | | A0(R/W)1 | RP0A1 |
| | | | | | L | L | CA | | (R/W/A)0(I/A)1 | R0(I/A)1 |
| | | | | | L | L | CA | | A0(R/W)1 | R0A1 |
| L | H | L | L | L/H | H | H | CA | Write/Block Write | (R/W/A)1(I/A)0 | WP1(I/A)0 |
| | | | | | H | H | CA | | A1(R/W)0 | WP1A0 |
| | | | | | H | L | CA | | (R/W/A)1(I/A)0 | W1(I/A)0 |
| | | | | | H | L | CA | | A1(R/W)0 | W1A0 |
| | | | | | L | H | CA | | (R/W/A)0(I/A)1 | WP0(I/A)1 |
| | | | | | L | H | CA | | A0(R/W)1 | WP0A1 |
| | | | | | L | L | CA | | (R/W/A)0(I/A)1 | W0(I/A)1 |
| | | | | | L | L | CA | | A0(R/W)1 | W0A1 |
| L | L | H | H | L/H | H | RA | | Activate Row | I1Any0 | A1Any0 |
| | | | | | L | RA | | | I0Any1 | A0Any1 |
| L | L | H | L | L | × | H | × | Precharge | (R/W/A/I)0(I/A)1 | I0I1 |
| | | | | | × | H | × | | (R/W/A/I)1(I/A)0 | I1I0 |
| | | | | | H | L | × | | (R/W/A/I)1(I/A)0 | I1(I/A)0 |
| | | | | | H | L | × | | (I/A)1(R/W/A/I)0 | I1(R/W/A/I)0 |
| | | | | | L | L | × | | (R/W/A/I)0(I/A)1 | I0(I/A)1 |
| | | | | | L | L | × | | (I/A)0(R/W/A/I)1 | I0(R/W/A/I)1 |
| L | L | L | H | L | × | × | × | Refresh | I0I1 | I0I1 |
| L | L | L | L | L | Op-Code | | | Mode Register Access | I0I1 | I0I1 |
| L | L | L | L | H | Op-Code | | | Special Register Access | (I/A)0(I/A)1 | (I/A)0(I/A)1 |

**Notes 1.** If the μPD481850 is in a state other than above listed in the "From State" column, the command is illegal.

**2.** The states listed under "To" might not be entered on the next clock cycle. Timing restrictions apply.

**Remark** Legend:
H = High level, L = Low level, × = High or Low level (Don't care),
BA = Bank address (A9), I = Idle, A = Bank active,
R = Read with No precharge (No precharge is posted)
W = Write with No precharge (No precharge is posted)
RP = Read with auto precharge (No precharge is posted)
WP = Write with auto precharge (No precharge is posted)
Any = Any State
X0Y1 = Bank0 is in state "X", Bank1 = in state "Y"
(X/Y)0Z1 = Z1(X/Y)0 = Bank0 is in state "X" or "Y", Bank1 is in state "Z"

## 5. Initialization

The synchronous GRAM is initialized in the power-on sequence according to the following.

(1) To stabilize internal circuits, when power is applied, a 100-μs or longer pause must precede any signal toggling.

(2) After the pause, both banks must be precharged using the Precharge command (The Precharge all banks command is convenient).

(3) Once the precharge is completed and the minimum $t_{RP}$ is satisfied, the mode register can be programmed.
After the mode register set cycle, $t_{RSC}$ (20 ns minimum) pause must be satisfied as well.

(4) Two or more CBR (Auto) refresh must be performed.

**Remarks 1.** The sequence of Mode register programming and Refresh above may be transposed.
**2.** CKE and DQM may be held high until the Precharge command is asserted to ensure data-bus Hi-Z.

## 6. Programming the Mode Register

The mode register is programmed by the Mode register set command using address bits A9 through A0 as data inputs. The register retains data until it is reprogrammed or the device loses power.

The mode register has four fields;

Options : A9 through A7
$\overline{CAS}$ latency: A6 through A4
Wrap type : A3
Burst length: A2 through A0

Following mode register programming, no command can be asserted before at least 20 ns ($t_{RSC}$) have elapsed.

### $\overline{CAS}$ Latency

$\overline{CAS}$ latency is the most critical of the parameters being set. It tells the device how many clocks must elapse before the data will be available.

The value is determined by the frequency of the clock and the speed grade of the device. The table on page 52 shows the relationship of $\overline{CAS}$ latency to the clock period and the speed grade of the device.

### Burst Length

Burst Length is the number of words that will be output or input in a read or write cycle. After a read burst is completed, the output bus will become Hi-Z.

The burst length is programmable as 1, 2, 4, 8 or full page (256 columns).

### Wrap Type (Burst Sequence)

The wrap type specifies the order in which the burst data will be addressed. This order is programmable as either "Sequential" or "Interleave". The method chosen will depend on the type of CPU in the system.

Some microprocessor cache system are optimized for sequential addressing and others for interleaved addressing. The table on the page 27 shows the addressing sequence for each burst length using them. Sequential mode supports bursts of 1, 2, 4 and 8, Interleave mode supports bursts of 4 and 8. Additionally, sequential sequence supports the full page length.

## 7. Mode Register

| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | | | | | | | | JEDEC Standard Test Set (refresh counter test) |
| 1 | 0 | 0 | LTMODE | | | WT | BL | | | Burst Read and Single Write (for Write Through Cache) |
| | 1 | 0 | | | | | | | | Use in future |
| × | 1 | 1 | V | V | V | V | V | V | V | Vender Specific |
| 0 | 0 | 0 | LTMODE | | | WT | BL | | | Mode Register Set |

V = Valid
× = Don't care

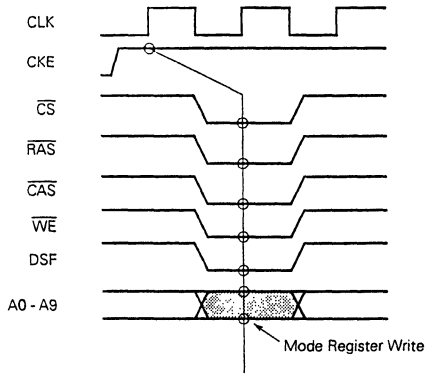
| | Bits2-0 | WT = 0 | WT = 1 |
|---|---|---|---|
| Burst length | 000 | 1 | R |
| | 001 | 2 | R |
| | 010 | 4 | 4 |
| | 011 | 8 | 8 |
| | 100 | R | R |
| | 101 | R | R |
| | 110 | R | R |
| | 111 | Full page | R |

| | | |
|---|---|---|
| Wrap type | 0 | Sequential |
| | 1 | Interleave |

| | Bits6-4 | $\overline{CAS}$ latency |
|---|---|---|
| Latency mode | 000 | R |
| | 001 | 1 |
| | 010 | 2 |
| | 011 | 3 |
| | 100 | R |
| | 101 | R |
| | 110 | R |
| | 111 | R |

**Remark** R: Reserved

### Mode Register Write Timing



### 7.1 Burst Length and Sequence

[Burst of Two]

| Starting Address (column address A0, binary) | Sequential Addressing Sequence (decimal) | Interleave Addressing Sequence (decimal) |
|---|---|---|
| 0 | 0, 1 | Not support |
| 1 | 1, 0 | Not support |

[Burst of Four]

| Starting Address (column address A1 - A0, binary) | Sequential Addressing Sequence (decimal) | Interleave Addressing Sequence (decimal) |
|---|---|---|
| 00 | 0, 1, 2, 3 | 0, 1, 2, 3 |
| 01 | 1, 2, 3, 0 | 1, 0, 3, 2 |
| 10 | 2, 3, 0, 1 | 2, 3, 0, 1 |
| 11 | 3, 0, 1, 2 | 3, 2, 1, 0 |

[Burst of Eight]

| Starting Address (column address A2 - A0, binary) | Sequential Addressing Sequence (decimal) | Interleave Addressing Sequence (decimal) |
|---|---|---|
| 000 | 0, 1, 2, 3, 4, 5, 6, 7 | 0, 1, 2, 3, 4, 5, 6, 7 |
| 001 | 1, 2, 3, 4, 5, 6, 7, 0 | 1, 0, 3, 2, 5, 4, 7, 6 |
| 010 | 2, 3, 4, 5, 6, 7, 0, 1 | 2, 3, 0, 1, 6, 7, 4, 5 |
| 011 | 3, 4, 5, 6, 7, 0, 1, 2 | 3, 2, 1, 0, 7, 6, 5, 4 |
| 100 | 4, 5, 6, 7, 0, 1, 2, 3 | 4, 5, 6, 7, 0, 1, 2, 3 |
| 101 | 5, 6, 7, 0, 1, 2, 3, 4 | 5, 4, 7, 6, 1, 0, 3, 2 |
| 110 | 6, 7, 0, 1, 2, 3, 4, 5 | 6, 7, 4, 5, 2, 3, 0, 1 |
| 111 | 7, 0, 1, 2, 3, 4, 5, 6 | 7, 6, 5, 4, 3, 2, 1, 0 |

**Full page burst is an extension of the above tables of Sequential Addressing, with the length being 256.**

## 8. Programming the Special Register

The special register is programming by the Special register set command using address bits A9 through A0 and data bits DQ0 through DQ31. The color and mask register retain data until it is reprogrammed or the device losed power.

The special register has four fields.

Reserved : A9 through A7
Color register : A6
Mask register : A5
Reserved : A4 through A0

Following special register programming, no command can be asserted before at least 20 ns ($t_{RSC}$) have elapsed.

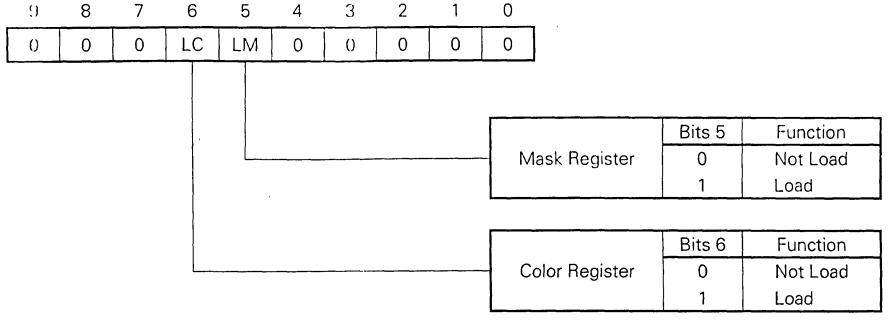
**Color Register**

Color register is used as write data in Block Write cycle. In Special Register set command, if A5 is "0" and A6 is "1", the color register is selected. And the data of DQ0 through DQ31 is stored to color register as color data (write data).

**Mask Register**

Mask register is used as write mask data in Write and Block Write cycle. In Special Register set command, if A5 is "1" and A6 is "0", the mask register is selected. And the data of DQ0 through DQ31 is stored to mask register as write mask data.

**Special Register**




**Remark** If LC and LM are both high (1), data of Mask and Color register will be unknown.

## 9. Address Bits of Bank-Select and Precharge

Row (Activate command)

| A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
|---|---|---|---|---|---|---|---|---|---|

| | |
|---|---|
| 0 | Select Bank A "Activate" command |
| 1 | Select Bank B "Activate" command |

(Precharge command)

| A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
|---|---|---|---|---|---|---|---|---|---|

| A8 | A9 | Result |
|---|---|---|
| 0 | 0 | Precharge Bank A |
| 0 | 1 | Precharge Bank B |
| 1 | × | Precharge All Banks |

× : Don't care

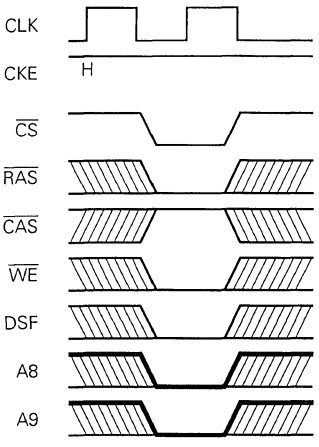
Col. ($\overline{CAS}$ strobes)

| A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
|---|---|---|---|---|---|---|---|---|---|

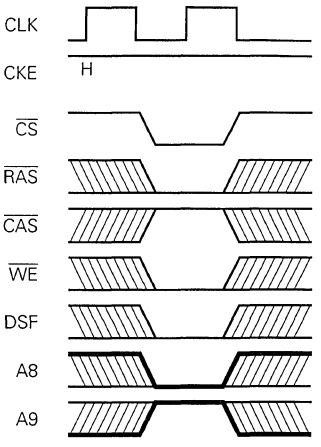
| | |
|---|---|
| 0 | disables Auto-Precharge (End of Burst) |
| 1 | enables Auto-Precharge (End of Burst) |

| | |
|---|---|
| 0 | enables Read/Write commands for Bank A |
| 1 | enables Read/Write commands for Bank B |

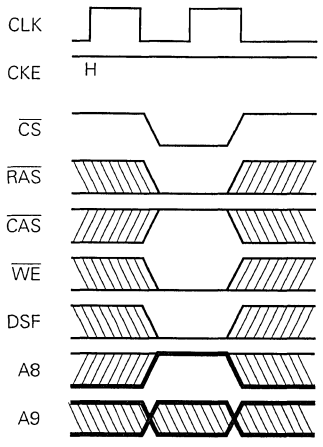
**Precharge for Bank A**




**Precharge for Bank B**




**Precharge for All Banks**



## 10. Precharge

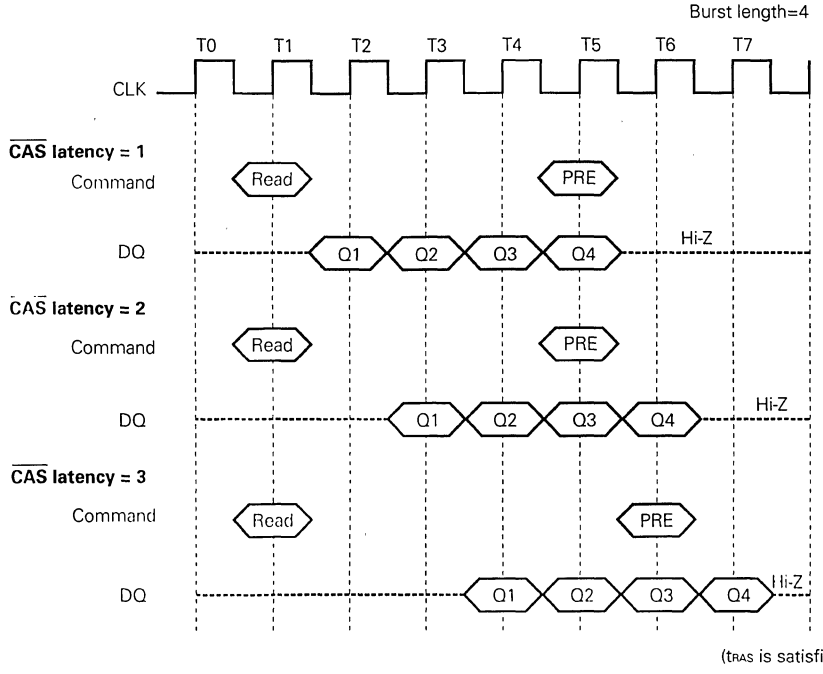
The precharge command can be asserted anytime after $t_{RAS(MIN.)}$ is satisfied.

Soon after the precharge command is asserted, precharge operation performed and the synchronous GRAM enters the idle state after $t_{RP}$ is satisfied. The parameter $t_{RP}$ is the time required to perform the precharge.

The earliest timing in a read cycle that a precharge command can be asserted without losing any data in the burst is as follows.

It is depending on the $\overline{CAS}$ latency.

$\overline{CAS}$ latency = 1 : At the same clock as the last read data.
$\overline{CAS}$ latency = 2 or 3 : One clock earlier than the last read data.




($t_{RAS}$ is satisfied)

In order to write all data to the memory cell correctly, the asynchronous parameter "$t_{DPL}$" must be satisfied. The $t_{DPL(MIN.)}$ specification defines the earliest time that a precharge command can be asserted. Minimum number of clocks are calculated by dividing $t_{DPL\ (MIN.)}$ with clock cycle time.

In summary, the precharge command can be asserted relative to reference clock that indicates the last data word is valid. In the following table, minus means clocks before the reference; plus means time after the reference.

| $\overline{CAS}$ latency | Read | Write |
|---|---|---|
| 1 | 0 | $+t_{DPL\ (MIN.)}$ |
| 2 | −1 | $+t_{DPL\ (MIN.)}$ |
| 3 | −1 | $+t_{DPL\ (MIN.)}$ |

## 11. Auto Precharge

During a read or write/block write command cycle, A8 controls whether auto precharge is selected. A8 high in the read or write/block write command (Read with Auto precharge command or Write with Auto precharge command/Block Write with Auto precharge command), auto precharge is selected and begins after the burst access automatically.

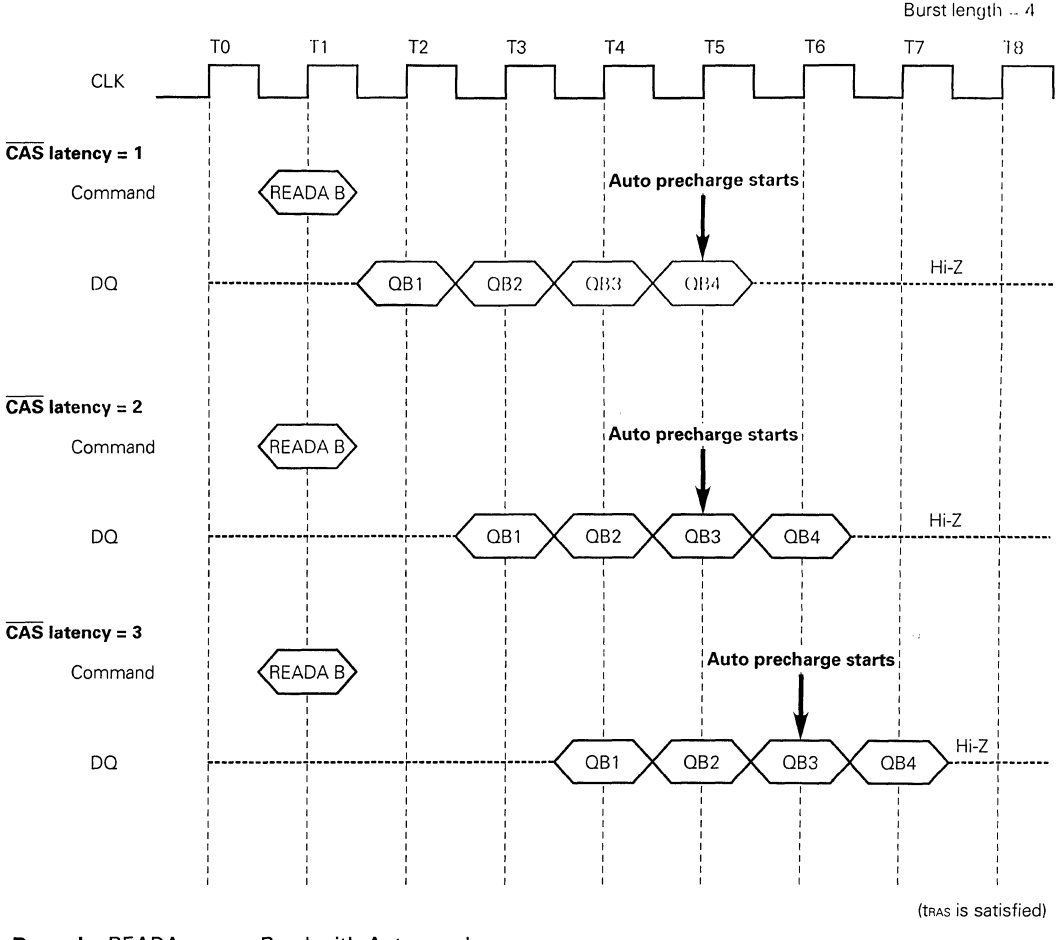
When the $t_{RAS}$ is not satisfied, the precharge does not start at above timing. And the precharge will start when the $t_{RAS}$ is satisfied.

The clock that begins the auto precharge cycle is depend on both the $\overline{CAS}$ latency programmed into the mode register and whether READ or WRITE/BLOCK WRITE cycle.

### 11.1 Read with Auto Precharge

When using auto precharge in READ cycle, knowing when the precharge starts is important because the next activate command to the bank being precharged cannot be executed until the precharge cycle ends. Once auto precharge has started, an activate command to the bank can be asserted after $t_{RP}$ has been satisfied.

During READ cycle, the auto precharge begins after $t_{RAS}$ and begins on the clock that indicates the last data word output during the burst is valid ($\overline{CAS}$ latency of 1) or one clock earlier ($\overline{CAS}$ latency of 2 or 3).

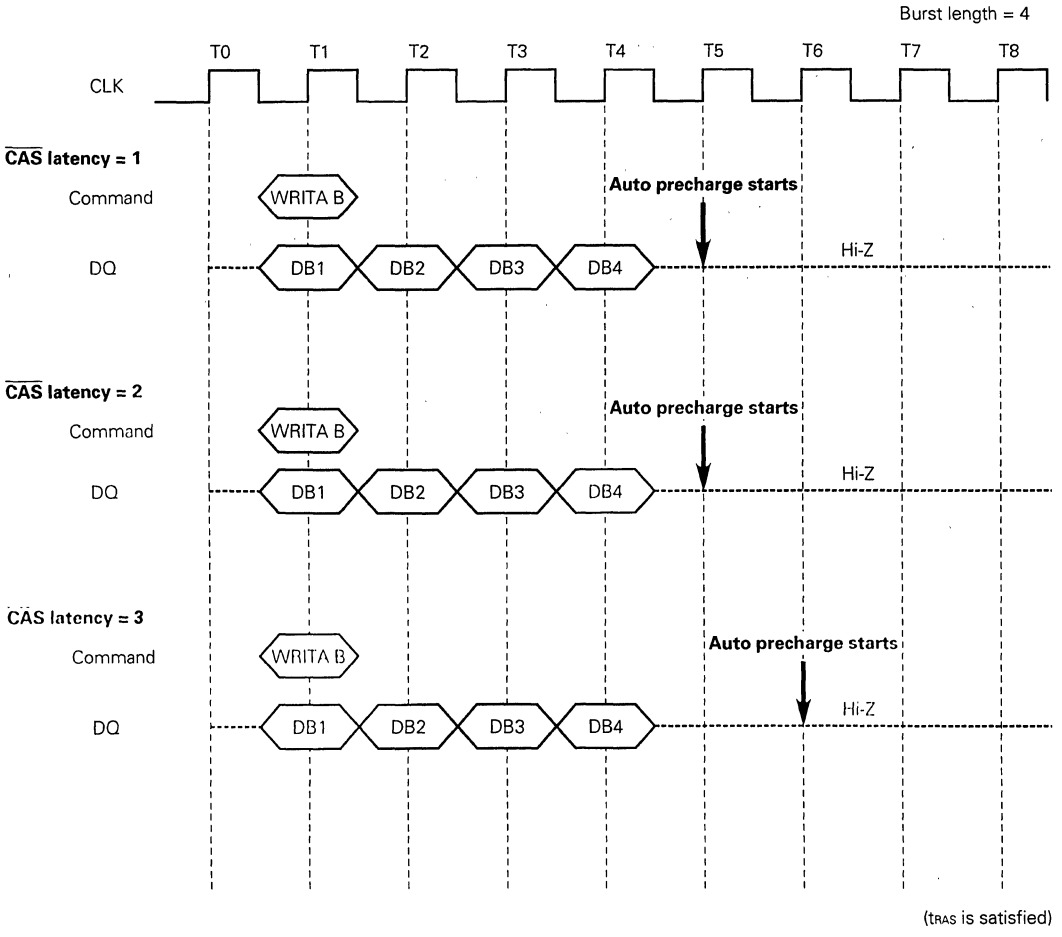



**Remark** READA means Read with Auto precharge

### 11.2 Write with Auto Precharge

In write cycle, the $t_{DAL}$ must be satisfied to assert the all commands to the bank being precharged. And it is not necessary to know when the precharge starts. In block write cycle, the $t_{BAL}$ must be satisfied to assert the all commands to the bank being precharged. And it is not necessary to know the precharge starts.

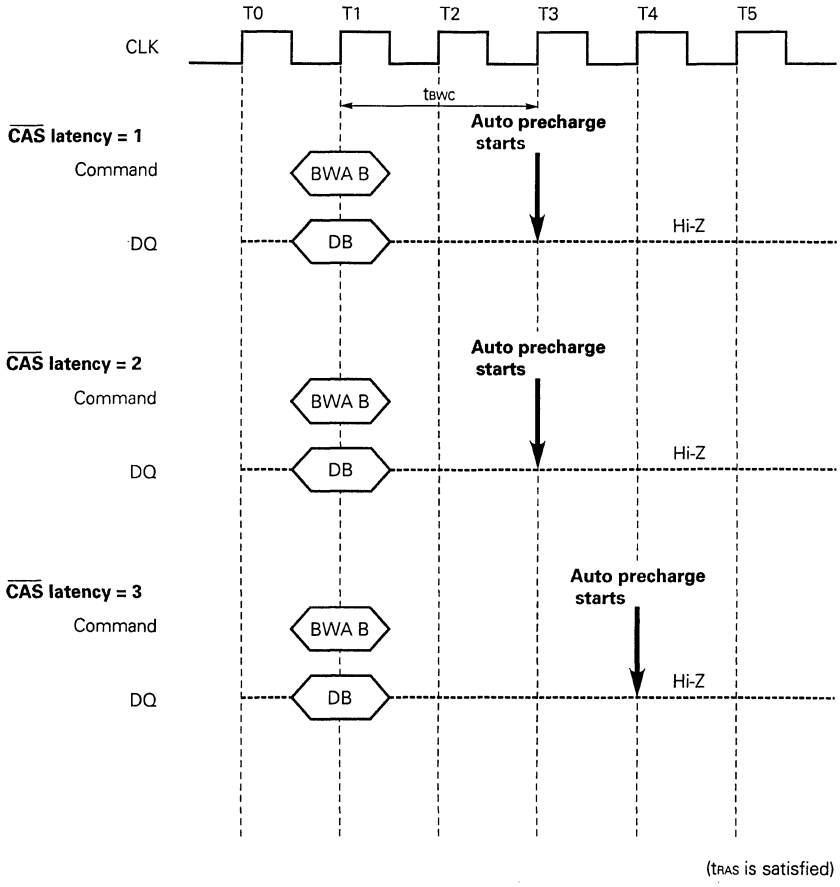
During WRITE cycle, the auto precharge begins after $t_{RAS}$ and begins one clock after the last data word input to the device ($\overline{CAS}$ latency of 1 or 2) or two clocks after ($\overline{CAS}$ latency of 3).




**Remark** WRITA means Write with Auto precharge

### 11.3 Block Write with Auto Precharge

During BLOCK WRITE cycle, the auto precharge begins one clock after the block write command to the device ($\overline{CAS}$ latency of 1) or two clocks after ($\overline{CAS}$ latency of 2) or three clocks after ($\overline{CAS}$ latency of 3).




(tRAS is satisfied)

In summary, the auto precharge cycle begins relative to a reference clock that indicates the last data word is valid.

In the table below, minus means clocks before the reference; plus means clocks after the reference.

| $\overline{CAS}$ latency | Read | Write | Block Write |
|---|---|---|---|
| 1 | 0 | +1 | +2 |
| 2 | −1 | +1 | +2 |
| 3 | −1 | +2 | +3 |

## 12. Write/Block Write with Write Per Bit

To use WPB operation

(1) Execute Special register set command and set WPB data (32 bits) to mask register.

(2) Execute Bank Activate with WPB enable command (ACTWPB) after $t_{RSC}$ (20 ns) period from Special register set command (SRS).

(3) Execute Write/Block write command after $t_{RCD}$ period from ACTWPB.

In case SRS command is executed in activate state to set new WPB data, it is necessary to take $t_{RSC}$ (20 ns) interval between SRS and Write/Block write command.

**Remark** Mask data = Mask register's data (WPB) + DQMi
DQMi is prior to Mask register's data (WPB)

## 13. Block Write

In block write cycle, write data from color register can be written in 8 columns at one write cycle.
It is also possible to execute Block write cycle with write per bit. Column Mask by DQi is available.

To use Block write operation

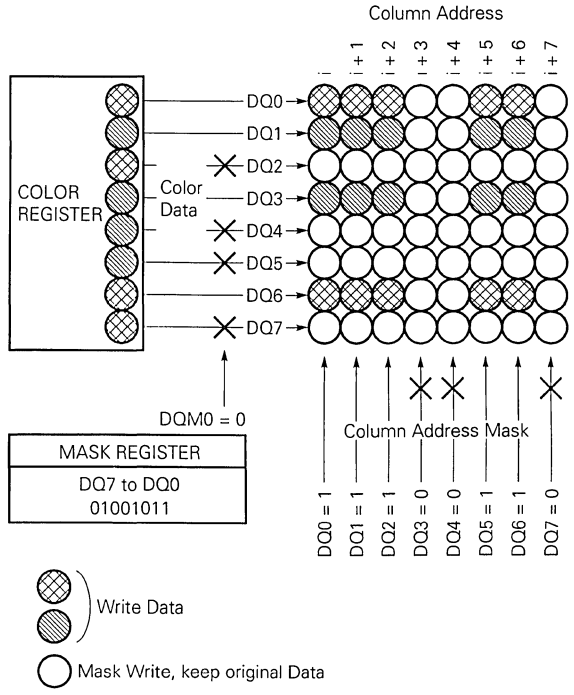
(1) Execute Special register set command and set color data (32 bits) to color register.

(2) Execute Bank Activate (ACT) or Bank Activate with WPB enable command (ACTWPB) after $t_{RSC}$ (20 ns) period from SRS.

(3) Execute Block write command after $t_{RCD}$ period from ACT or ACTWPB.

In case new Write/Block write is executed or, it is necessary to take $t_{BWC}$ interval from Block Write command to new Write/Block write command.

**Block Write Function**




**Remarks 1.** i is times of 8 numeric.

**2.** This diagram shows only for DQ0 - 7. The other DQ is similar as this.

**Column Mask**

DQ0 - 7 : Column Mask for DQ0 - 7
DQ8 - 15 : Column Mask for DQ8 - 15
DQ16 - 23: Column Mask for DQ16 - 23
DQ24 - 31: Column Mask for DQ24 - 31

**Write per Bit**

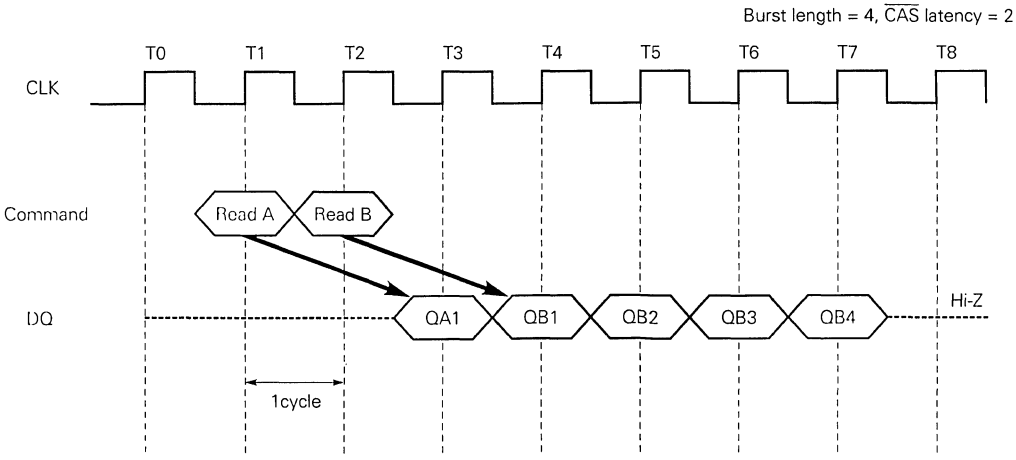
Mask data = Mask Register + DQMi
DQMi is prior to data of Mask Register.

## 14. Read/Write Command Interval

### 14.1 Read to Read Command Interval

During READ cycle, when new Read command is asserted, it will be effective after $\overline{CAS}$ latency, even if the previous READ operation does not completed. READ will be interrupted by another READ.

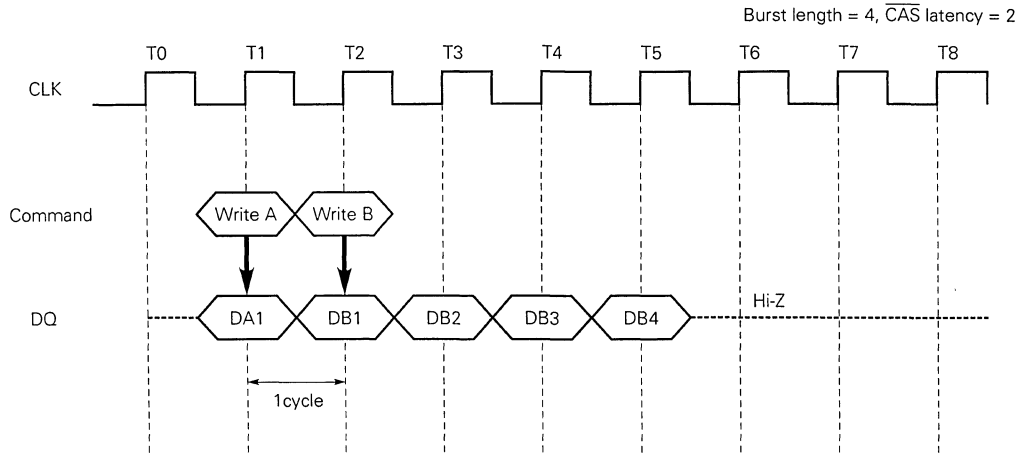
The interval between the commands is minimum 1 cycle. Each Read command can be asserted in every clock without any restriction.




### 14.2 Write to Write Command Interval

During WRITE cycle, when new Write command is asserted, the previous burst will terminate and the new burst will begin with a new Write commnad. WRITE will be interrupted by another WRITE.

The interval between the commands is minimum 1 cycle. Each Write command can be asserted in every clock without any restriction.

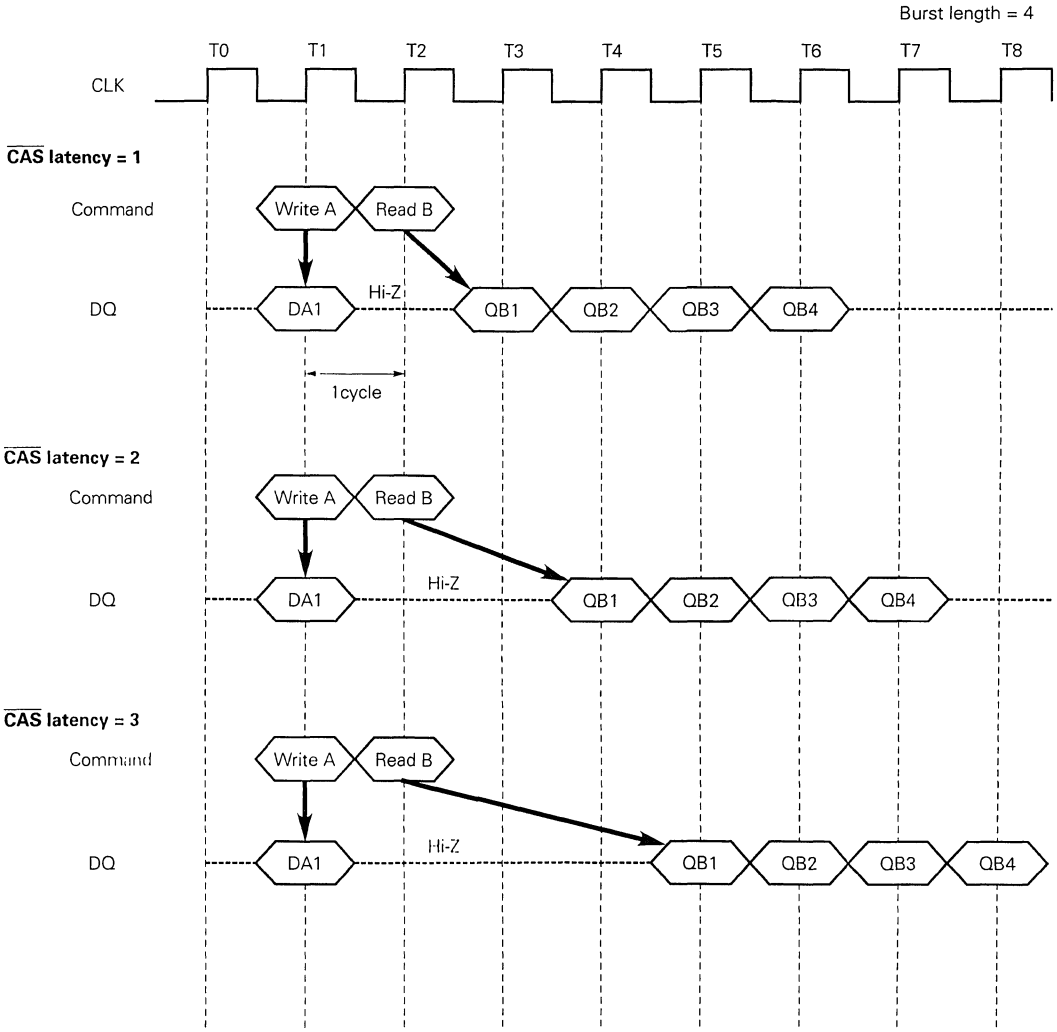


### 14.3 Write to Read Command Interval

Write command and Read command interval is also 1 cycle.

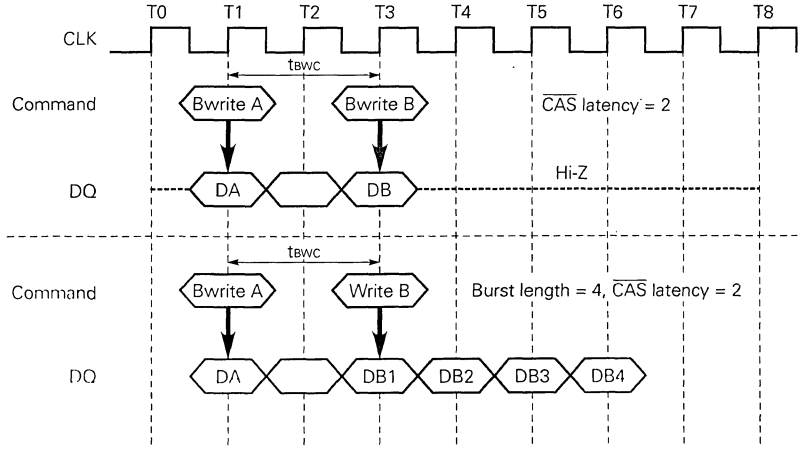
Only the write data before Read command will be written.

The data bus must be Hi-Z at least one cycle prior to the first DOUT.



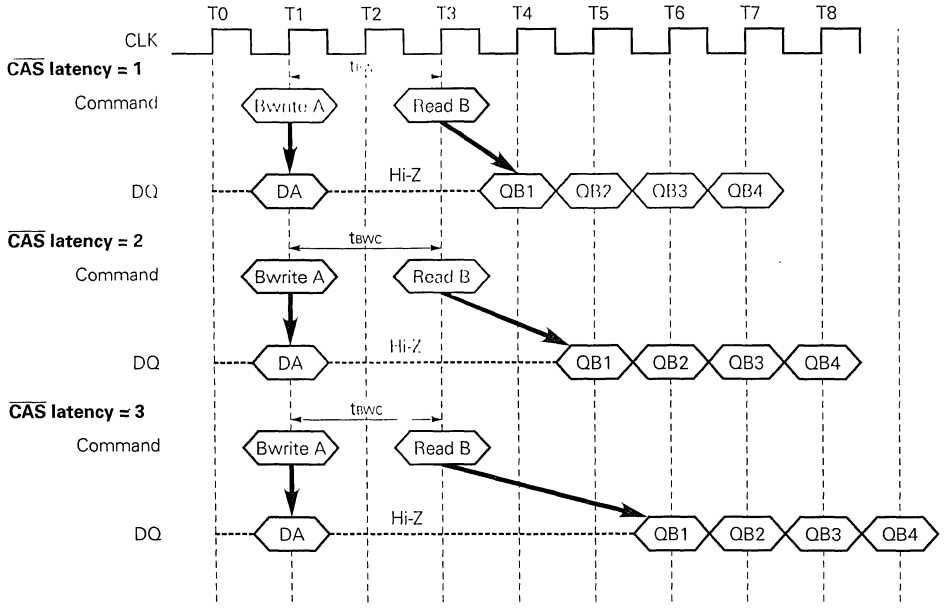
### 14.4 Block Write to Write or Write/Block Write Command Interval

The interval between BLOCK WRITE and new BLOCK WRITE or WRITE is $t_{BWC}$ or minimum 1 cycle. If $t_{CK}$ is less than $t_{BWC}$, NOP command should be issued for the cycle between BLOCK WRITE and the following WRITE or new BLOCK WRITE.




### 14.5 Block Write to Read Command Interval

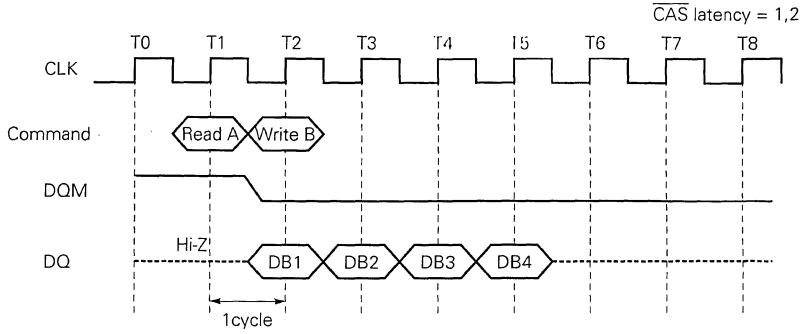
BLOCK WRITE command and READ command is also $t_{BWC}$ or minimum 1 cycle. The data bus must be Hi-Z at least one cycle prior to the first $D_{OUT}$.

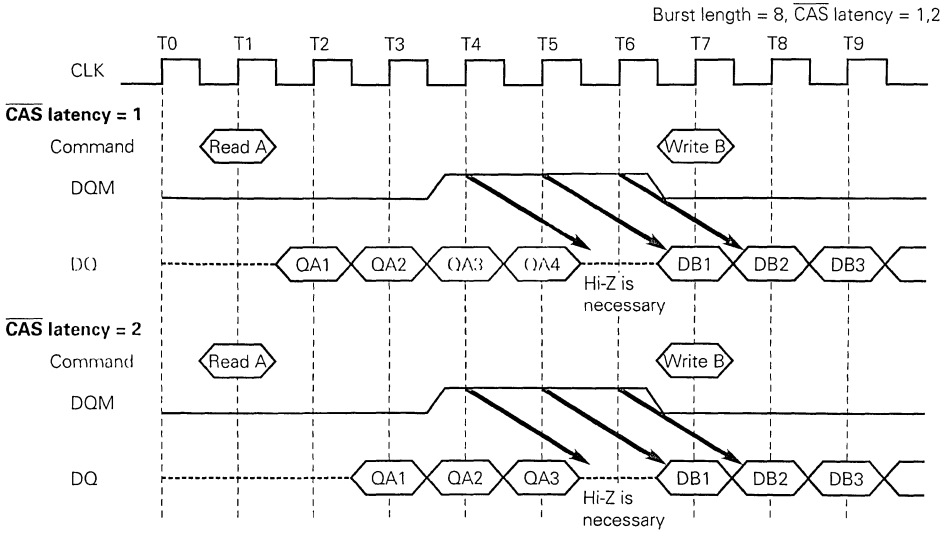


### 14.6 Read to Write/Block Write Command Interval

During READ cycle, Read can be interrupted by WRITE. But full page burst read can not be interrupted by WRITE. Full page burst read can be interrupted by Burst Stop command (BST) or Precharge command (Burst termination).

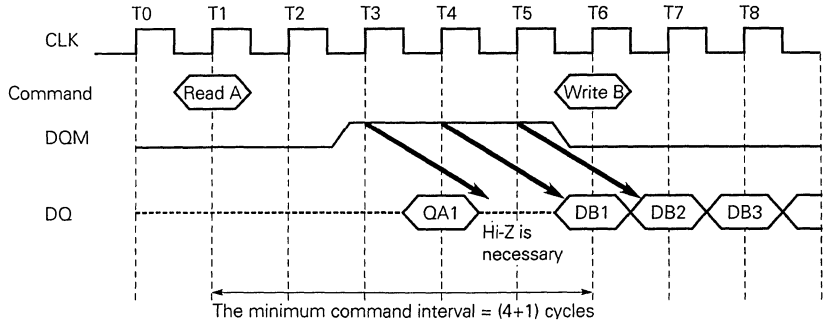
For $\overline{CAS}$ latency of 1 or 2, the READ and WRITE command interval is minimum 1 cycle. The data bus must be Hi-Z using DQM before WRITE to avoid data conflict. And DQM must be kept being High from at least 3 clocks to 1 clock before the Write command.



For $\overline{CAS}$ latency of 3, the READ and WRITE command interval is [Burst length + 1] cycles. The data bus must be Hi-Z using DQM before WRITE to avoid data conflict. And DQM must be kept being High from at least 3 clocks to 1 clock before the WRITE command.

**ex.) $\overline{CAS}$ latency = 3, Burst length = 4**



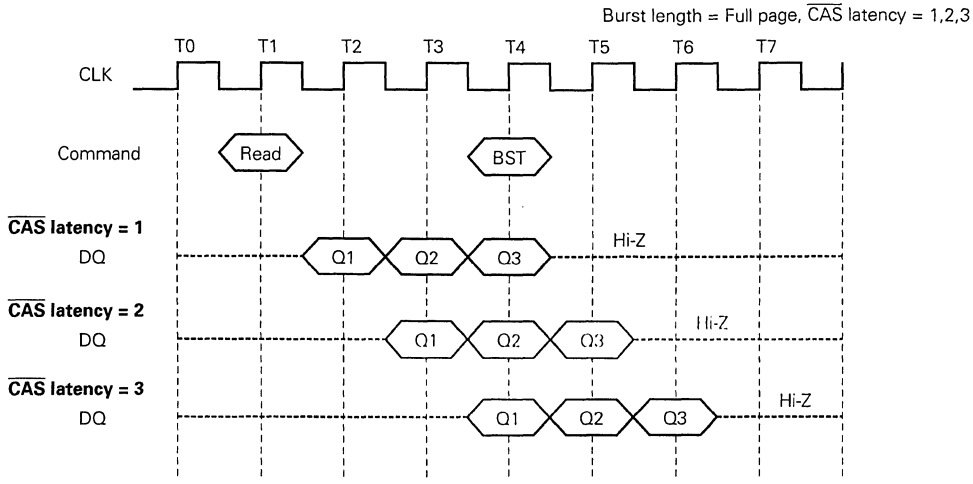
## 15. Burst Termination

Burst termination is to terminate a burst operation other than using a read or write command.

### 15.1 Burst Stop Command in Full Page

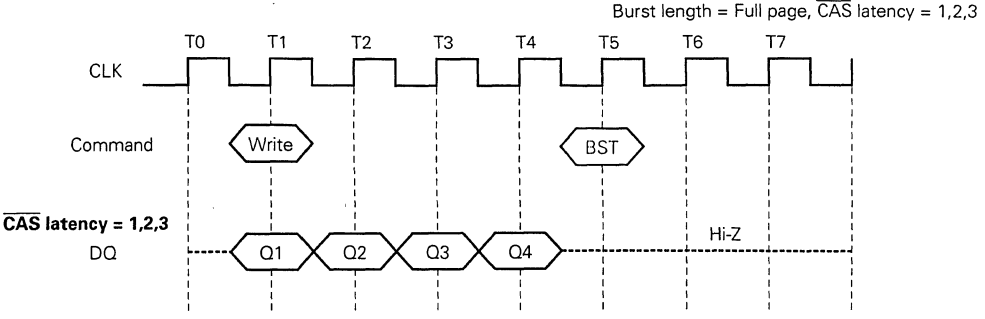
Burst Stop command is operated only in case full page burst mode. During the other burst mode, Burst Stop command is NOP.

During full page burst read cycle, when the burst stop command is asserted, the burst read data are terminated and the data bus goes to high-impedance after the $\overline{CAS}$ latency from the burst stop command.




**Remark** BST: Burst stop command

During full page burst write cycle, when the burst stop command is asserted, the burst read data are terminated and data bus goes to high-impedance at the same clock with the burst stop command.



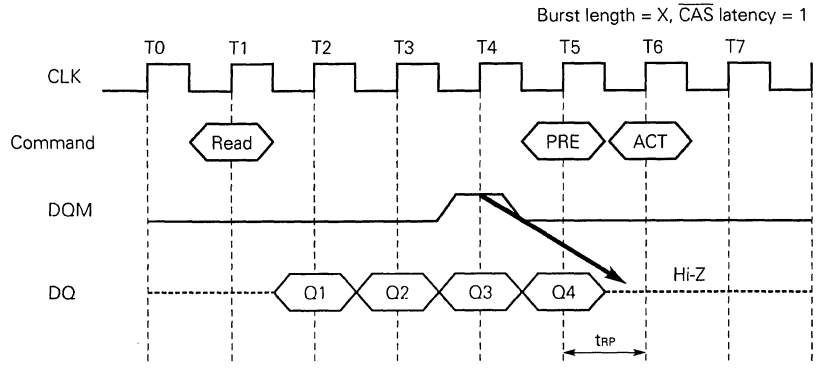

**Remark** BST: Burst stop command

## 15.2 Precharge Termination

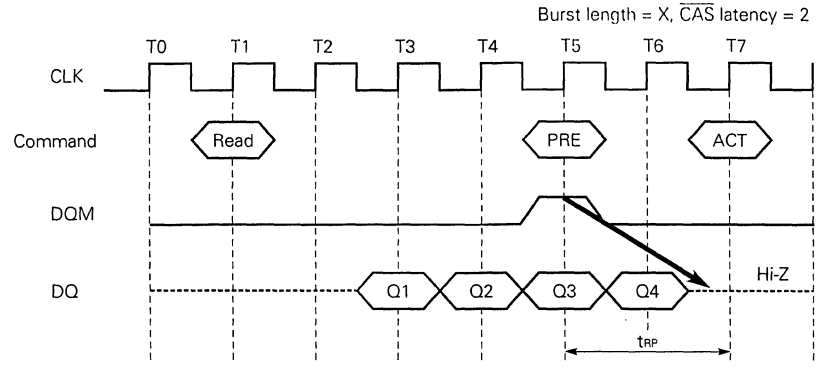
### 15.2.1 Precharge Termination in READ Cycle

During READ cycle, the burst read operation is terminated by a precharge command.
When the precharge command is asserted, the burst read operation is terminated and precharge starts.
The same bank can be activated again after $t_{RP}$ from the precharge command.
The DQM must be high to mask the invalid data.

When $\overline{CAS}$ latency is 1, the read data will remain valid until the precharge command is asserted.
Invalid data may appear one clock after valid data out.
The DQM may be high to mask the invalid data.

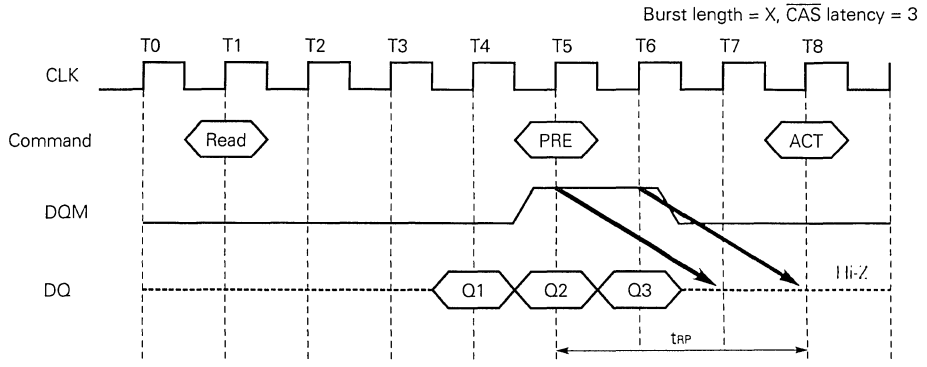



When $\overline{CAS}$ latency is 2, the read data will remain valid until one clock after the precharge command. Invalid data may appear one clock after valid data out.
The DQM may be high to mask the invalid data.



When $\overline{CAS}$ latency is 3, the read data will remain valid until one clock after the precharge command. Invalid data may appear one and two clocks after valid data out.

The DQM may be high to mask the invalid data.




### 15.2.2 Precharge Termination in WRITE Cycle

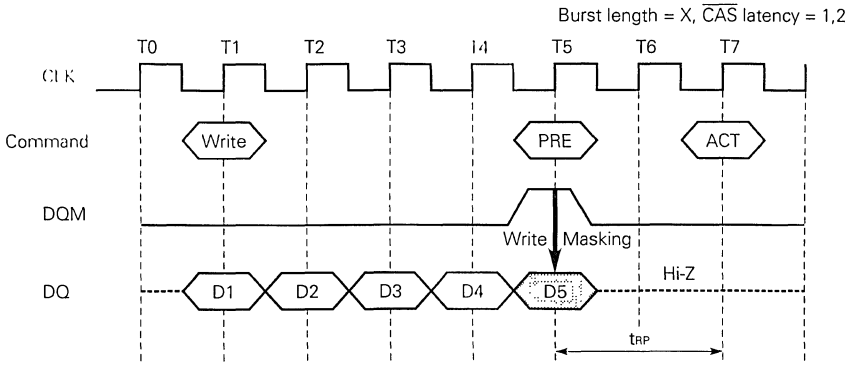
During WRITE cycle, the burst write operation is terminated by a precharge command.

When the precharge command is asserted, the burst write operation is terminated and precharge starts.

The same bank can be activated again after $t_{RP}$ from the precharge command.

The DQM must be high to mask invalid data in.

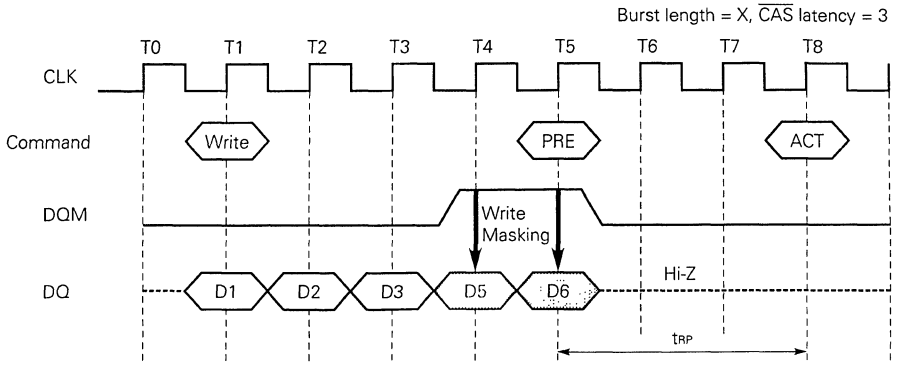
When $\overline{CAS}$ latency is 1 or 2, the write data written prior to the precharge command will be correctly stored. However, invalid data may be written at the same clock as the precharge command. To prevent this from happening, DQM must be high at the same clock as the precharge command. This will mask the invalid data.



When $\overline{CAS}$ latency is 3, the write data written more than one clock prior to the precharge command will be correctly stored.

However, invalid data may be written at one clock before and the same clock as the precharge command.

To prevent this from happening, DQM must be high from one clock prior to the precharge command until the precharge command. This will mask the invalid data.



## 16. Electrical Specifications (Preliminary)

- All voltage are referenced to $V_{SS}$ (GND).
- After power up, wait more than 100 μs and then, execute **Power on sequence and Auto Refresh** before proper device operation is achieved.

### Absolute Maximum Ratings

| Parameter | Symbol | Condition | Rating | Unit |
|---|---|---|---|---|
| Voltage on power supply pin relative to GND | $V_T$ | | −1.0 to +4.6 | V |
| Voltage on input pin relative to GND | $V_{CC}$, $V_{CC}Q$ | | −1.0 to +4.6 | V |
| Short circuit output current | $I_O$ | | 50 | mA |
| Power dissipation | $P_D$ | | 1 | W |
| Operating ambient temperature | $T_A$ | | 0 to 70 | °C |
| Storage temperature | $T_{stg}$ | | −55 to +125 | °C |

**Caution Exposing the device to stress above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational section of this specification. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.**

### Recommended Operating Conditions

| Parameter | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Supply voltage | $V_{CC}$ | | 3.0 | 3.3 | 3.6 | V |
| High level input voltage | $V_{IH}$ | | 2.0 | | $V_{CC}$ + 0.3 | V |
| Low level input voltage | $V_{IL}$ | | −0.3 | | +0.8 | V |
| Operating ambient temperature | $T_A$ | | 0 | | 70 | °C |

### Capacitance ($T_A$=25°C, f=1MHz)

| Parameter | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input capacitance | $C_{I1}$ | A0 to A9 | 2 | | 4 | pF |
| | $C_{I2}$ | CLK, CKE, $\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, DSF, DQM | 2 | | 4 | pF |
| Data input/output capacitance | $C_{I/O}$ | DQ0 to DQ31 | 2 | | 5 | pF |

**DC Characteristics 1 (Recommended Operating Conditions unless otherwise noted)**

| Parameter | Symbol | Test condition | | Grade | MAX. | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Operating current | $I_{CC1}$ | Burst length=1<br>$t_{RAS} \geq t_{RAS(MIN.)}$<br>$t_{RP} \geq t_{RP(MIN.)}$<br>$I_O$=0mA | | -10 | 105 | mA | 1 |
| | | | | -12 | 90 | | |
| | | | | -15 | 85 | | |
| Precharge standby current in Power down mode | $I_{CC2P}$ | CKE ≤ $V_{IL(MAX.)}$ $t_{CK}$=15ns | | | 7 | mA | |
| | $I_{CC2PS}$ | CKE ≤ $V_{IL(MAX.)}$ $t_{CK}$=∞ | | | 6 | | |
| Precharge standby current in Non power down mode | $I_{CC2N}$ | CKE ≥ $V_{IH(MIN.)}$ $t_{CK}$=15ns<br>$\overline{CS}$ ≥ $V_{IH(MIN.)}$<br>Input signals are changed one time during 30ns. | | | 36 | mA | |
| | $I_{CC2NS}$ | CKE ≥ $V_{IH(MIN.)}$ $t_{CK}$=∞<br>Input signals are stable. | | | 22 | | |
| Active standby current in Power down mode | $I_{CC3P}$ | CKE ≤ $V_{IL(MAX.)}$ $t_{CK}$=15ns | | | 7 | mA | |
| | $I_{CC3PS}$ | CKE ≤ $V_{IL(MAX.)}$ $t_{CK}$=∞ | | | 6 | | |
| Active standby current in Non power down mode | $I_{CC3N}$ | CKE ≥ $V_{IH(MIN.)}$ $t_{CK}$=15ns<br>$\overline{CS}$ ≥ $V_{IH(MIN.)}$<br>Input signals are changed one time during 30 ns. | | | 36 | mA | |
| | $I_{CC3NS}$ | CKE ≥ $V_{IH(MIN.)}$ $t_{CK}$=∞<br>Input signals are stable. | | | 22 | | |
| Operating current (Burst mode) | $I_{CC4}$ | $t_{CK} \geq t_{CK(MIN.)}$<br>$I_O$=0mA | $\overline{CAS}$ latency = 1 | -10 | 210 | mA | 2 |
| | | | | -12 | 180 | | |
| | | | | -15 | 165 | | |
| | | | $\overline{CAS}$ latency = 2 | -10 | 280 | | |
| | | | | -12 | 235 | | |
| | | | | -15 | 220 | | |
| | | | $\overline{CAS}$ latency = 3 | -10 | 365 | | |
| | | | | -12 | 310 | | |
| | | | | -15 | 285 | | |
| Refresh current | $I_{CC5}$ | $t_{RC} \geq t_{RC(MIN.)}$ | | -10 | 85 | mA | 3 |
| | | | | -12 | 80 | | |
| | | | | -15 | 75 | | |
| Self refresh Current | $I_{CC6}$ | CKE ≤ 0.2V | | | 6 | mA | |
| Operating Current (Block Write Mode) | $I_{CC7}$ | $t_{CK} \geq t_{CK(MIN.)}$, $I_O$ = 0 mA,<br>$\overline{CAS}$ cycle = 20 ns | | | 250 | mA | |

**Notes 1.** $I_{CC1}$ depends on output loading and cycle rates. Specified values are obtained with the output open. In addition to this, $I_{CC1}$ is measured on condition that addresses are changed only one time during $t_{CK(MIN.)}$.

**2.** $I_{CC4}$ depends on output loading and cycle rates. Specified values are obtained with the output open. In addition to this, $I_{CC4}$ is measured on condition that addresses are changed only one time during $t_{CK(MIN.)}$.

**3.** $I_{CC5}$ is measured on condition that addresses are changed only one time during $t_{CK(MIN.)}$.

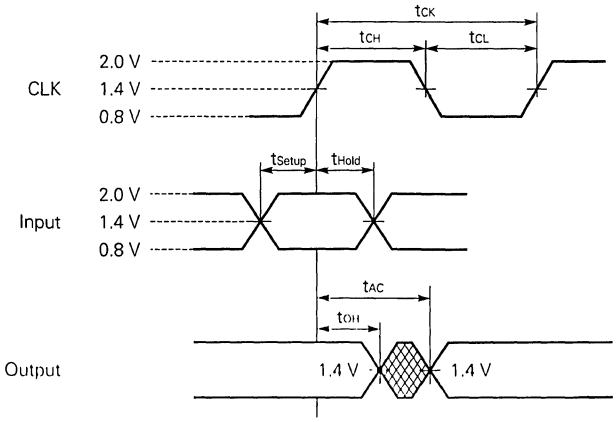
**DC Characteristics 2 (Recommended Operating Conditions unless otherwise noted)**

| Parameter | Symbol | Test condition | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input leakage current | $I_{I(L)}$ | $V_I$=0 to 3.6V, all other pins not under test =0V | −1.0 | | +1.0 | μA |
| Output leakage current | $I_{O(L)}$ | $D_{OUT}$ is disabled, $V_O$=0 to 3.6V | −1.0 | | +1.0 | μA |
| High level output voltage | $V_{OH}$ | $I_O$=−2mA | 2.4 | | | V |
| Low level output voltage | $V_{OL}$ | $I_O$=+2mA | | | 0.4 | V |

## AC Characteristics (Recommended Operating Conditions unless otherwise noted)

### Test Conditions

- AC measurements assume $t_T$=1ns.
- Reference level for measuring timing of input signals is 1.4V. Transition times are measured between $V_{IH}$ and $V_{IL}$.
- If $t_T$ is longer than 1 ns, reference level for measuring timing of input signals is $V_{IH\,(MIN.)}$ and $V_{IL\,(MAX.)}$.
- An access time is measured at 1.4V.




### Synchronous Characteristics (1/2)

| Parameter | | Symbol | -10 | | -12 | | -15 | | Unit | Note |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Clock cycle time | $\overline{CAS}$ latency=3 | $t_{CK3}$ | 10 | (100MHz) | 12 | (83MHz) | 15 | (66MHz) | ns | |
| | $\overline{CAS}$ latency=2 | $t_{CK2}$ | 15 | (66MHz) | 18 | (55MHz) | 19.5 | (50MHz) | ns | |
| | $\overline{CAS}$ latency=1 | $t_{CK1}$ | 30 | (33MHz) | 36 | (28MHz) | 39 | (25MHz) | ns | |
| Access time from CLK | $\overline{CAS}$ latency=3 | $t_{AC3}$ | | 9 | | 11 | | 14 | ns | 1 |
| | $\overline{CAS}$ latency=2 | $t_{AC2}$ | | 12 | | 15 | | 16.5 | ns | 1 |
| | $\overline{CAS}$ latency=1 | $t_{AC1}$ | | 27 | | 33 | | 36 | ns | 1 |
| CLK high level width | | $t_{CH}$ | 3.5 | | 4 | | 5 | | ns | |
| CLK low level width | | $t_{CL}$ | 3.5 | | 4 | | 5 | | ns | |
| Data-out hold time | | $t_{OH}$ | 4 | | 4 | | 4 | | ns | |
| Data-out low-impedance time | | $t_{LZ}$ | 0 | | 0 | | 0 | | ns | |
| Data-out high-impedance time | $\overline{CAS}$ latency = 3 | $t_{HZ3}$ | 4 | 8 | 4 | 8 | 4 | 10 | ns | |
| | $\overline{CAS}$ latency = 2 | $t_{HZ2}$ | 4 | 11 | 4 | 11 | 4 | 11 | ns | |
| | $\overline{CAS}$ latency = 1 | $t_{HZ1}$ | 4 | 27 | 4 | 27 | 4 | 27 | ns | |
| Data-in setup time | | $t_{DS}$ | 3 | | 3.5 | | 3.5 | | ns | |
| Data-in hold time | | $t_{DH}$ | 1 | | 1.5 | | 1.5 | | ns | |
| Address setup time | | $t_{AS}$ | 3 | | 3.5 | | 3.5 | | ns | |
| Address hold time | | $t_{AH}$ | 1 | | 1.5 | | 1.5 | | ns | |
| CKE setup time | | $t_{CKS}$ | 3 | | 3.5 | | 3.5 | | ns | |
| CKE hold time | | $t_{CKH}$ | 1 | | 1.5 | | 1.5 | | ns | |
| CKE setup time (Power down exit) | | $t_{CKSP}$ | 3 | | 3.5 | | 3.5 | | ns | |

**Note 1.** Loading capacitance is 30 pF.

**Synchronous Characteristics** (2/2)

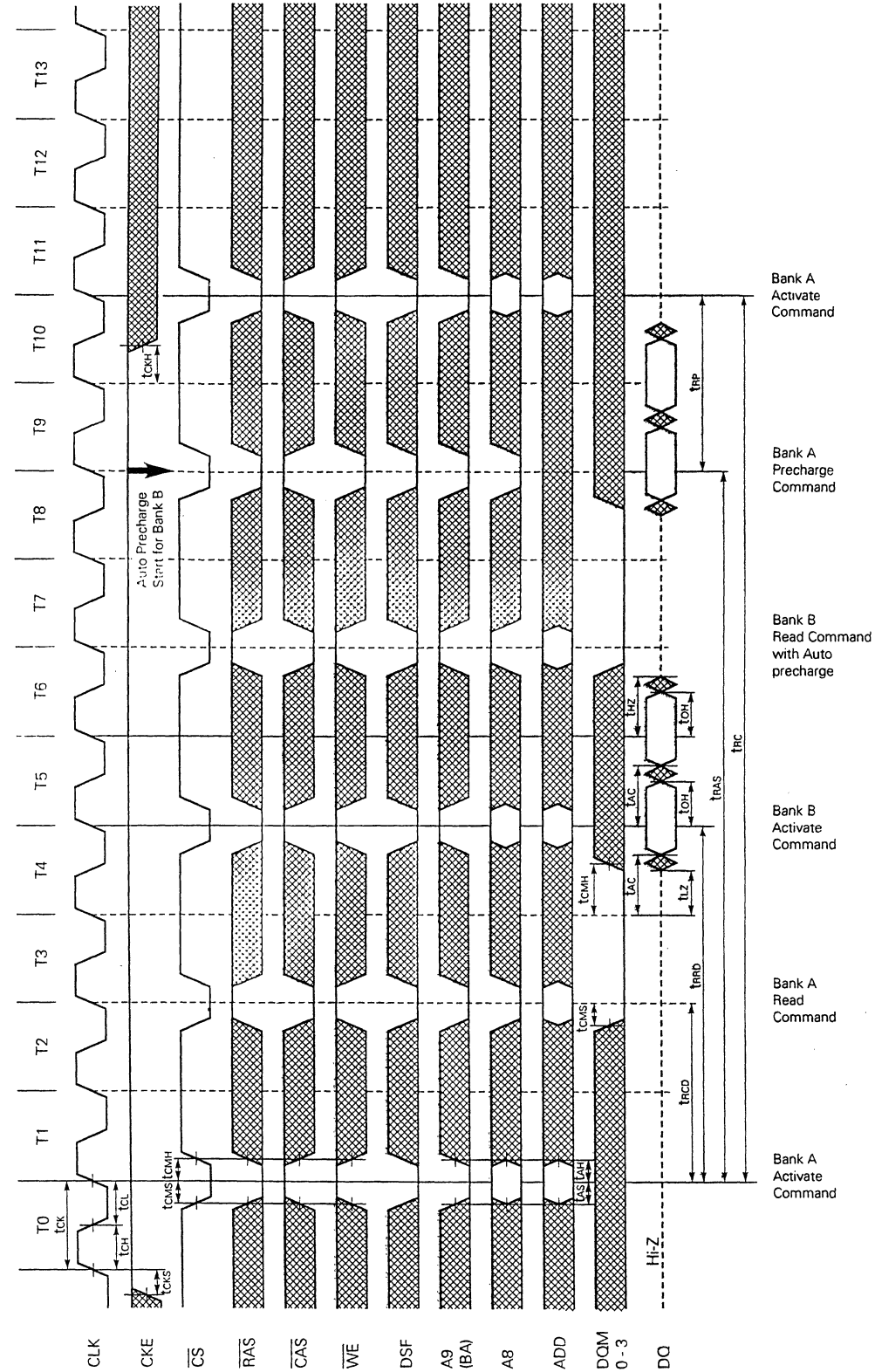
| Parameter | Symbol | -10 | | -12 | | -15 | | Unit | Note |
|---|---|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| Command ($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, DSF, DQM) setup time | $t_{CMS}$ | 3 | | 3.5 | | 3.5 | | ns | |
| Command ($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, DSF, DQM) hold time | $t_{CMH}$ | 1 | | 1.5 | | 1.5 | | ns | |

**Asynchronous Characteristics**

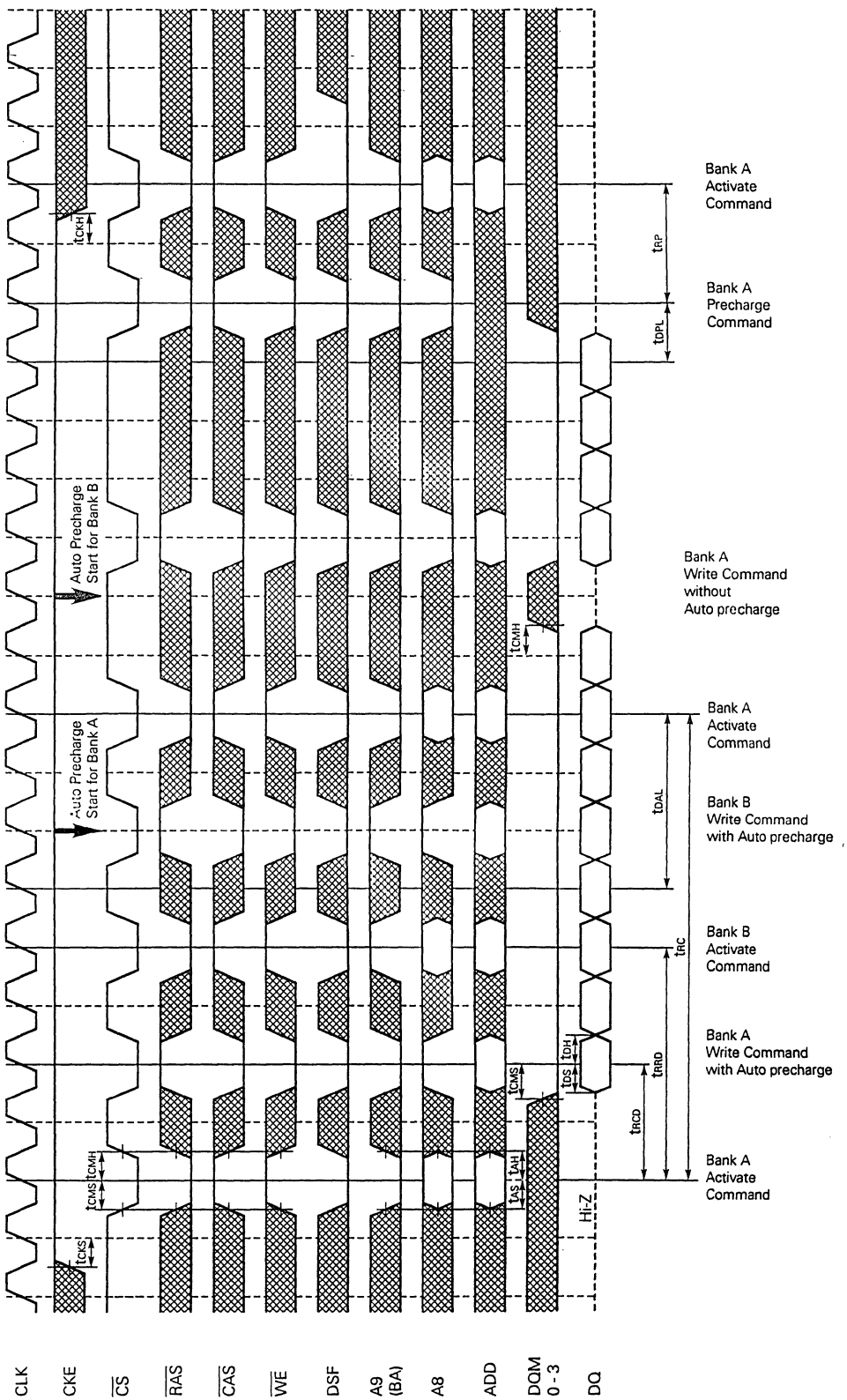
| Parameter | | Symbol | -10 | | -12 | | -15 | | Unit | Note |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | | |
| REF to REF/ACT Command period | | $t_{RC}$ | 100 | | 120 | | 130 | | ns | |
| ACT to PRE Command period | | $t_{RAS}$ | 70 | 120,000 | 84 | 120,000 | 90 | 120,000 | ns | |
| PRE to ACT Command period | | $t_{RP}$ | 30 | | 36 | | 39 | | ns | |
| Delay time ACT to READ/WRITE Command | | $t_{RCD}$ | 30 | | 36 | | 39 | | ns | |
| ACT(0) to ACT(1) Command period | | $t_{RRD}$ | 30 | | 36 | | 39 | | ns | |
| Data-in to PRE Command period | $\overline{CAS}$ latency=3 | $t_{DPL3}$ | 1CLK+10 | | 1CLK+12 | | 1CLK+15 | | ns | |
| | $\overline{CAS}$ latency=2 | $t_{DPL2}$ | 15 | | 18 | | 19.5 | | ns | |
| | $\overline{CAS}$ latency=1 | $t_{DPL1}$ | 15 | | 18 | | 19.5 | | ns | |
| Data-in to ACT (REF) Command period (Auto precharge) | $\overline{CAS}$ latency=3 | $t_{DAL3}$ | 2CLK+30 | | 2CLK+36 | | 2CLK+45 | | ns | |
| | $\overline{CAS}$ latency=2 | $t_{DAL2}$ | 1CLK+30 | | 1CLK+36 | | 1CLK+39 | | ns | |
| | $\overline{CAS}$ latency=1 | $t_{DAL1}$ | 1CLK+30 | | 1CLK+36 | | 1CLK+39 | | ns | |
| Block write cycle time | | $t_{BWC}$ | 20 | | 24 | | 30 | | ns | |
| Block write data-in to PRE Command period | $\overline{CAS}$ latency=3 | $t_{BPL3}$ | 1CLK+20 | | 1CLK+24 | | 1CLK+30 | | ns | |
| | $\overline{CAS}$ latency=2 | $t_{BPL2}$ | 30 | | 36 | | 39 | | ns | |
| | $\overline{CAS}$ latency=1 | $t_{BPL1}$ | 30 | | 36 | | 36 | | ns | |
| Block write data-in Active (REF) Command Period (Auto Precharge) | $\overline{CAS}$ latency=3 | $t_{BAL3}$ | 2CLK+40 | | 2CLK+48 | | 2CLK+60 | | ns | |
| | $\overline{CAS}$ latency=2 | $t_{BAL2}$ | 1CLK+40 | | 1CLK+48 | | 1CLK+52 | | ns | |
| | $\overline{CAS}$ latency=1 | $t_{BAL1}$ | 1CLK+40 | | 1CLK+48 | | 1CLK+52 | | ns | |
| Mode register set cycle time | | $t_{RSC}$ | 20 | | 20 | | 20 | | ns | |
| Transition time | | $t_T$ | 1 | 30 | 1 | 30 | 1 | 30 | ns | |
| Refresh time | | $t_{REF}$ | | 16 | | 16 | | 16 | ms | |

## 16.1 AC Parameters for Read/Write Cycles

**AC Parameters for Read Timing** (Burst length = 2, $\overline{CAS}$ latency = 2)



**AC Parameters for Write Timing** (Burst length = 4, $\overline{CAS}$ latency = 2)

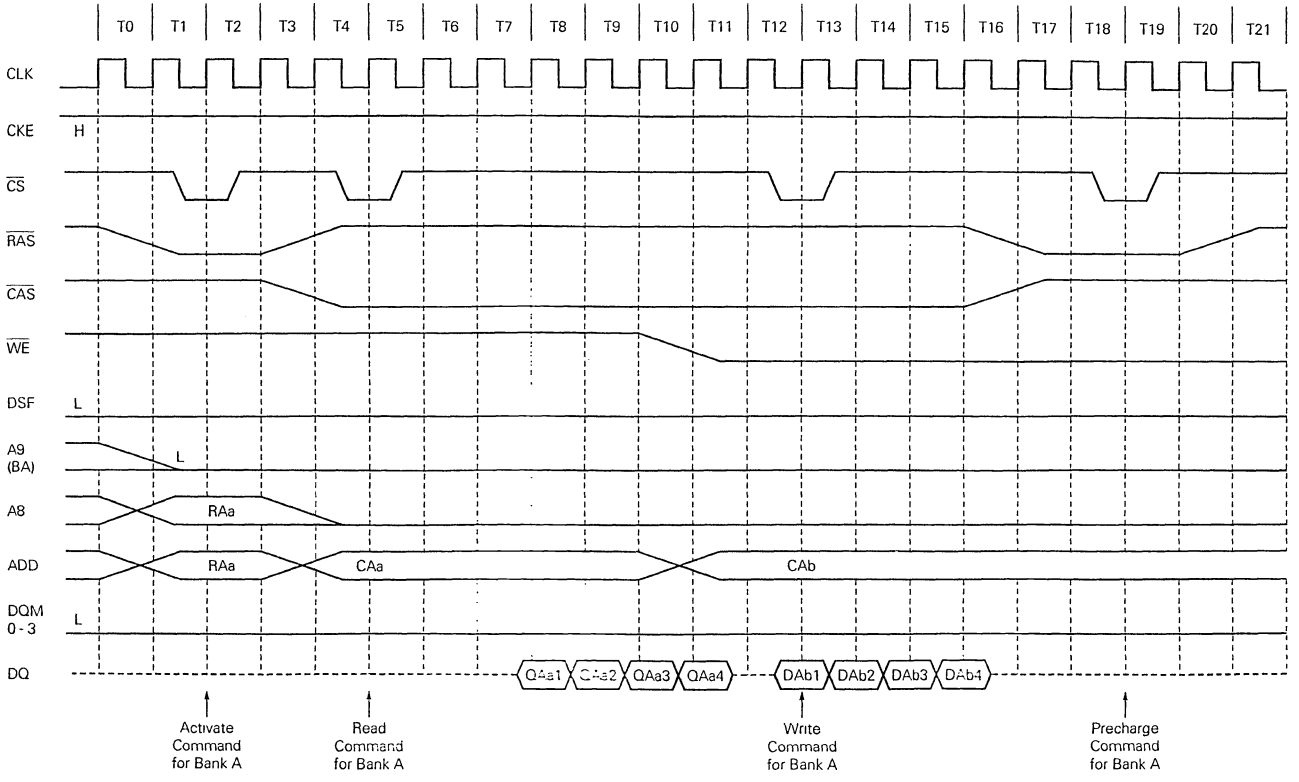


### 16.2 Relationship between Frequency and Latency

| Speed version | -10 | | | -12 | | | -15 | | |
|---|---|---|---|---|---|---|---|---|---|
| Clock cycle time [ns] | 10 | 15 | 30 | 12 | 18 | 36 | 15 | 19.5 | 39 |
| Frequency [MHz] | 100 | 66 | 33 | 83 | 55 | 28 | 66 | 50 | 25 |
| $\overline{CAS}$ latency | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 |
| [$t_{RCD}$] | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 |
| $\overline{RAS}$ latency ($\overline{CAS}$ latency + [$t_{RCD}$]) | 6 | 4 | 2 | 6 | 4 | 2 | 6 | 4 | 2 |
| [$t_{RC}$] | 10 | 7 | 4 | 10 | 7 | 4 | 10 | 7 | 4 |
| [$t_{RAS}$] | 7 | 5 | 3 | 7 | 5 | 3 | 7 | 5 | 3 |
| [$t_{RRD}$] | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 |
| [$t_{RP}$] | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 |
| [$t_{DPL}$] | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 |
| [$t_{DAL}$] | 5 | 3 | 2 | 5 | 3 | 2 | 5 | 3 | 2 |

### 16.3 $\overline{CS}$ Function

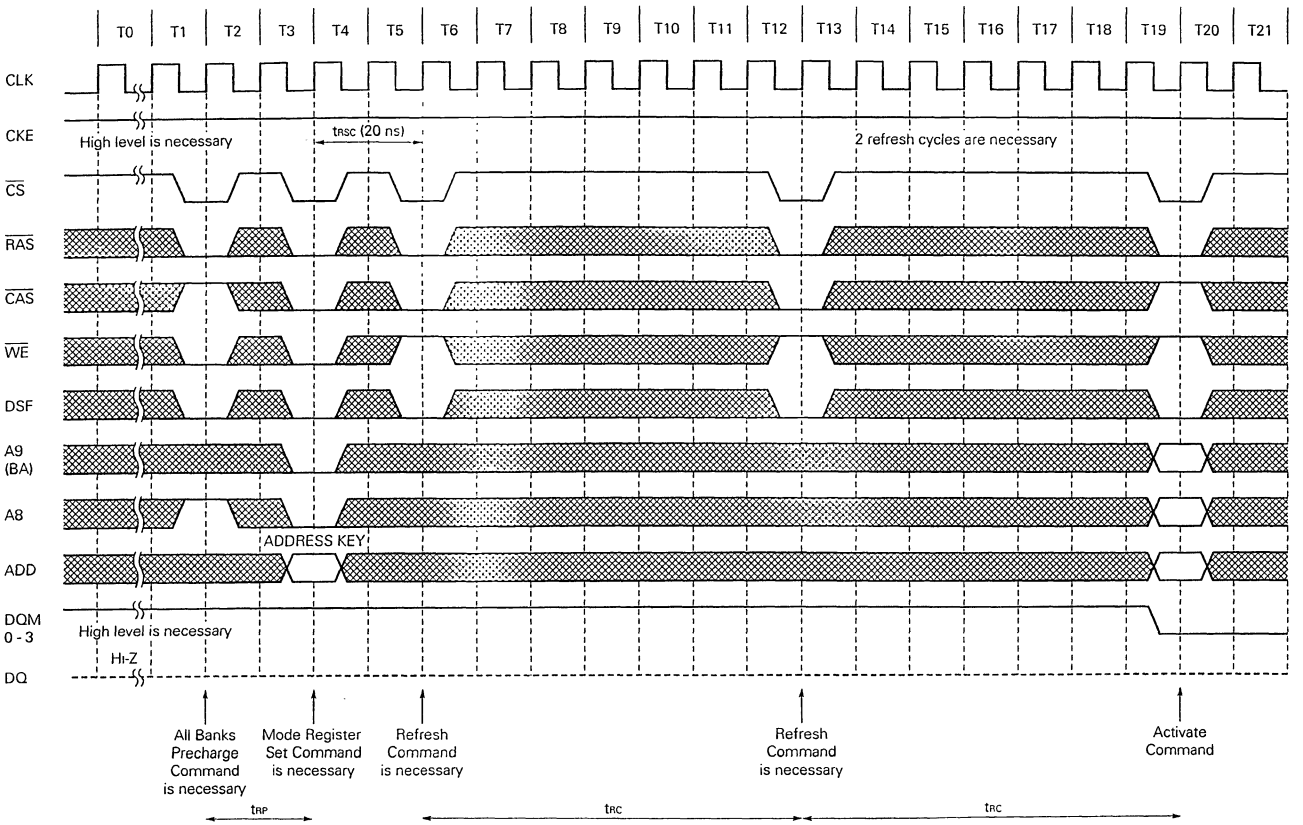
**$\overline{CS}$ Function (Only $\overline{CS}$ signal needs to be asserted at minimum rate)** (at 100 MHz Burst length = 4, $\overline{CAS}$ latency = 3)



## 16.4 Basic Cycles

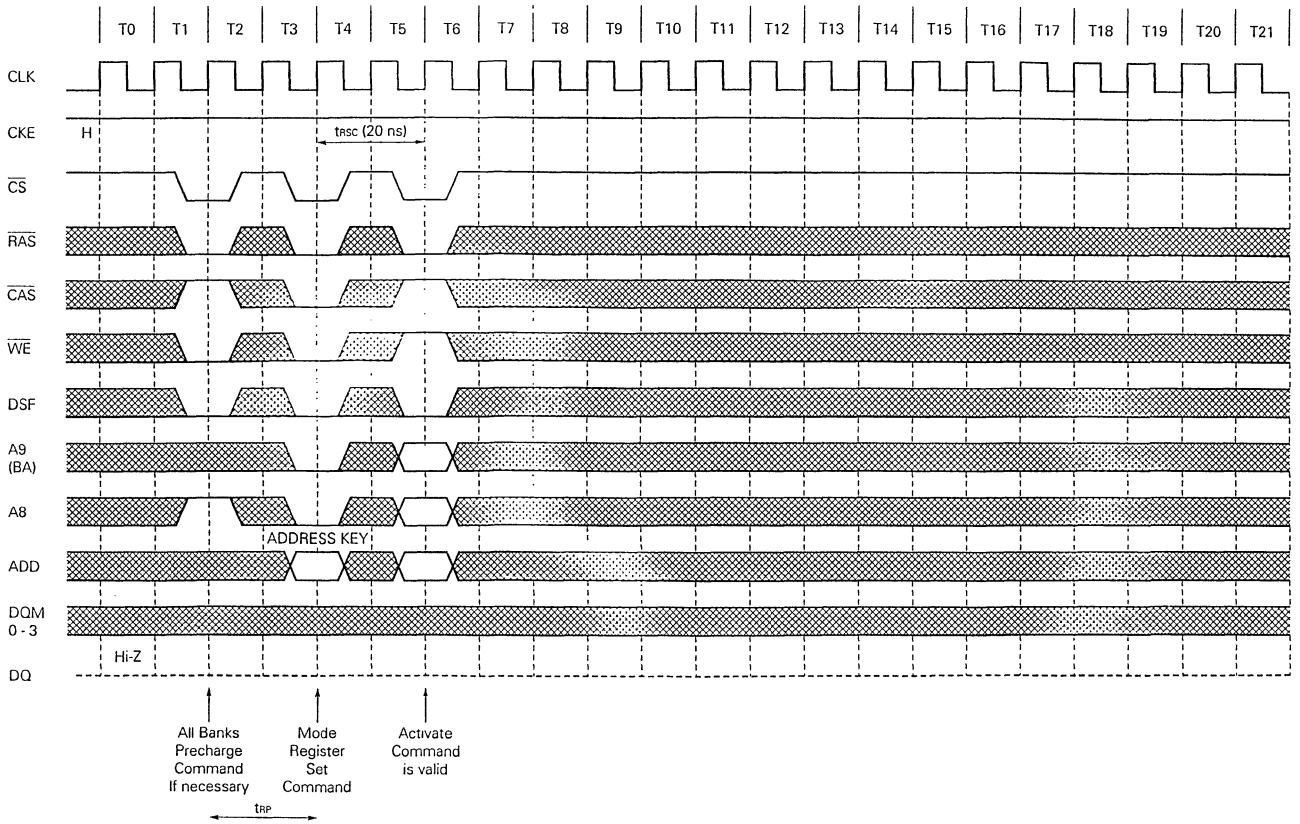
### 16.4.1 Initialization

**Power on Sequence and Auto Refresh**



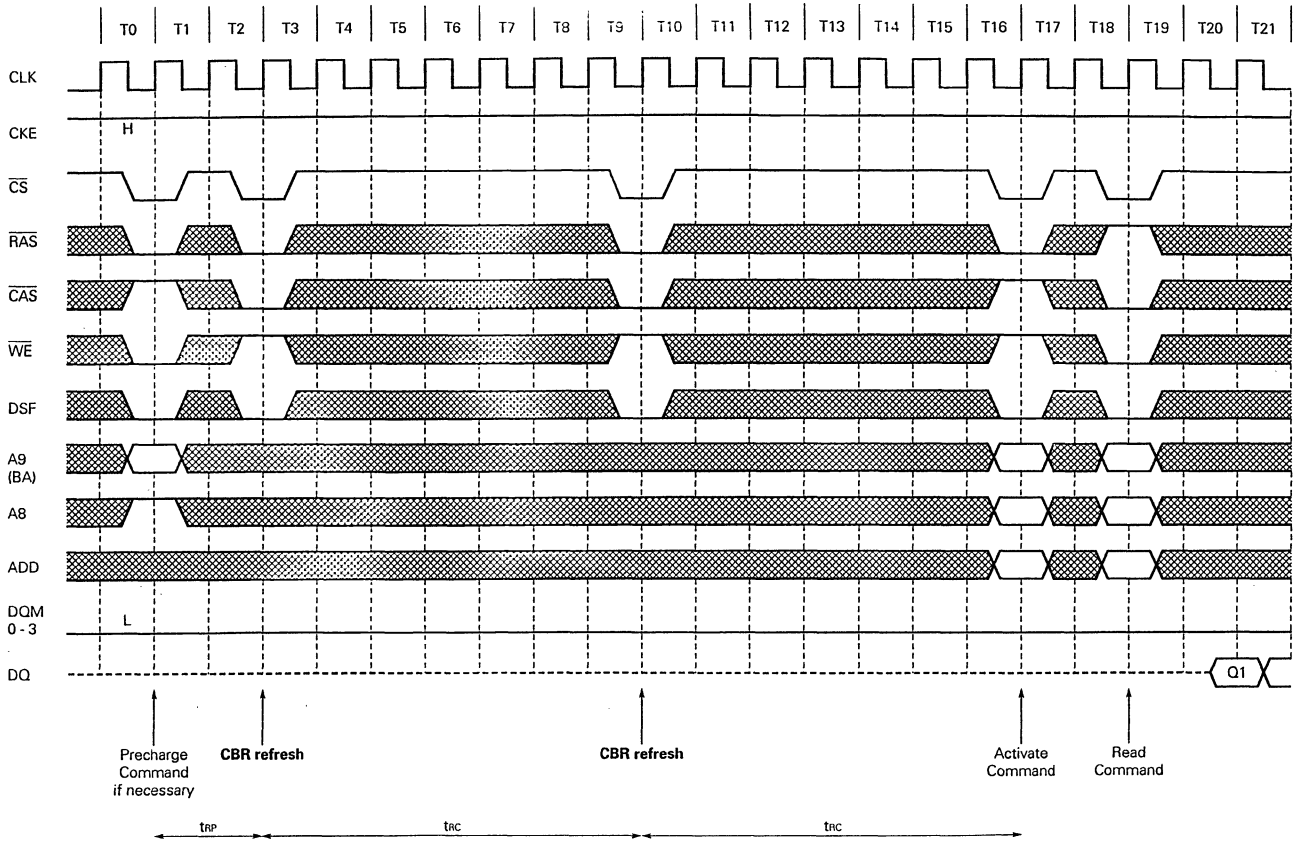
### 16.4.2 Mode Register Set

**Mode Register** (Burst length = 4, $\overline{CAS}$ latency = 2)

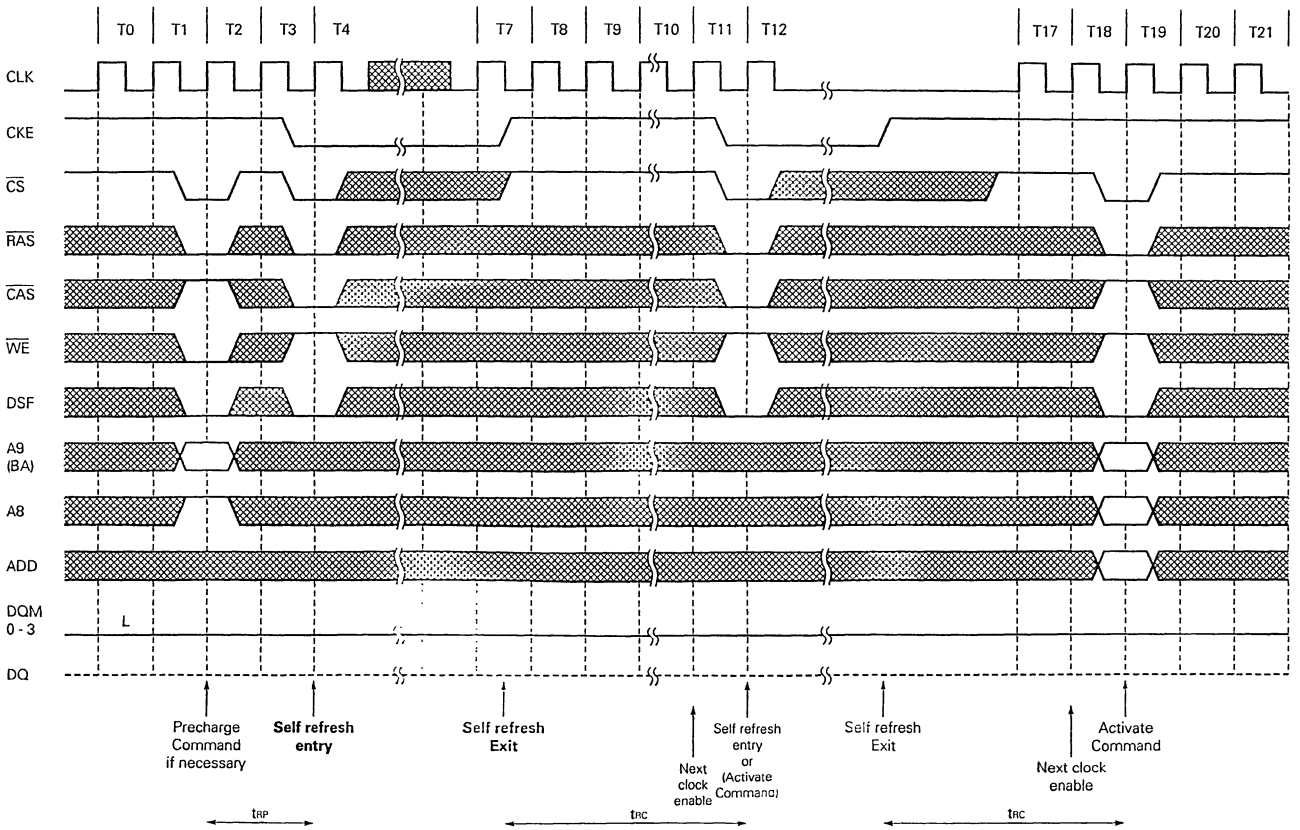


### 16.4.3 Refresh Cycle

**CBR Refresh** ($\overline{CAS}$ latency = 2)

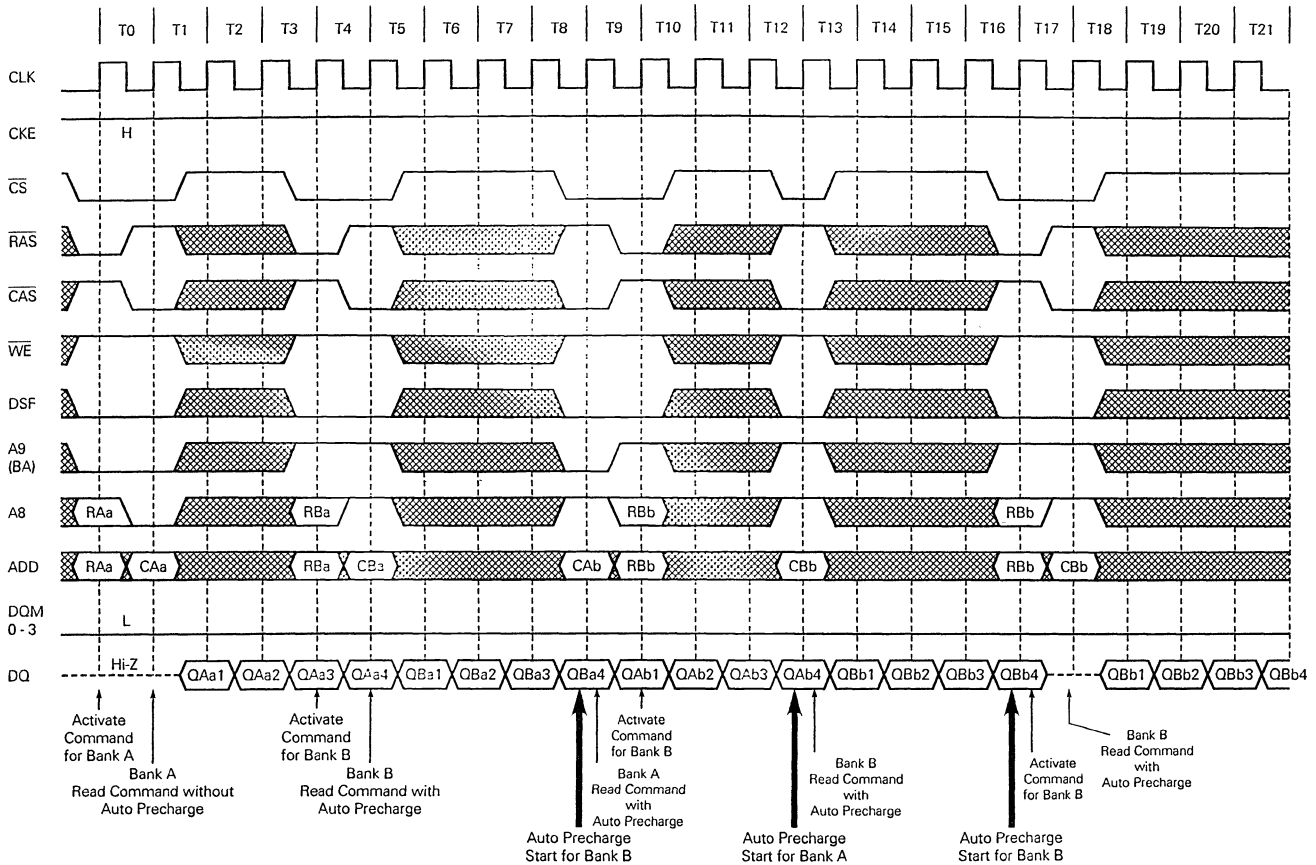


**Self Refresh (entry and exit)**

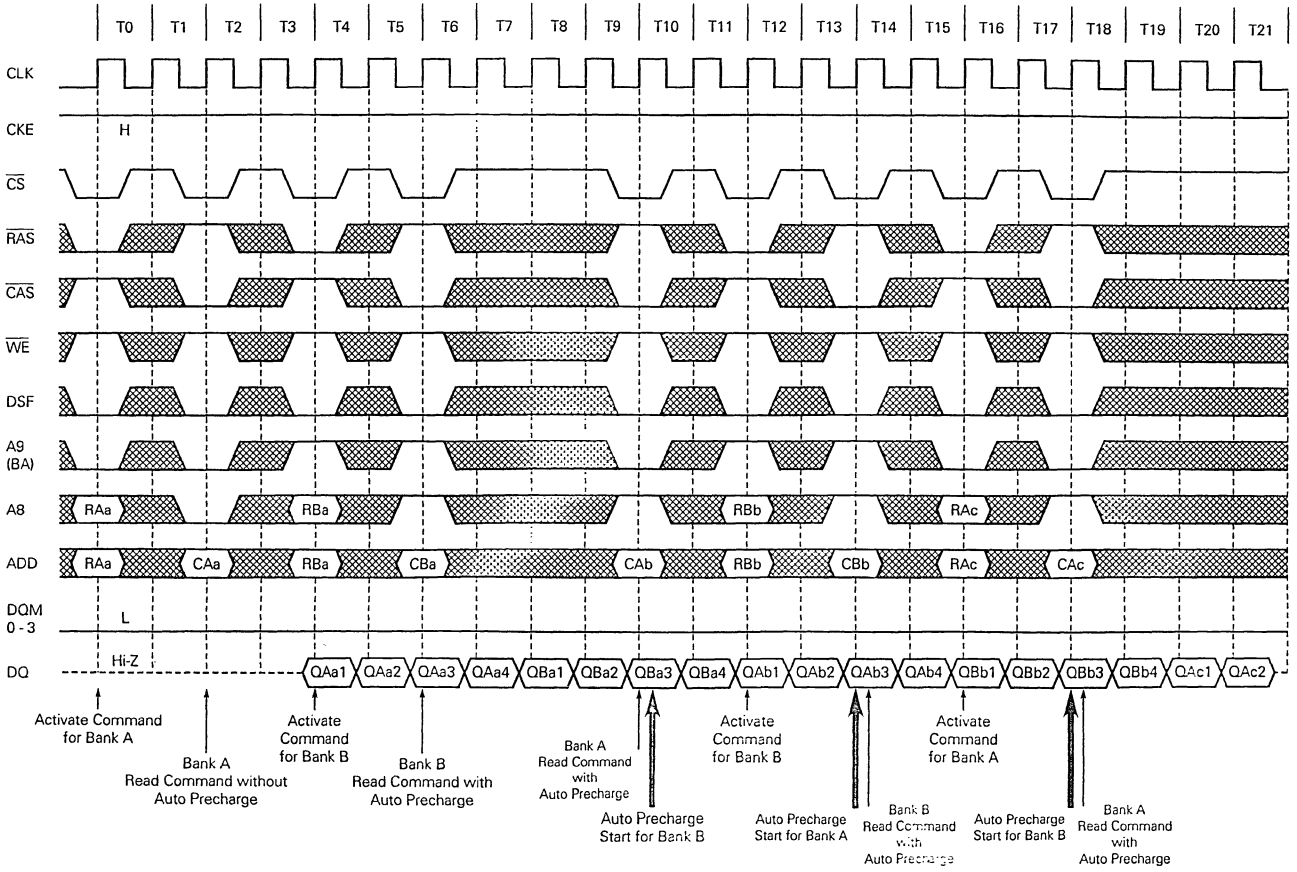


### 16.4.4 Cycle with Auto Precharge

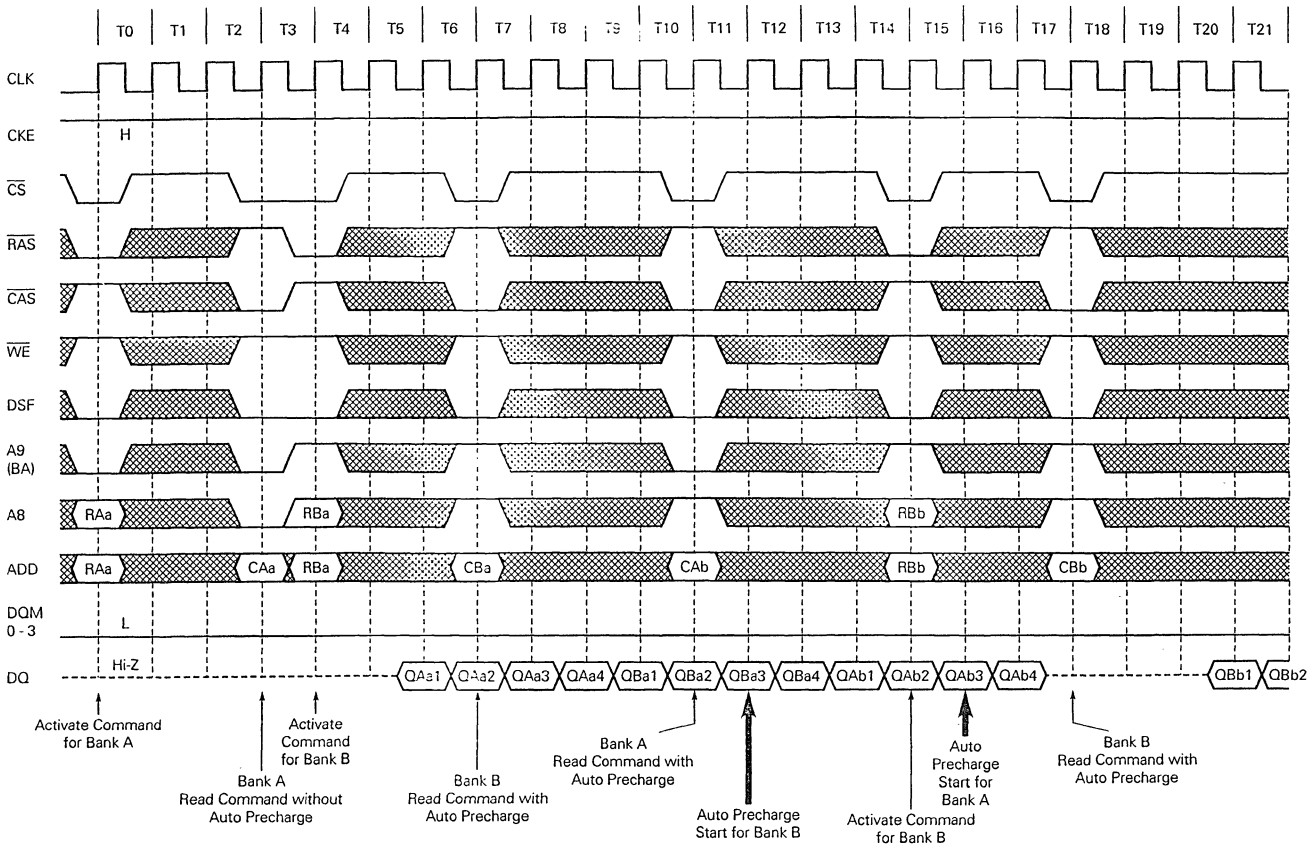
**Auto Precharge after Read Burst (1/3)** (Burst length = 4, $\overline{CAS}$ latency = 1)



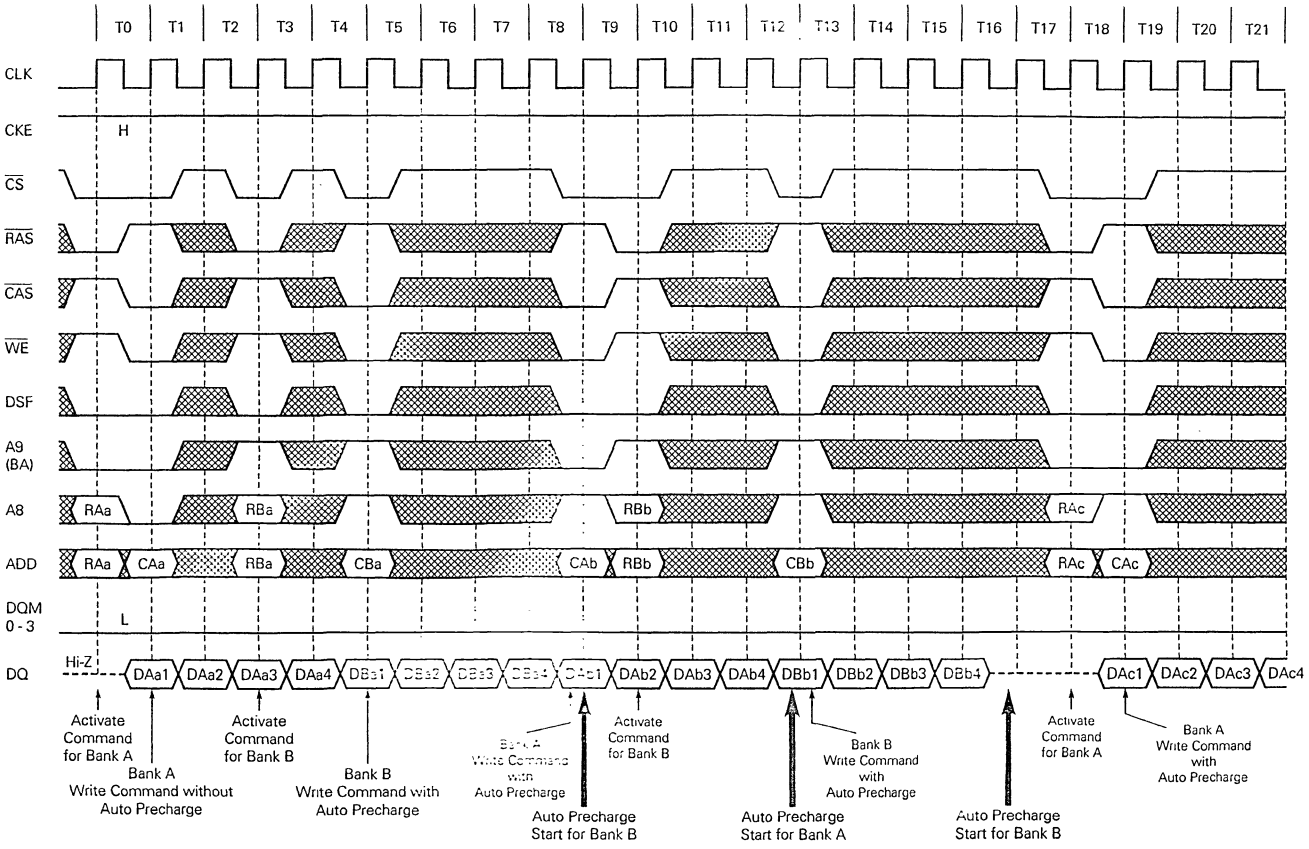
**Auto Precharge after Read Burst (2/3)** (Burst length = 4, $\overline{CAS}$ latency = 2)



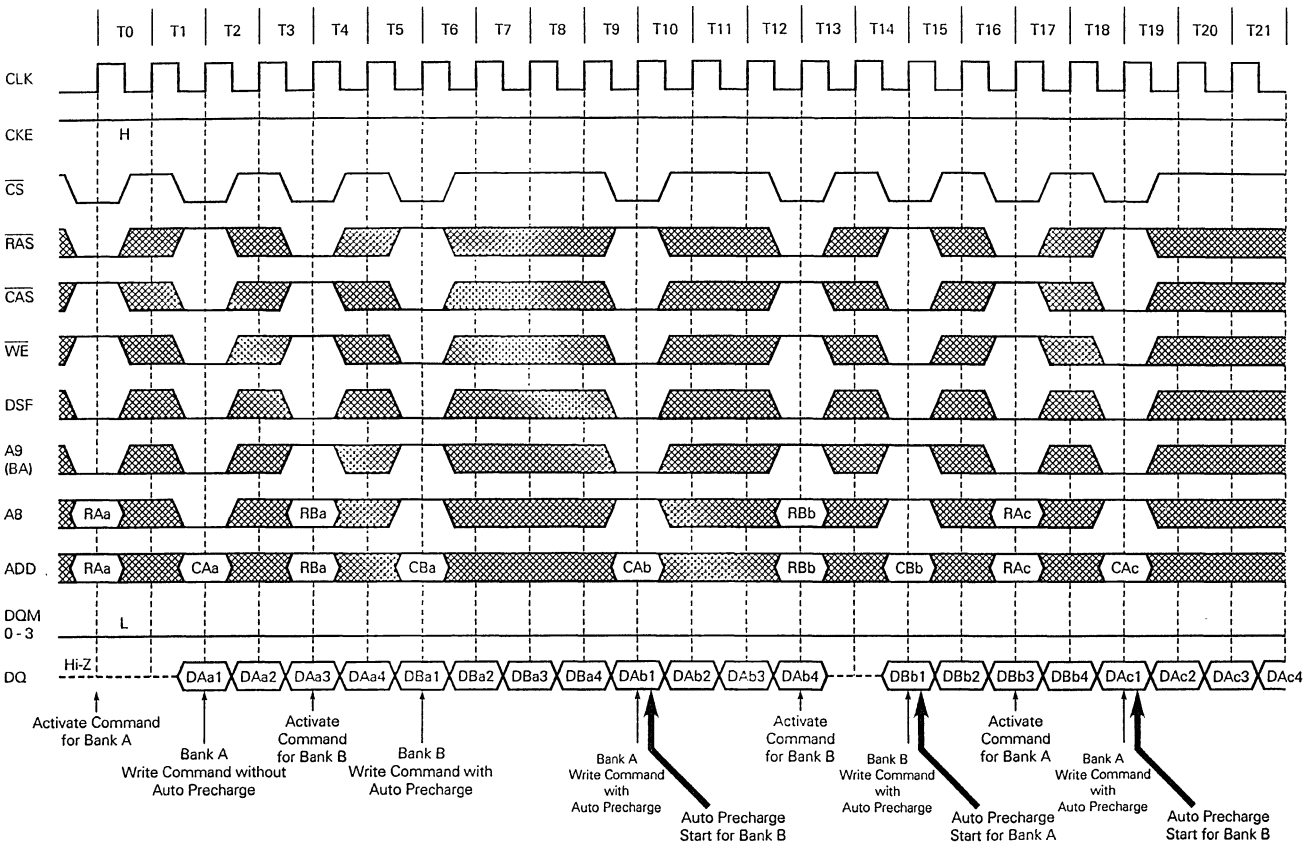
**Auto Precharge after Read Burst (3/3)** (Burst length = 4, CAS latency = 3)



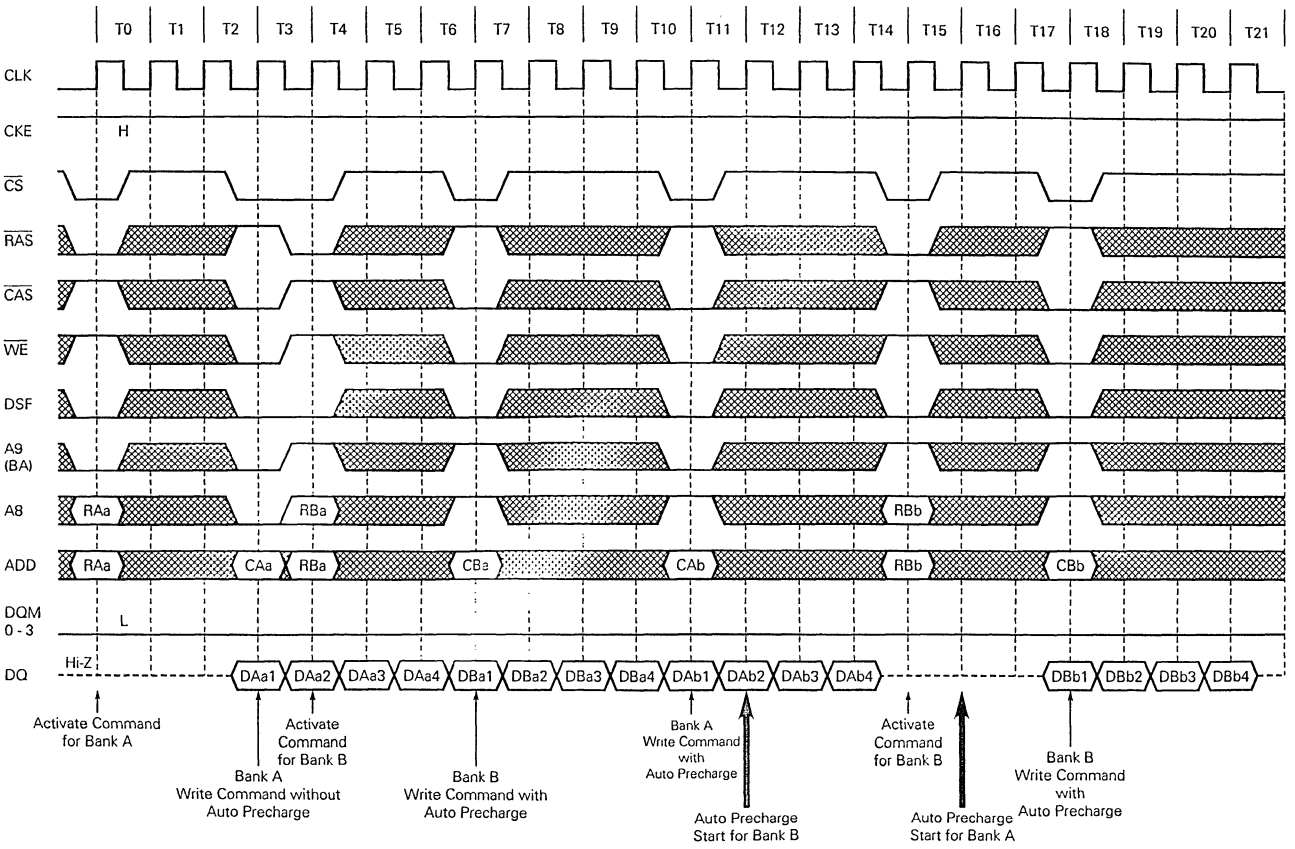
**Auto Precharge after Write Burst (1/3)** (Burst length = 4, $\overline{CAS}$ latency = 1)



**Auto Precharge after Write Burst (2/3)** (Burst length = 4, $\overline{CAS}$ latency = 2)

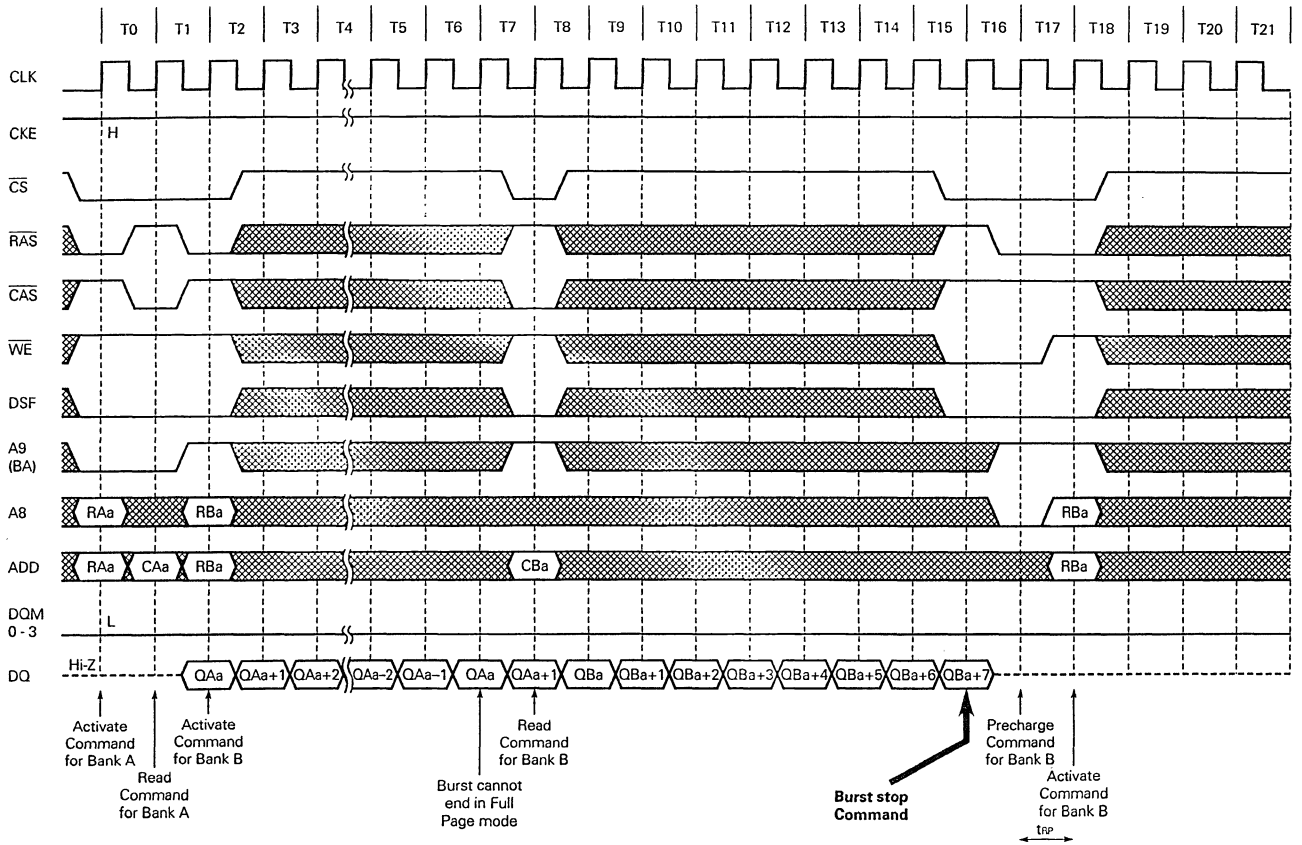


**Auto Precharge after Write Burst (3/3)** (Burst length = 4, $\overline{CAS}$ latency = 3)

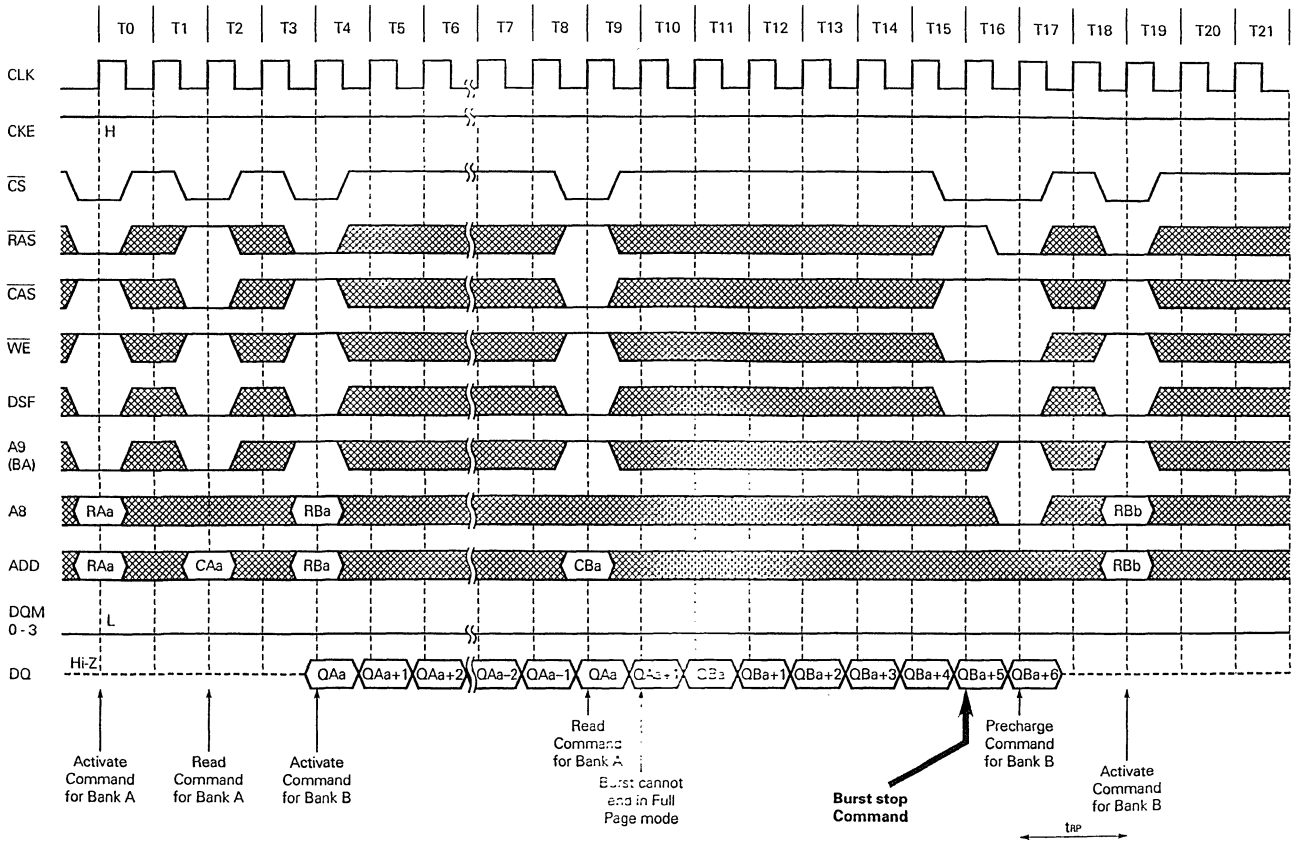


### 16.4.5 Full Page Mode Cycle

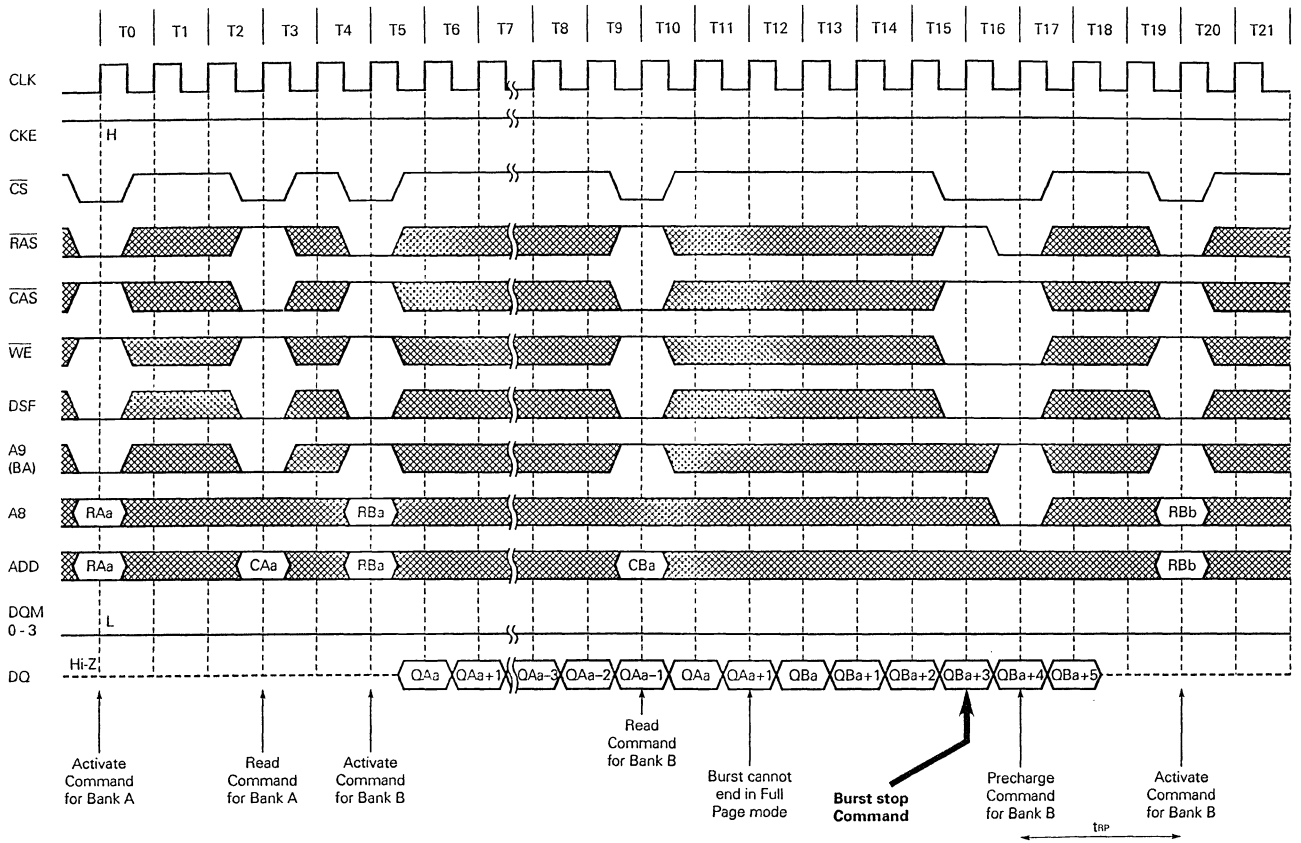
**Full Page READ Cycle (1/3)** ($\overline{CAS}$ latency = 1)



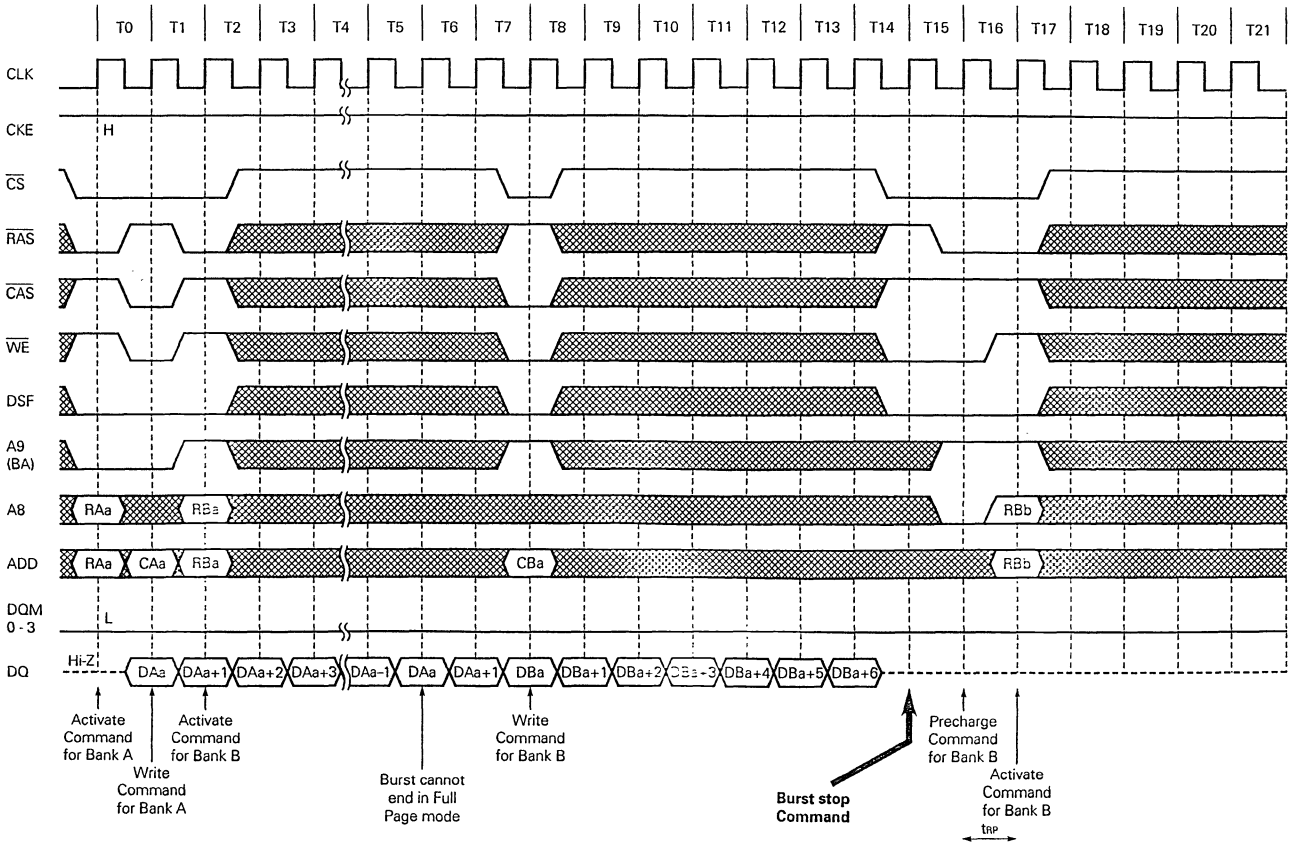
**Full Page READ Cycle (2/3) ($\overline{CAS}$ latency = 2)**



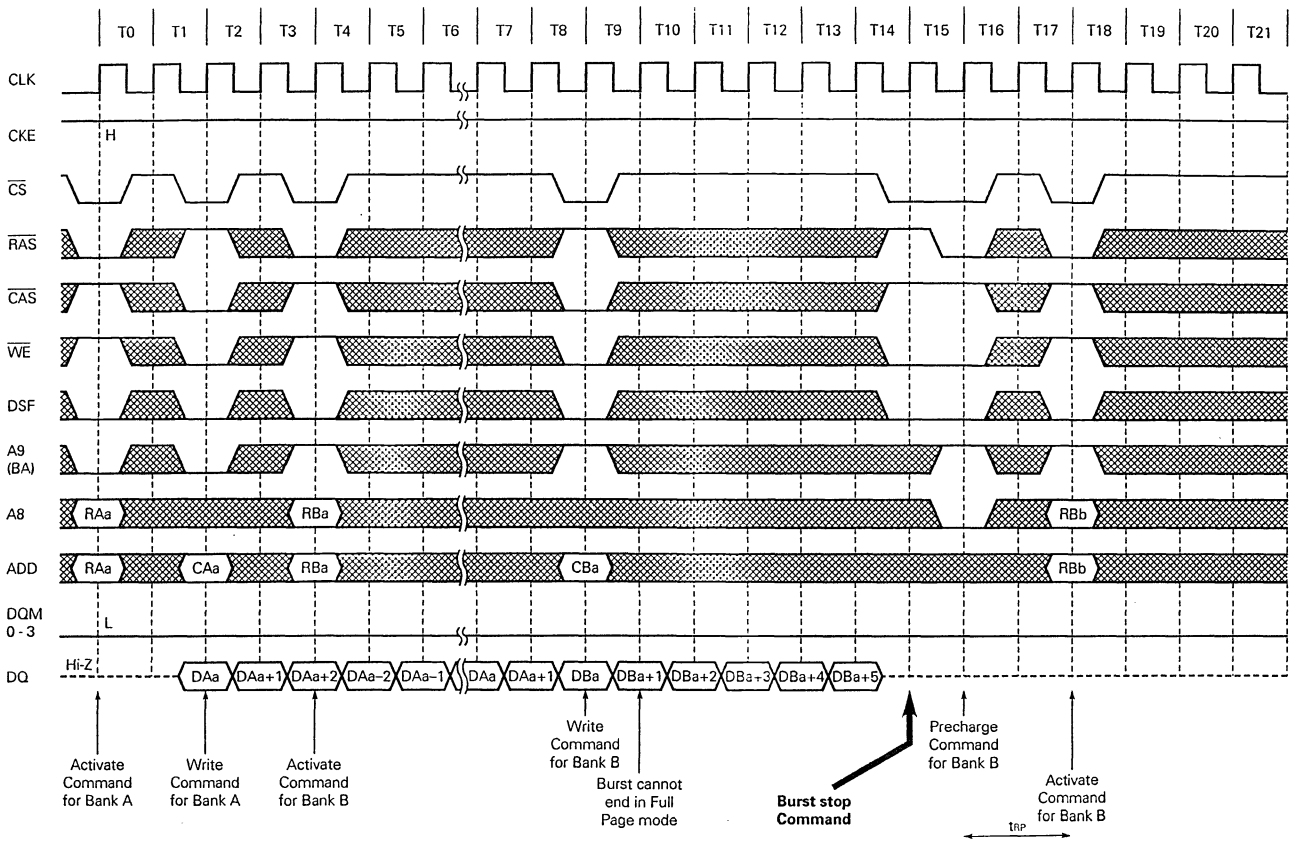
**Full Page READ Cycle (3/3)** (CAS latency = 3)



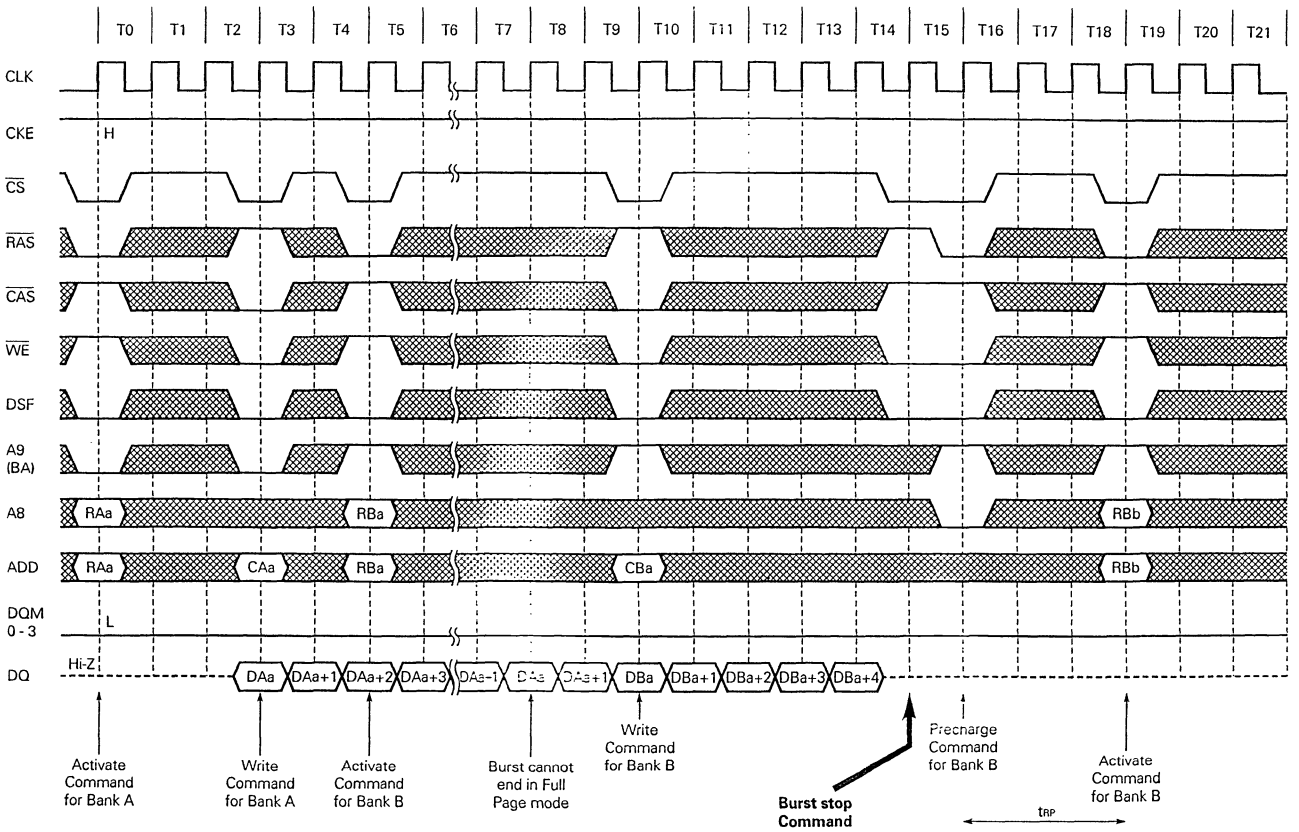
**Full Page WRITE Cycle (1/3)** ($\overline{CAS}$ latency = 1)



**Full Page WRITE Cycle (2/3)** ($\overline{CAS}$ latency = 2)

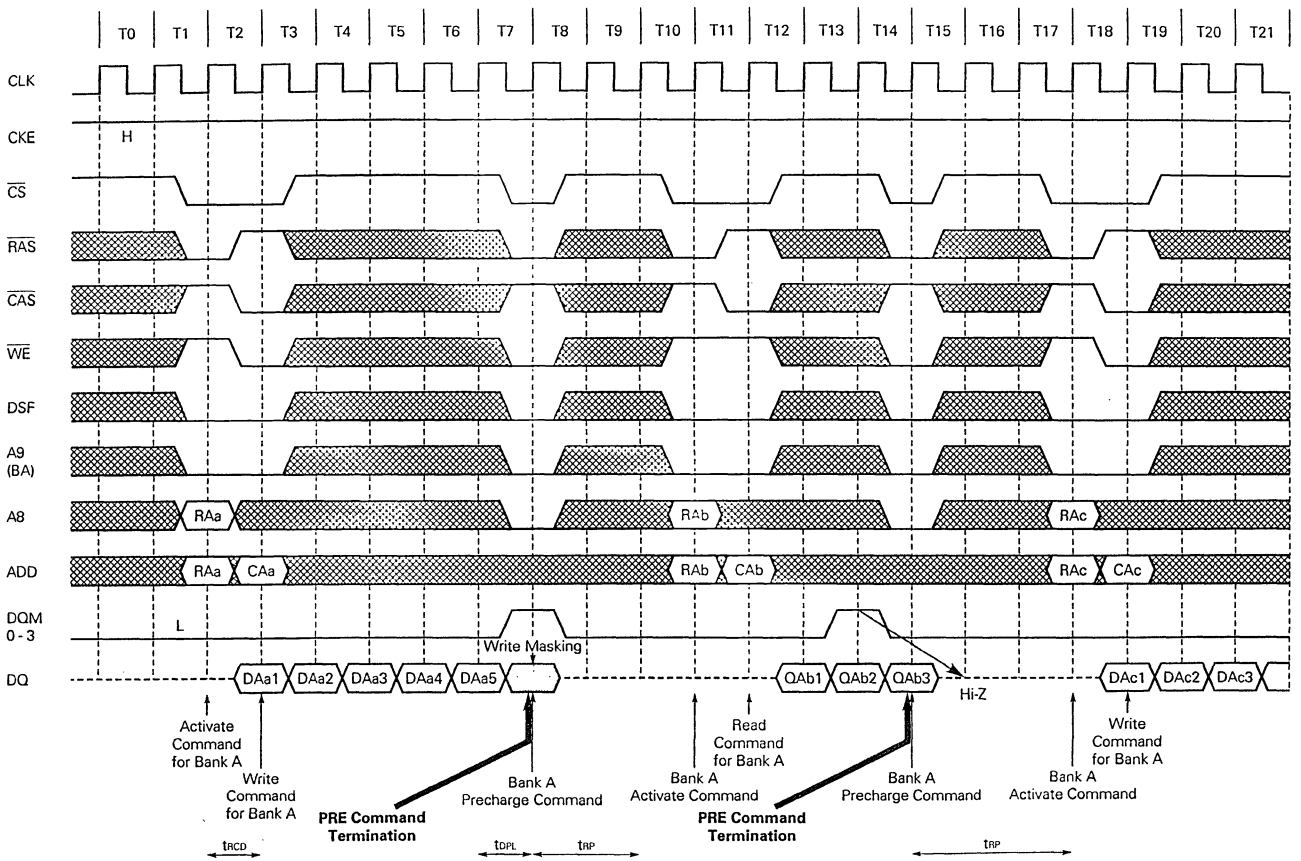


**Full Page WRITE Cycle (3/3)** ($\overline{CAS}$ latency = 3)

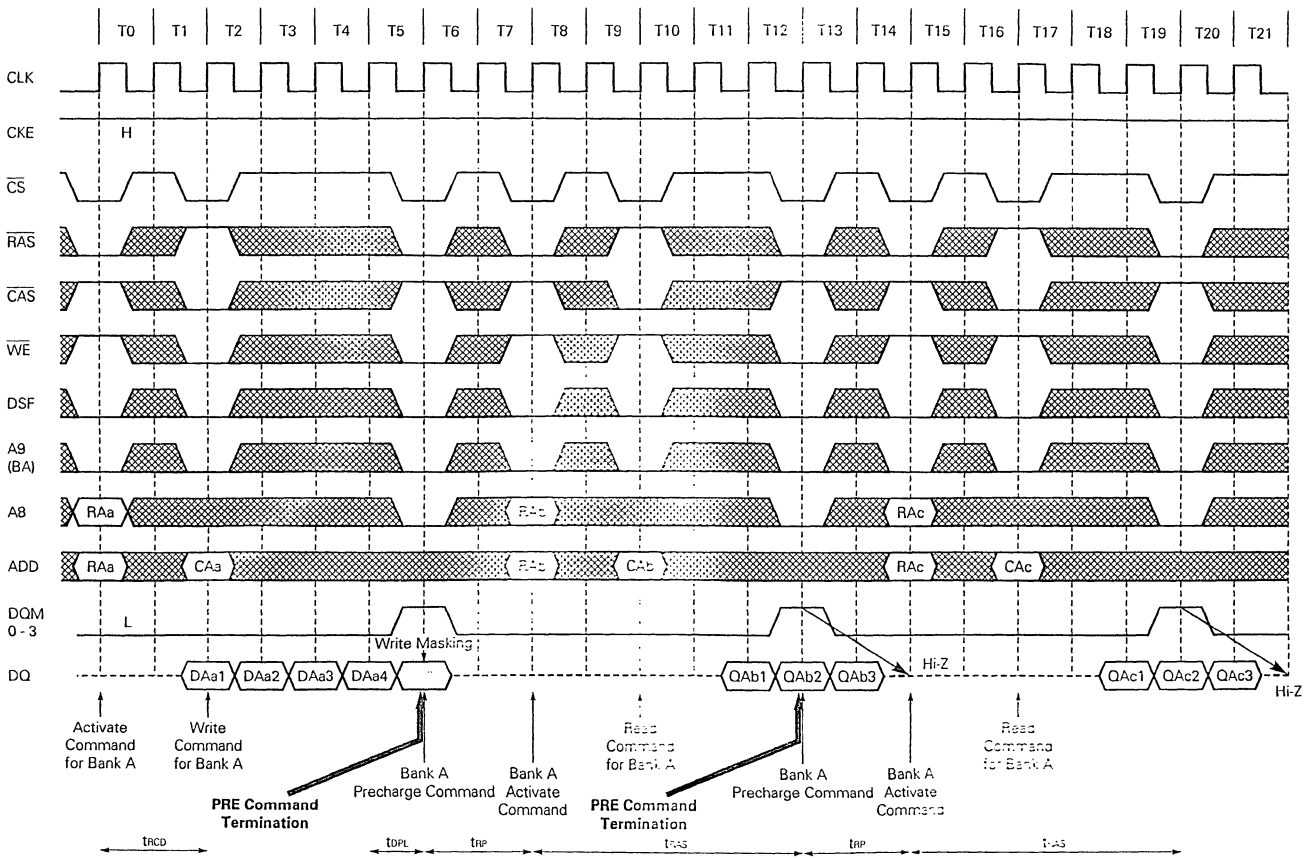


### 16.4.6 Precharge Termination Cycle

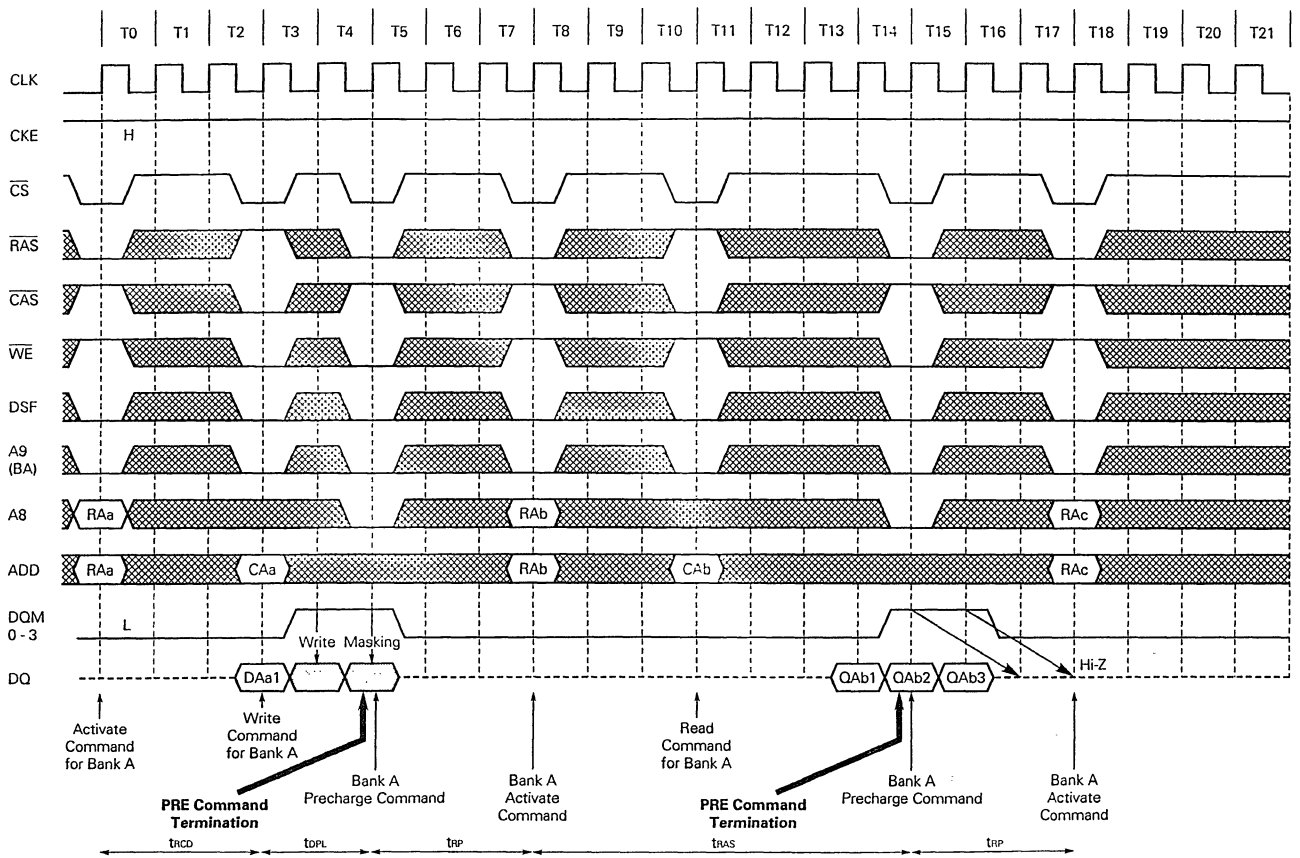
**PRE (Precharge) Termination of Burst (1/3)** (Burst length = 2, 4, 8, Full, $\overline{CAS}$ latency = 1)



**PRE (Precharge) Termination of Burst (2/3)** (Burst length = 2, 4, 8, Full, $\overline{CAS}$ latency = 2)

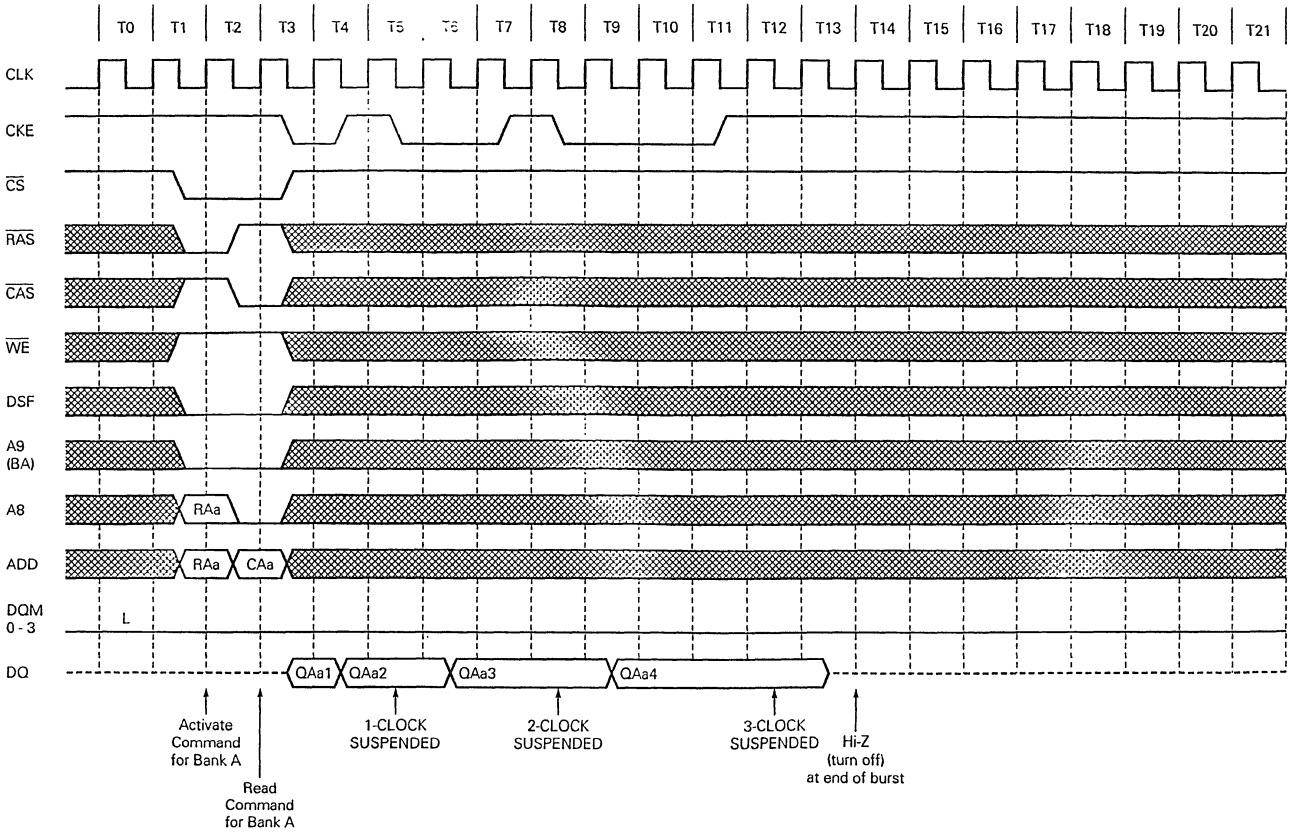


**PRE (Precharge) Termination of Burst (3/3)** (Burst length = 2, 4, 8, Full, $\overline{CAS}$ latency = 3)

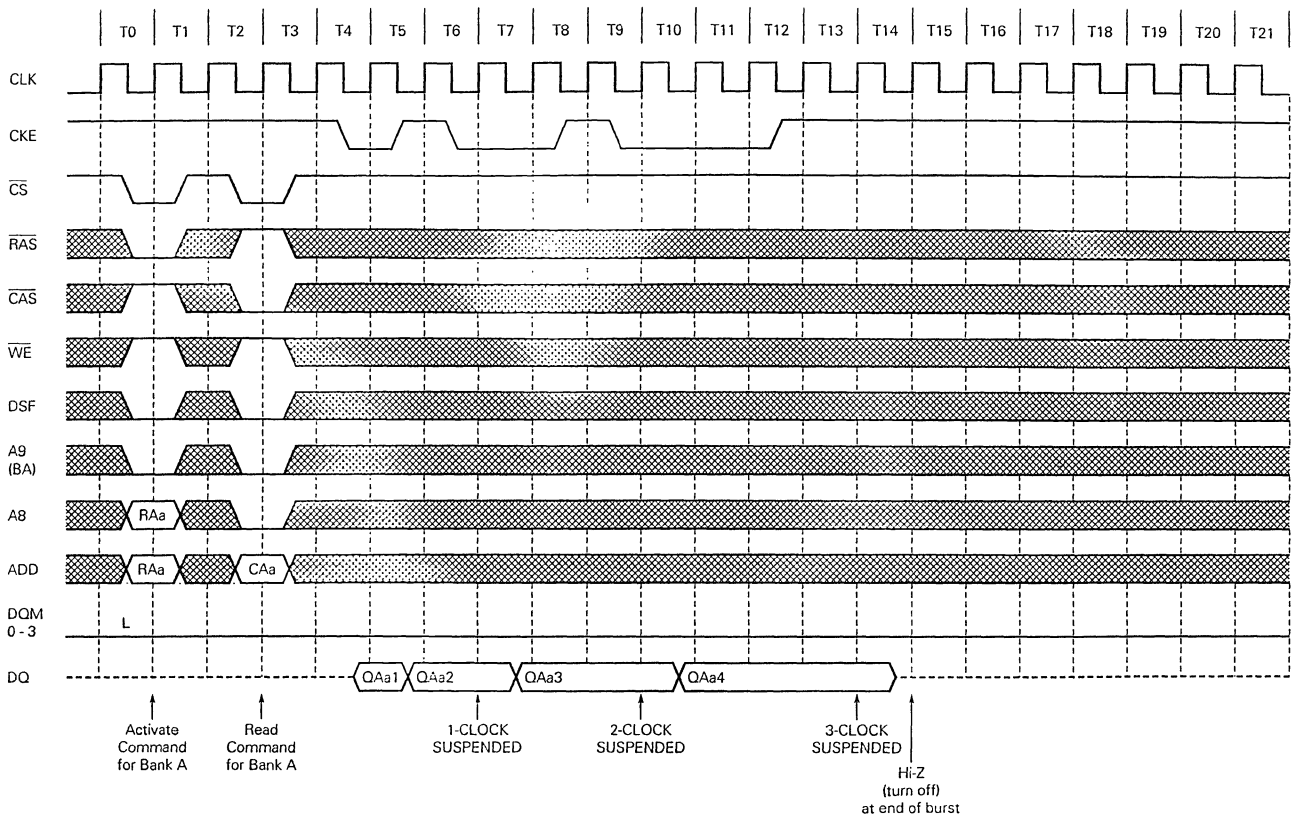


### 16.4.7 Clock Suspension

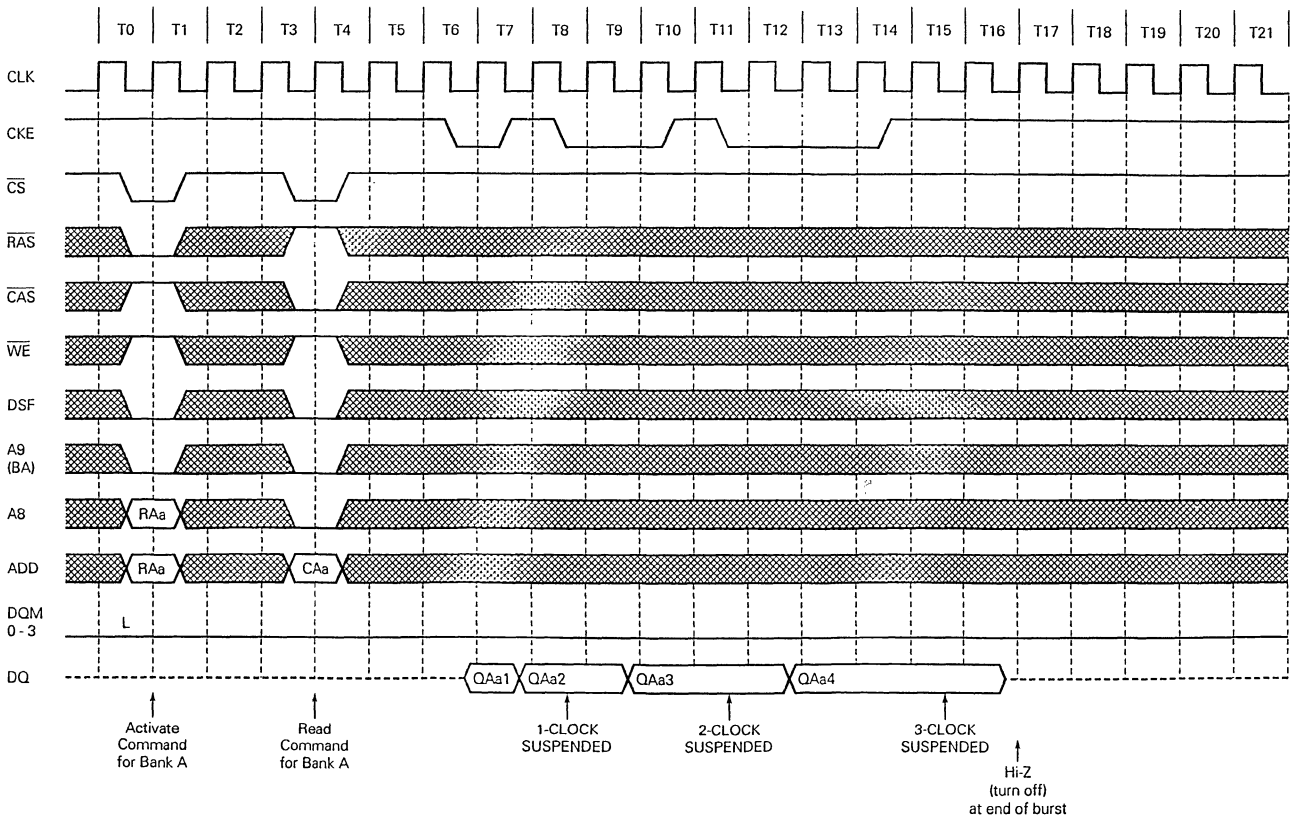
**Clock Suspension during Burst Read (using CKE Function) (1/3)** (Burst length = 4, $\overline{CAS}$ latency = 1)



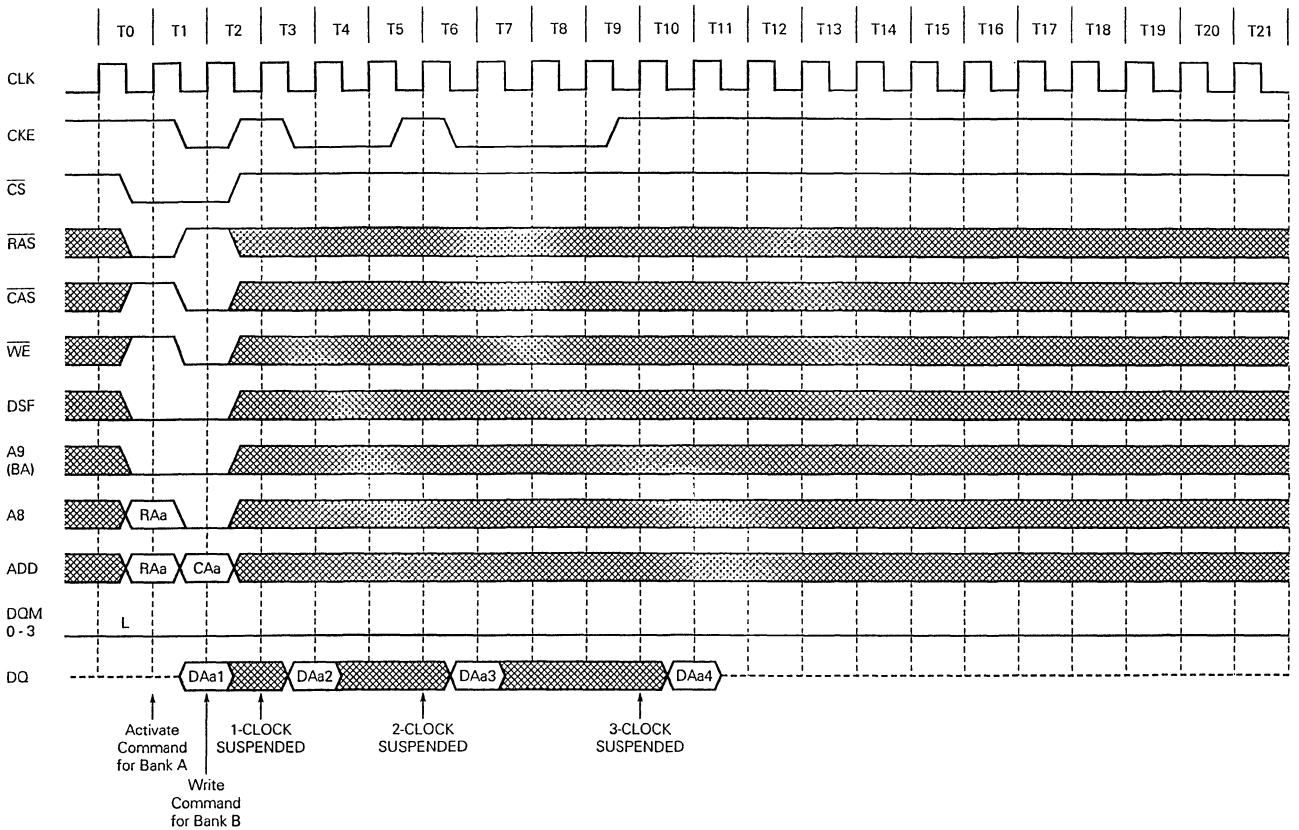
**Clock Suspension during Burst Read (using CKE Function) (2/3)** (Burst length = 4, CAS latency = 2)



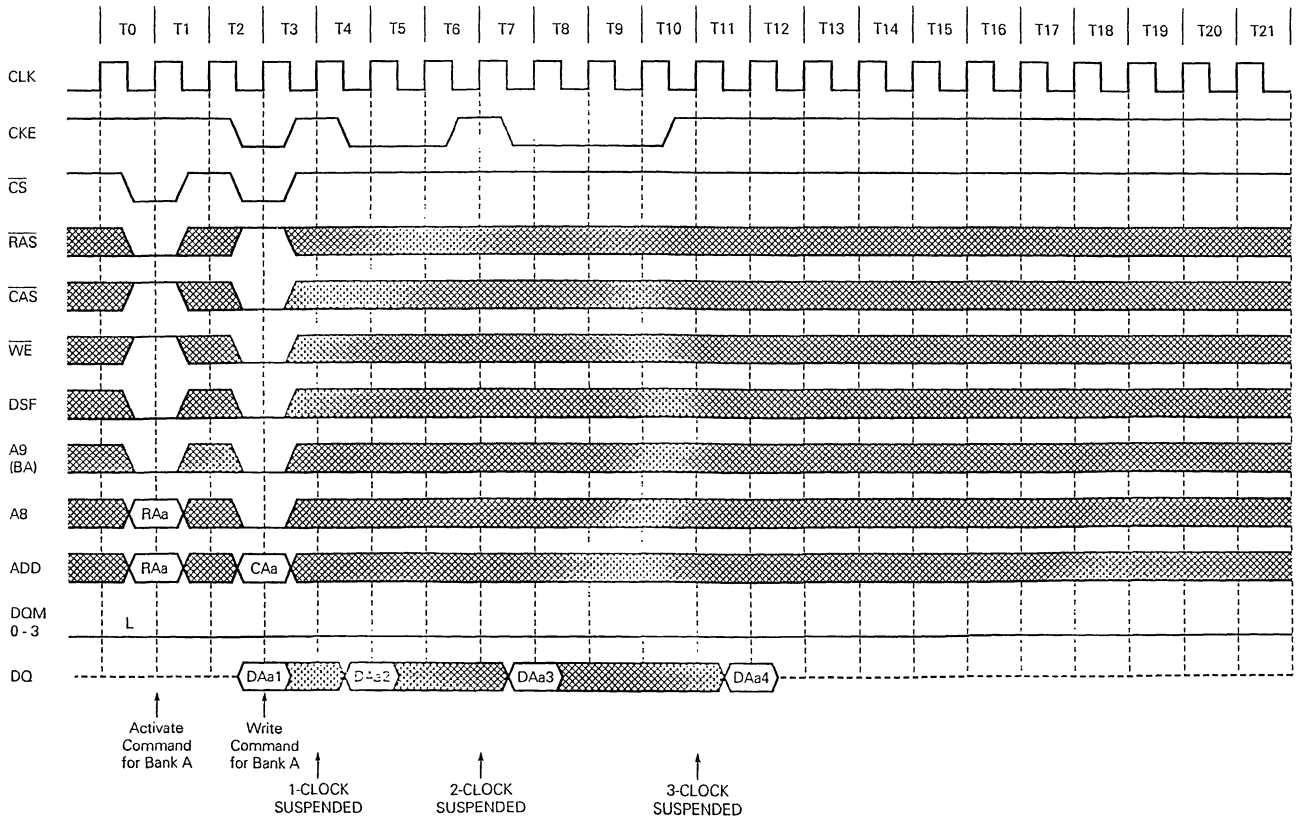
**Clock Suspension during Burst Read (using CKE Function) (3/3)** (Burst length = 4, $\overline{CAS}$ latency = 3)



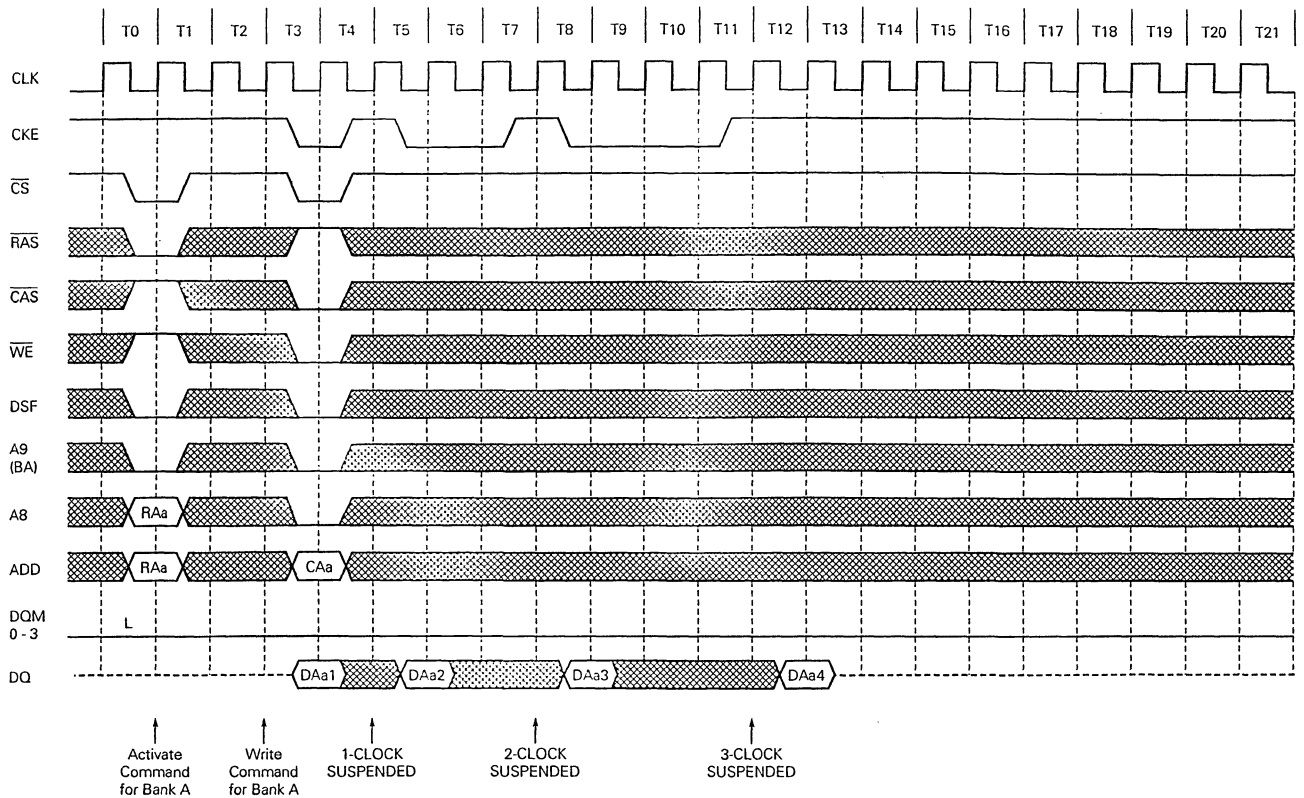
**Clock Suspension during Burst Write (using CKE Function) (1/3)** (Burst length = 4, CAS latency = 1)



**Clock Suspension during Burst Write (using CKE Function) (2/3)** (Burst length = 4, $\overline{CAS}$ latency = 2)

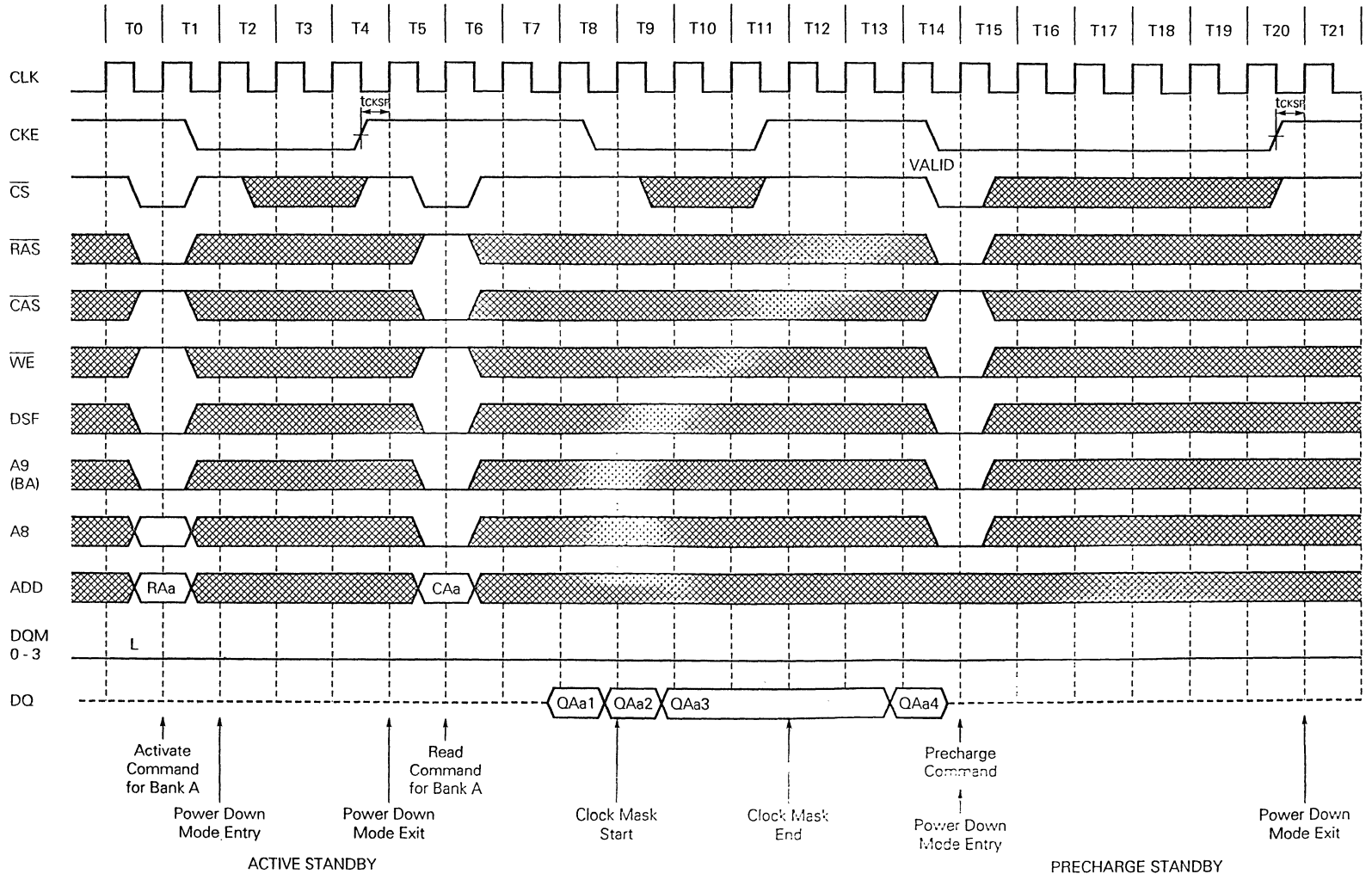


**Clock Suspension during Burst Write (using CKE Function) (3/3)** (Burst length = 4, $\overline{CAS}$ latency = 3)



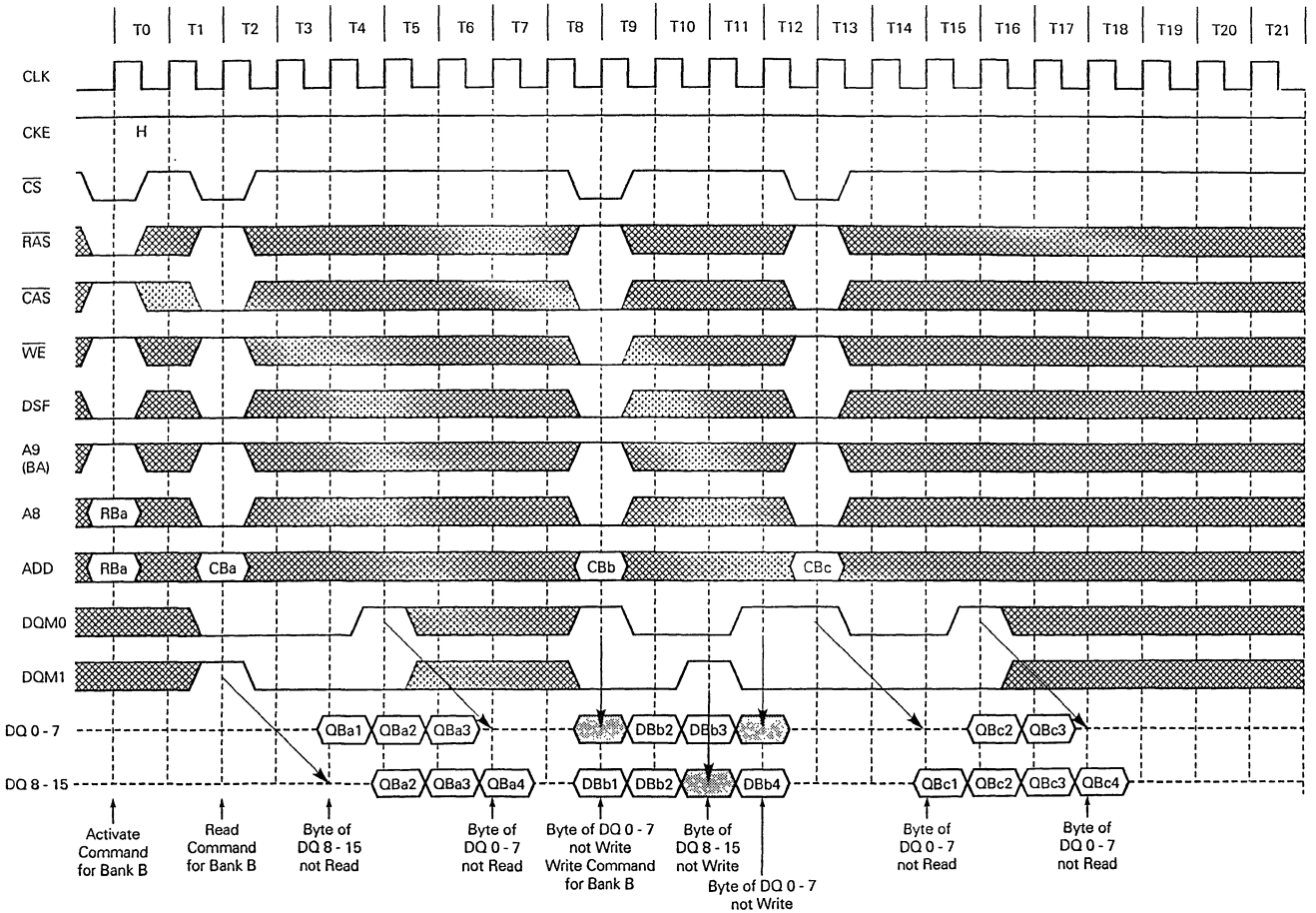
### 16.4.8 Power Down Mode

**Power Down Mode and Clock Suspension** (Burst length = 4, $\overline{CAS}$ latency = 2)

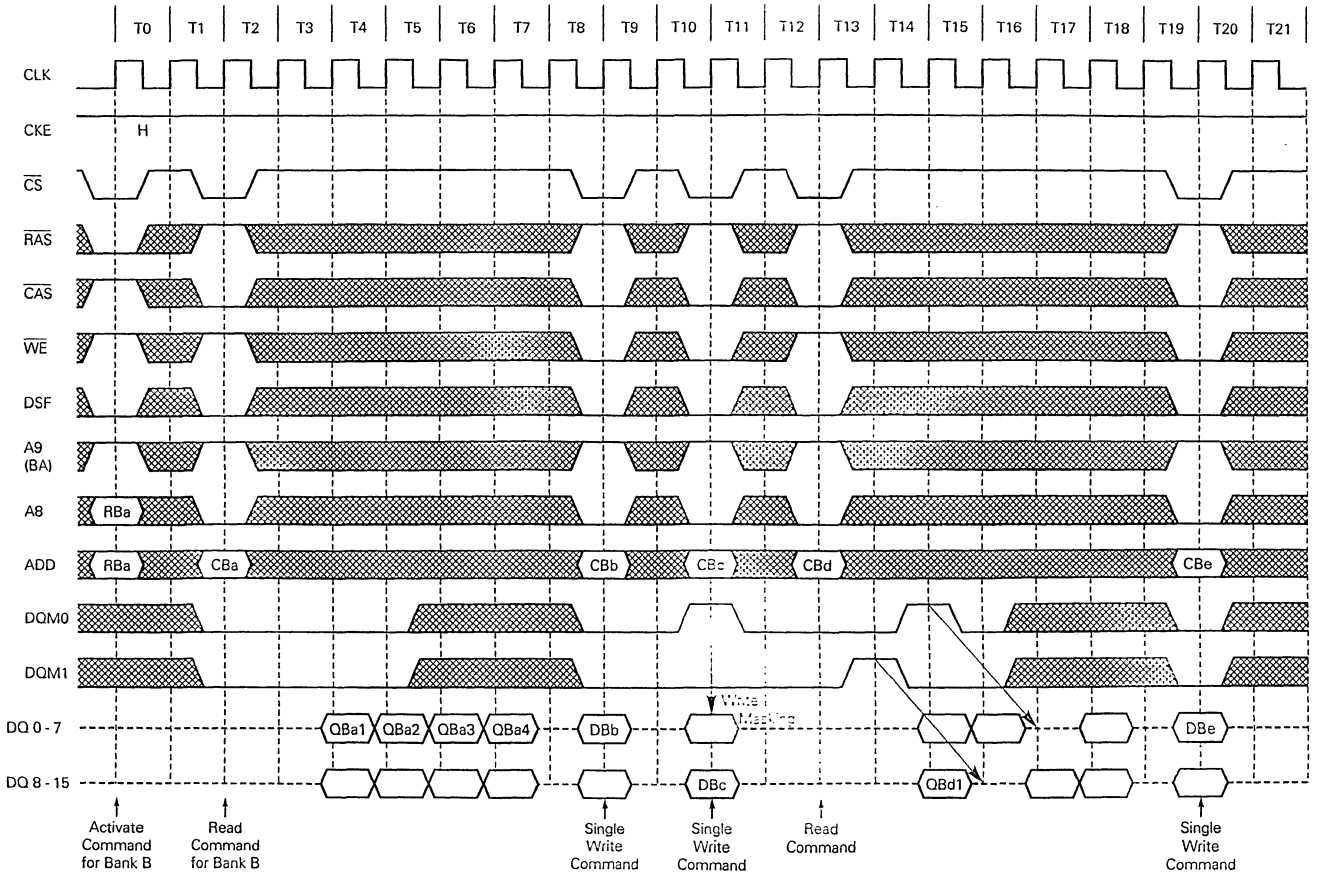


### 16.4.9 Other Cycles

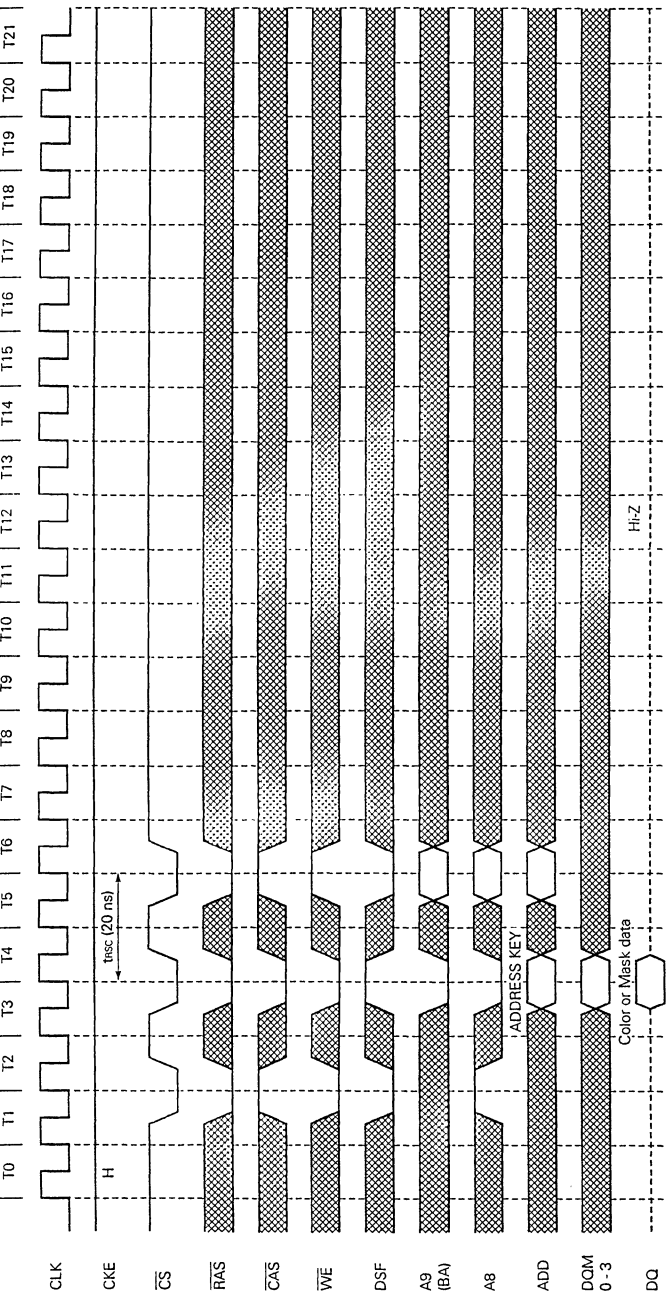
**Byte Read/Write Operation (by DQM)** (Burst length = 4, $\overline{CAS}$ latency = 2)



**Burst Read and Single Write** (Burst length = 4, $\overline{CAS}$ latency = 2)

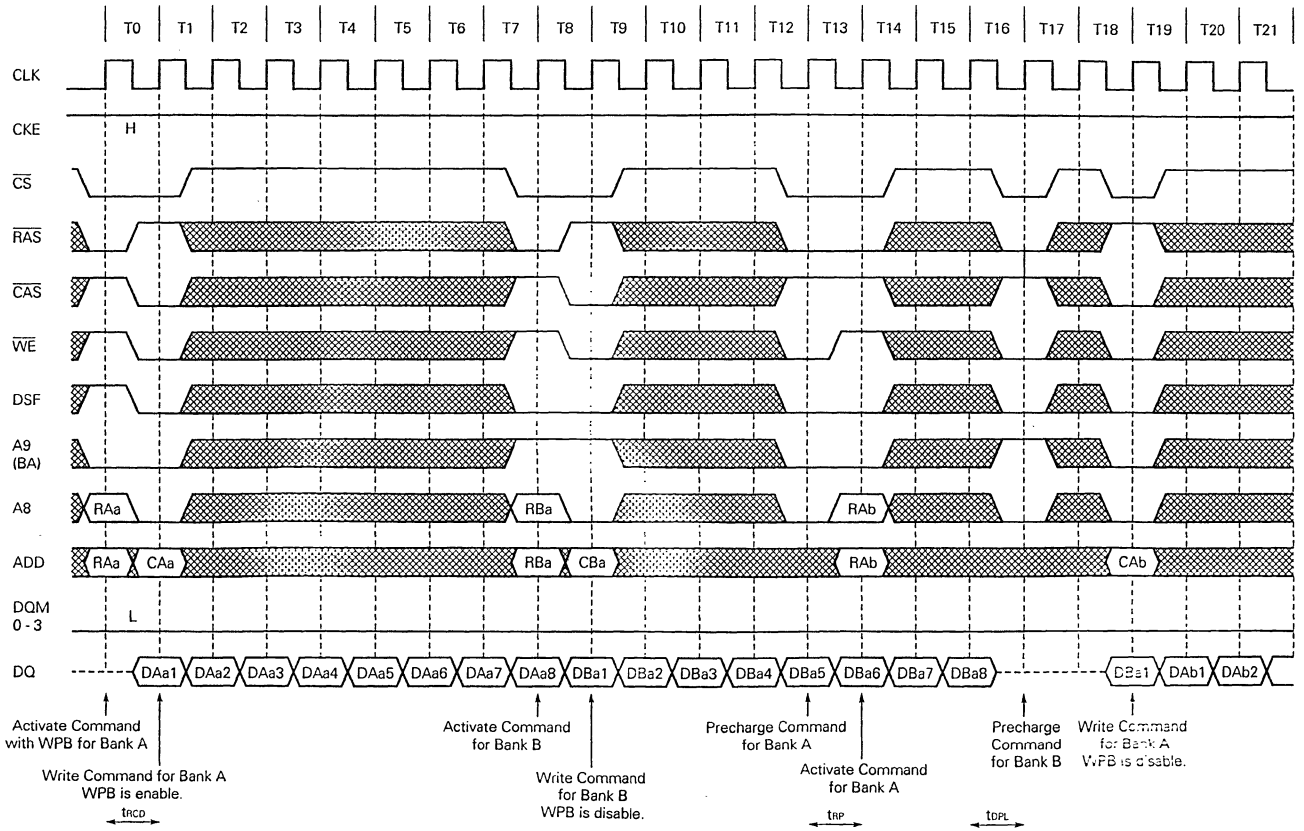


**Special Register Set** (Burst length = 4, $\overline{CAS}$ latency = 2)

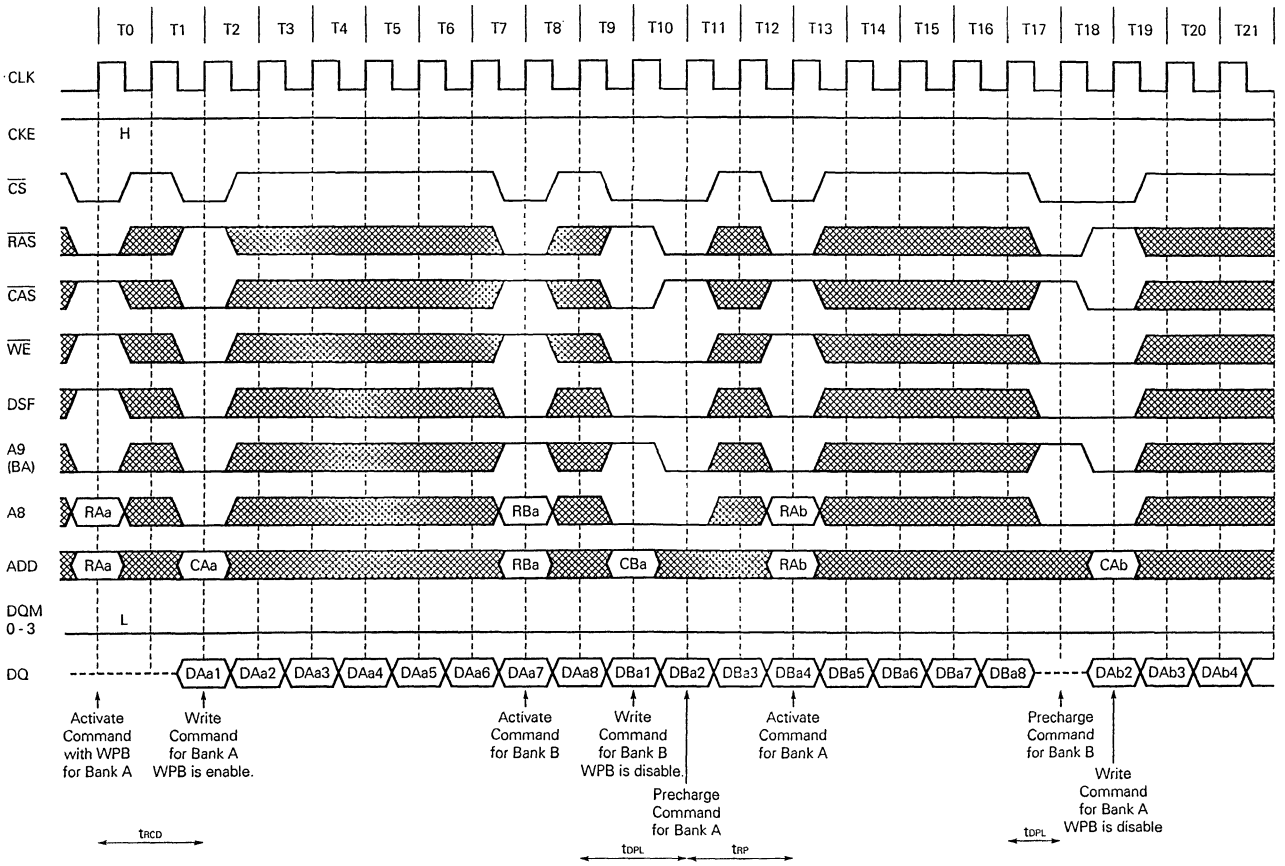



**Remark** Special Register Set command is able to input at any state.

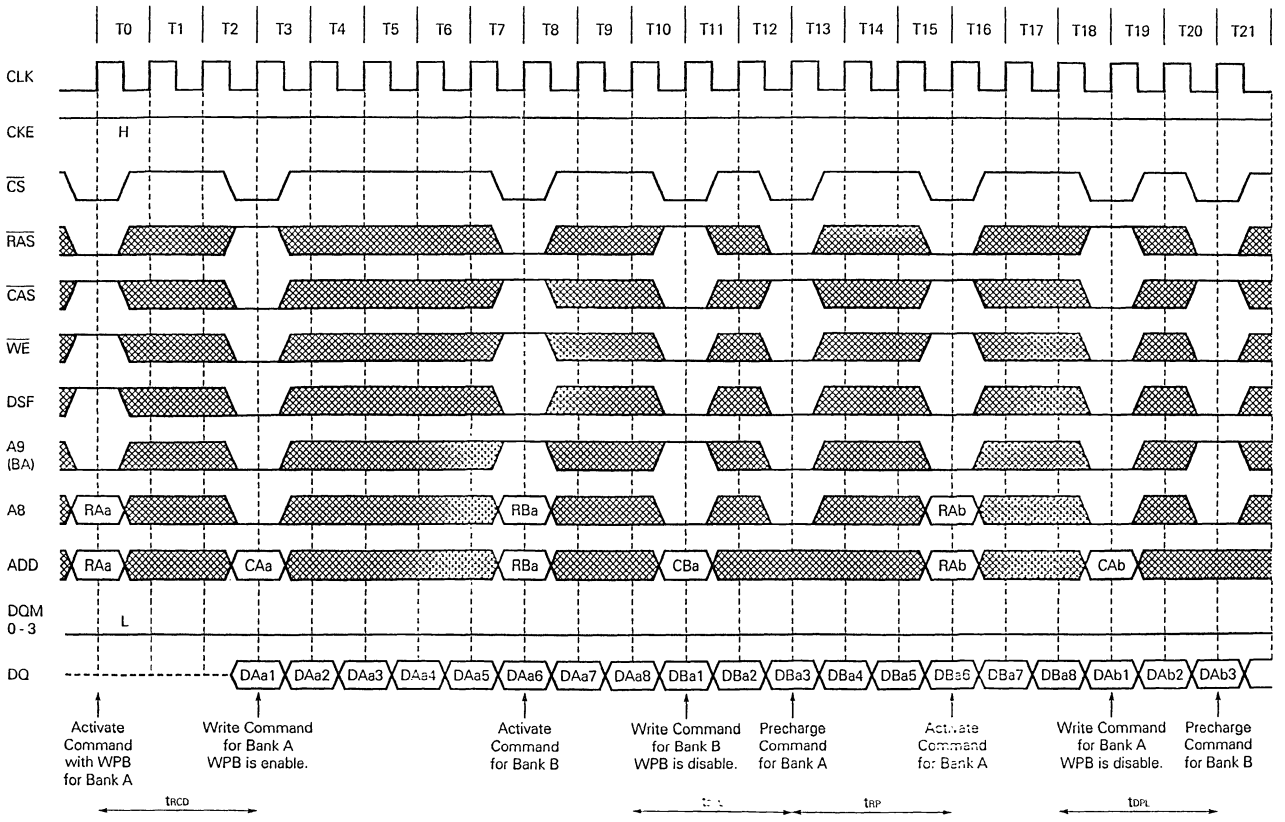
**Random Row Write with WPB (Pingpong banks) (1/3)** (Burst length = 8, $\overline{CAS}$ latency = 1)



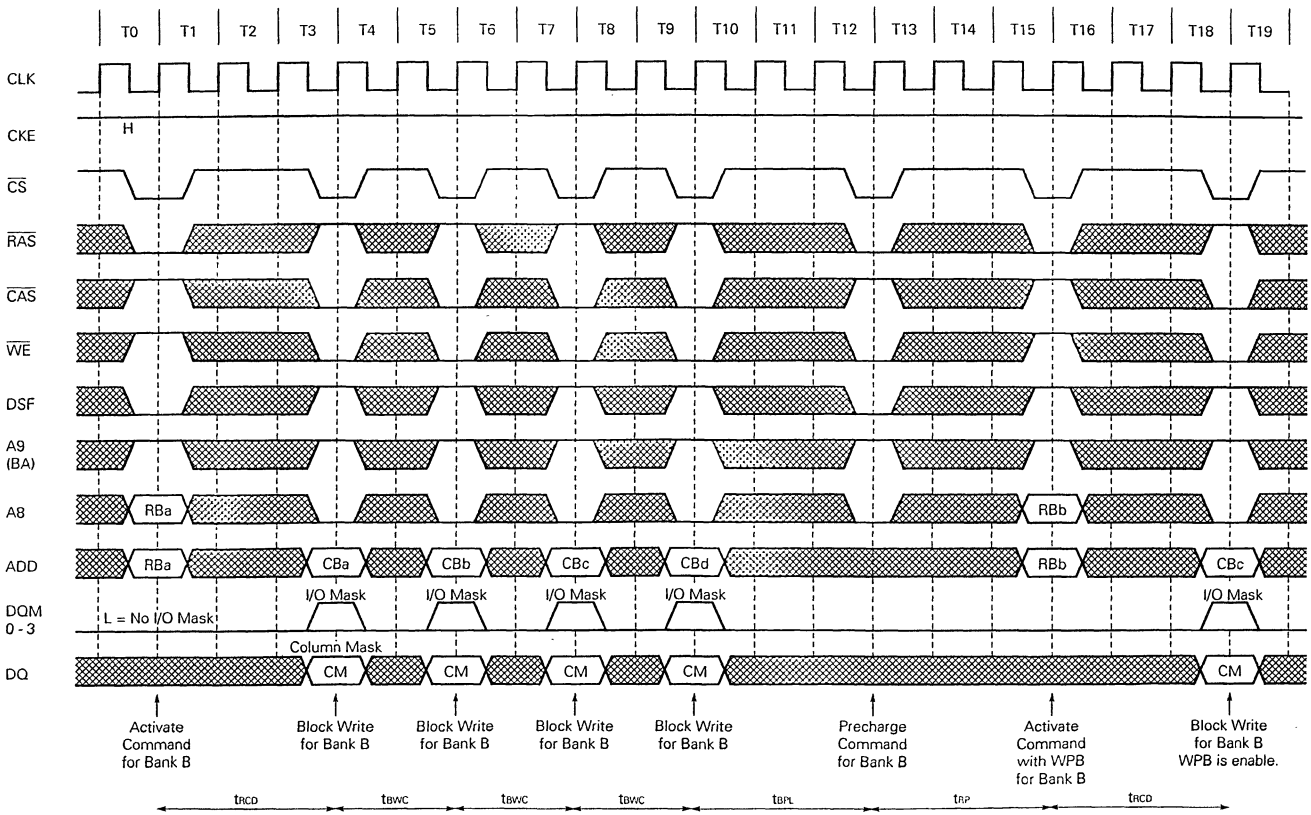
**Random Row Write with WPB (Pingpong banks) (2/3)** (Burst length = 8, CAS latency = 2)



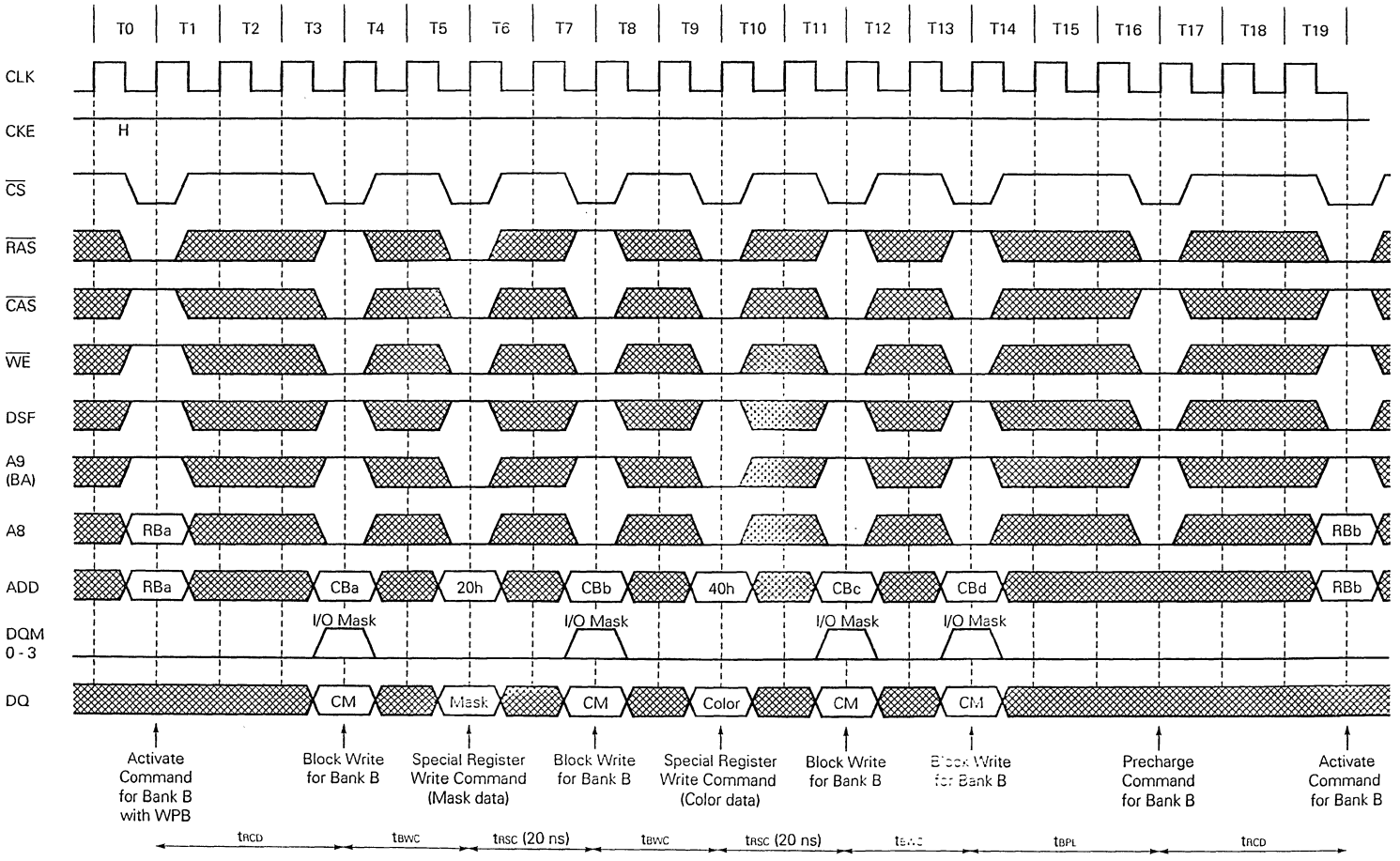
**Random Row Write with WPB (Pingpong banks) (3/3)** (Burst length = 8, $\overline{CAS}$ latency = 3)



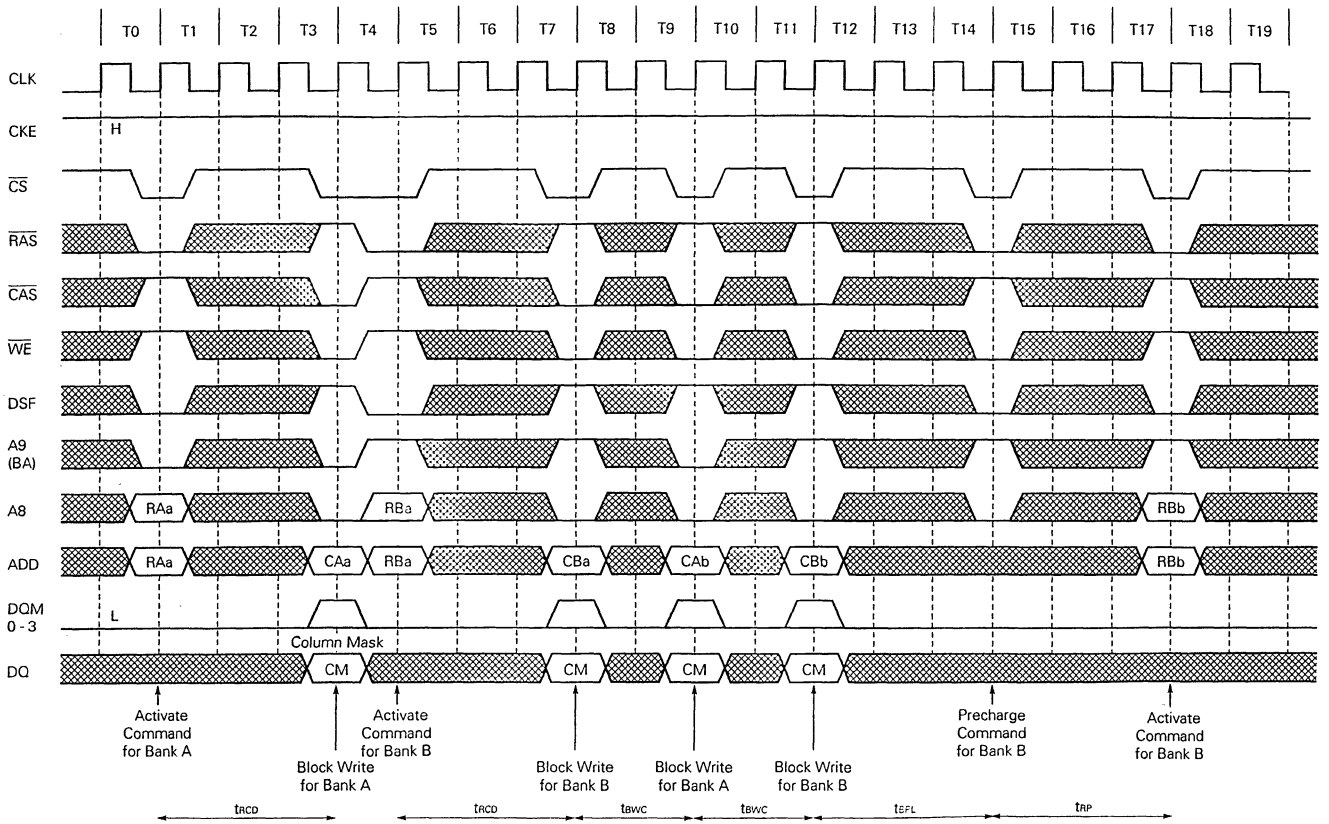
**Block Write (page at same bank)** ($\overline{CAS}$ latency = 3)



**Block Write (page at same bank) changing color and mask data ($\overline{CAS}$ latency = 3)**



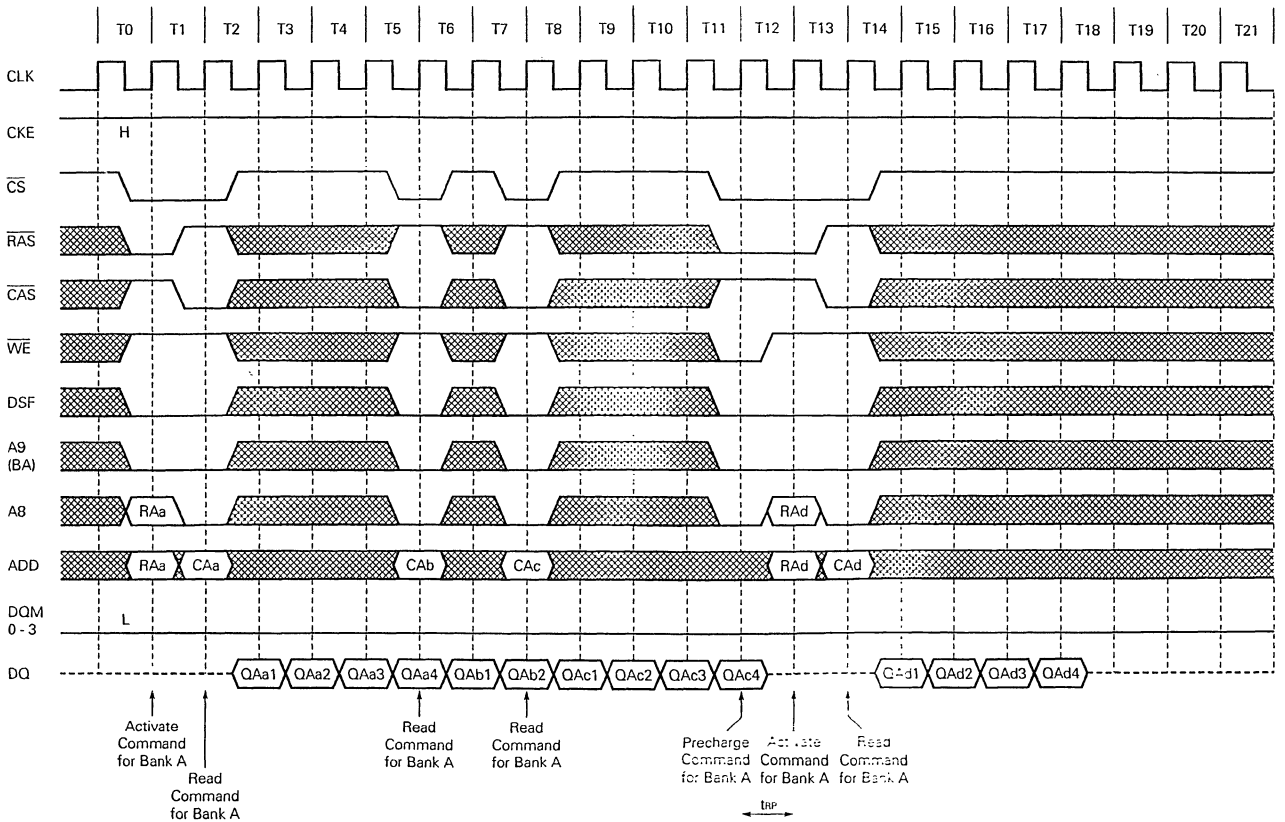
**Interleaved Block Write ($\overline{CAS}$ latency = 3)**



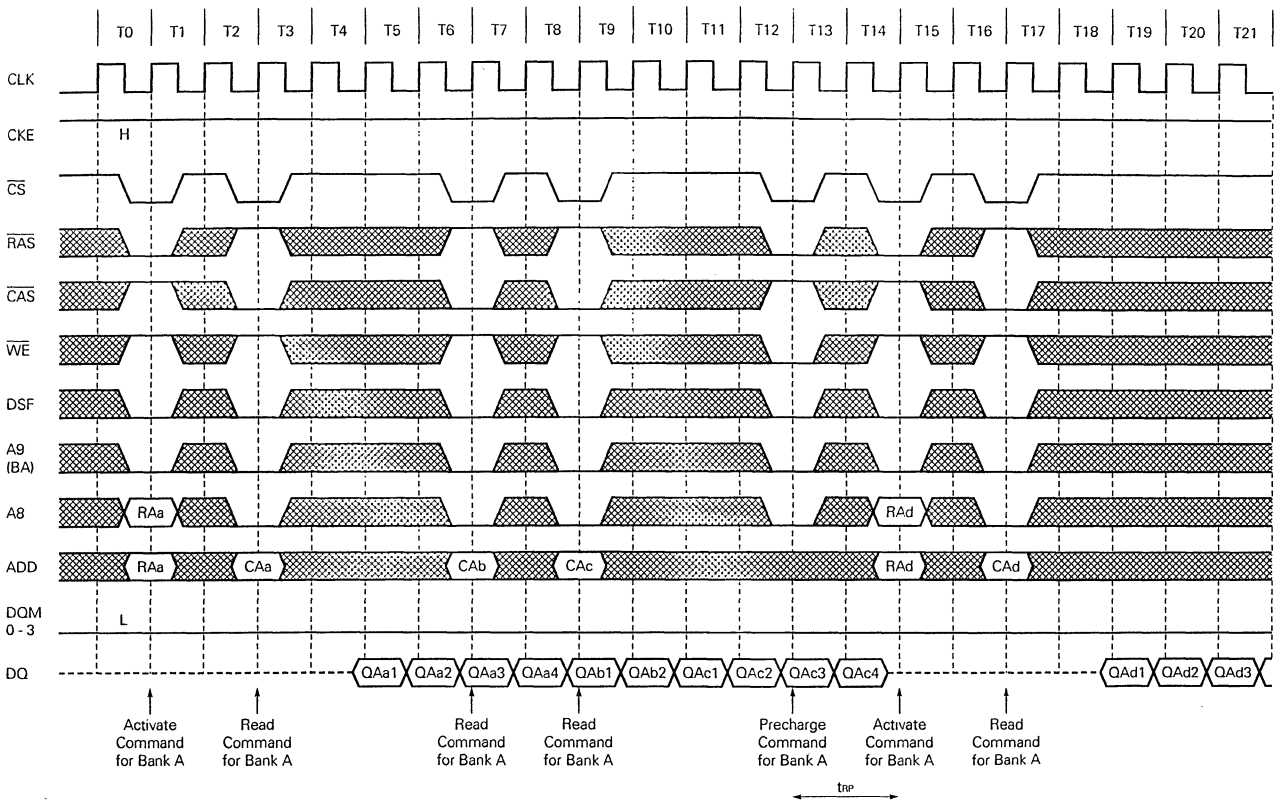
## 16.6 Application Cycles

### 16.6.1 Page Cycles with Same Bank

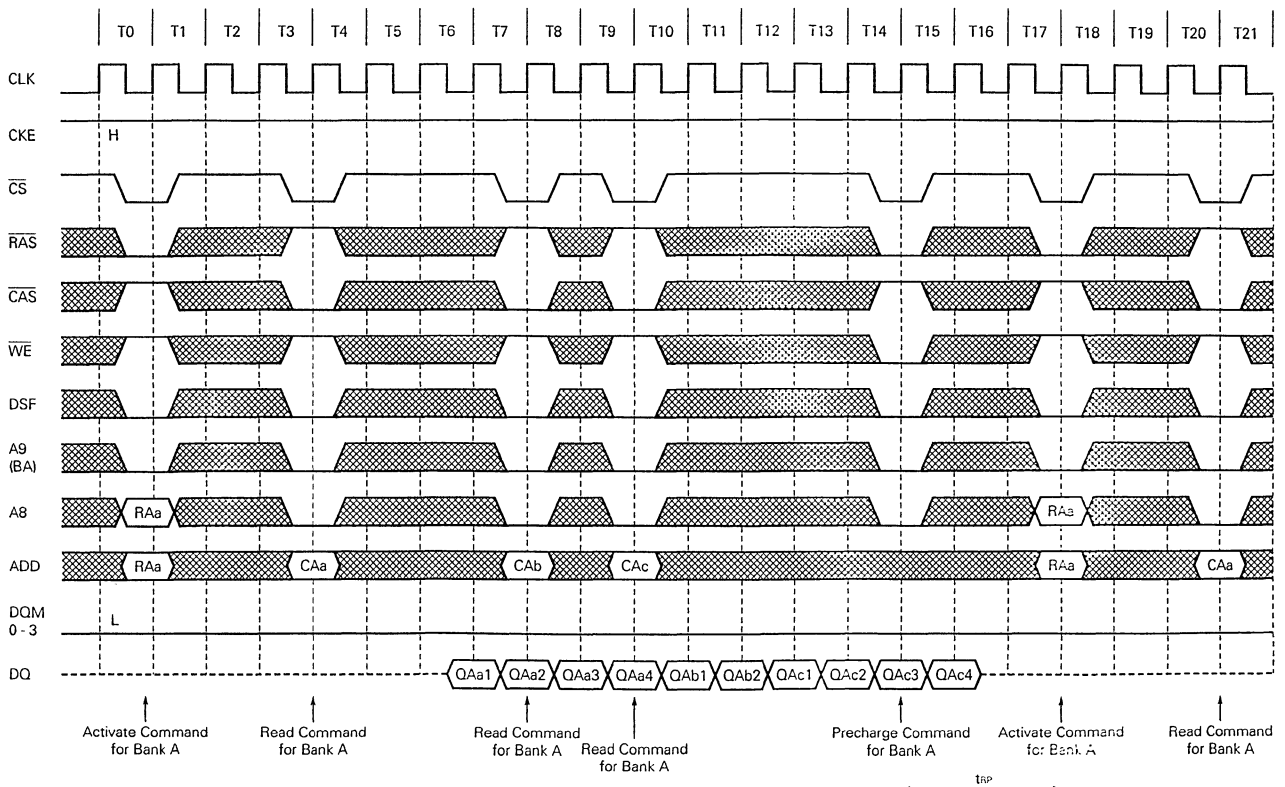
**Random Column Read (Page with same bank) (1/3)** (Burst length = 4, $\overline{CAS}$ latency = 1)



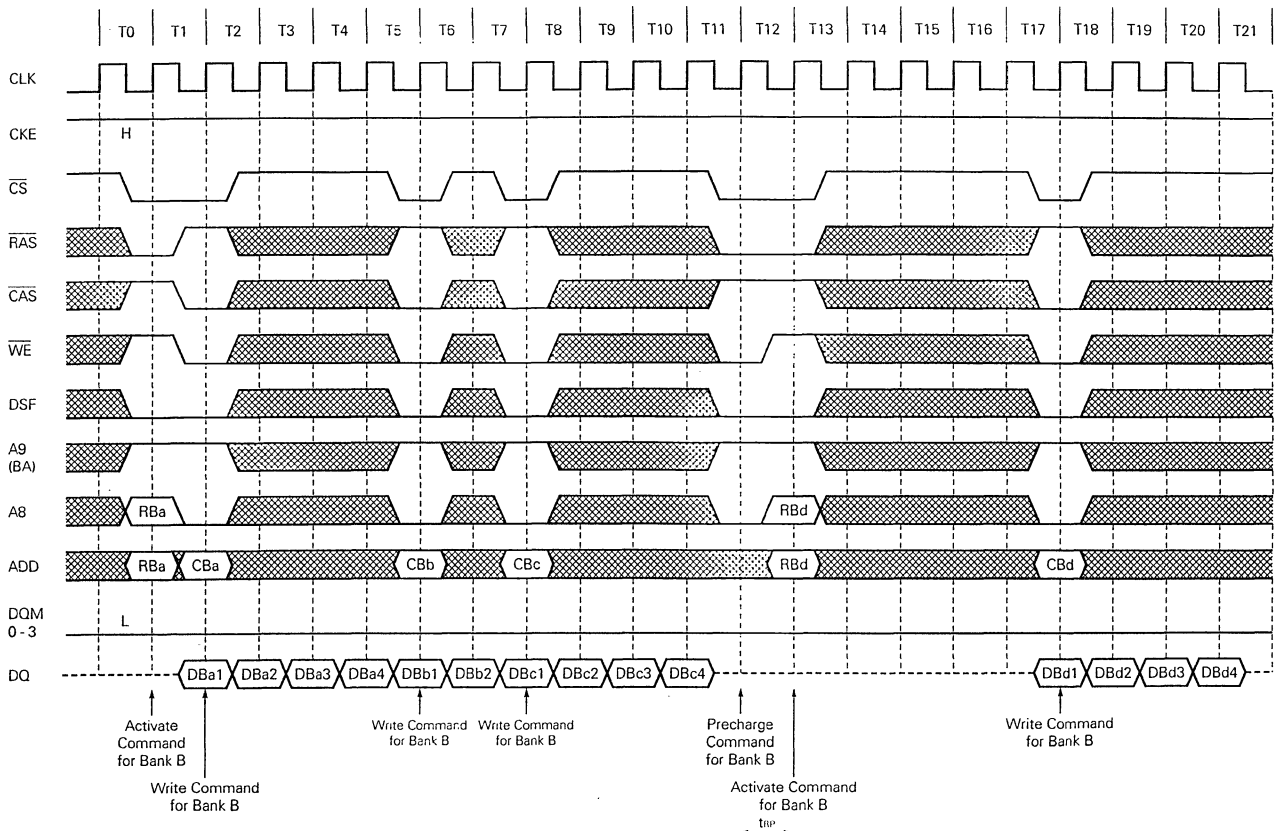
**Random Column Read (Page with same bank) (2/3)** (Burst length = 4, $\overline{CAS}$ latency = 2)



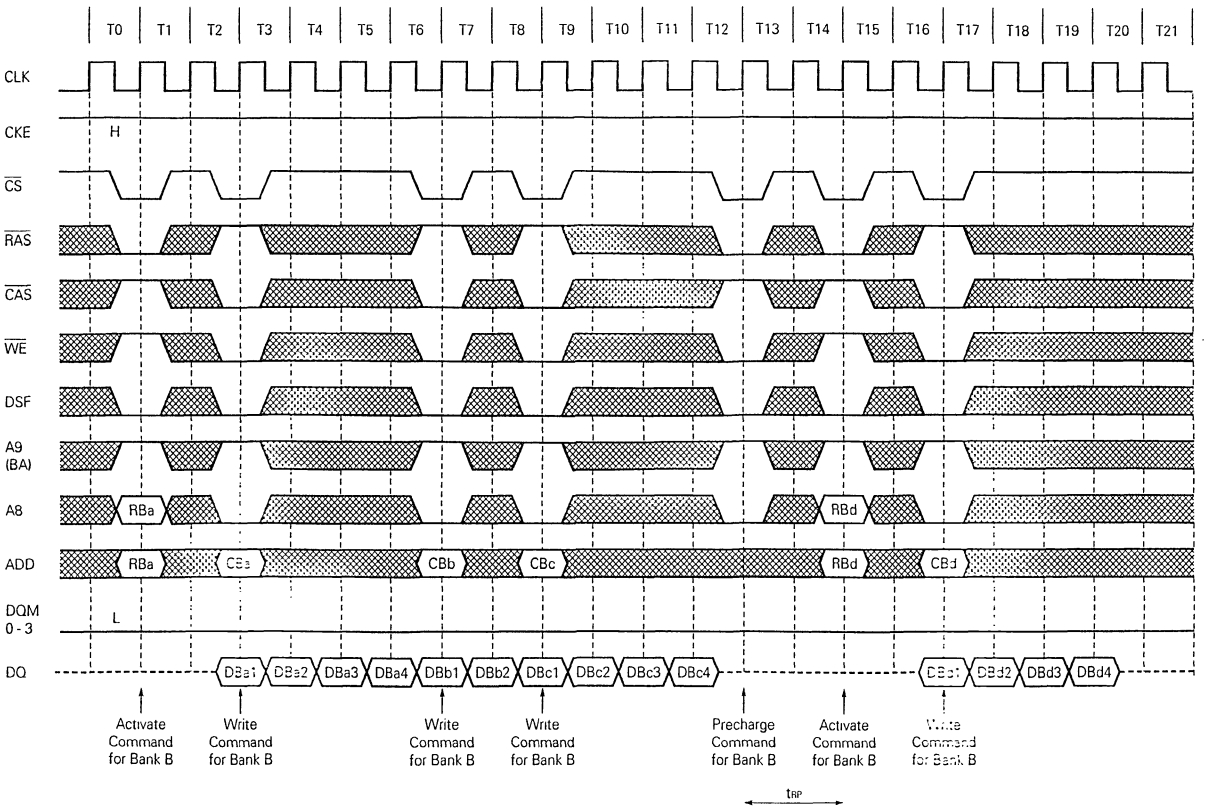
**Random Column Read (Page with same bank) (3/3)** (Burst length = 4, $\overline{CAS}$ latency = 3)



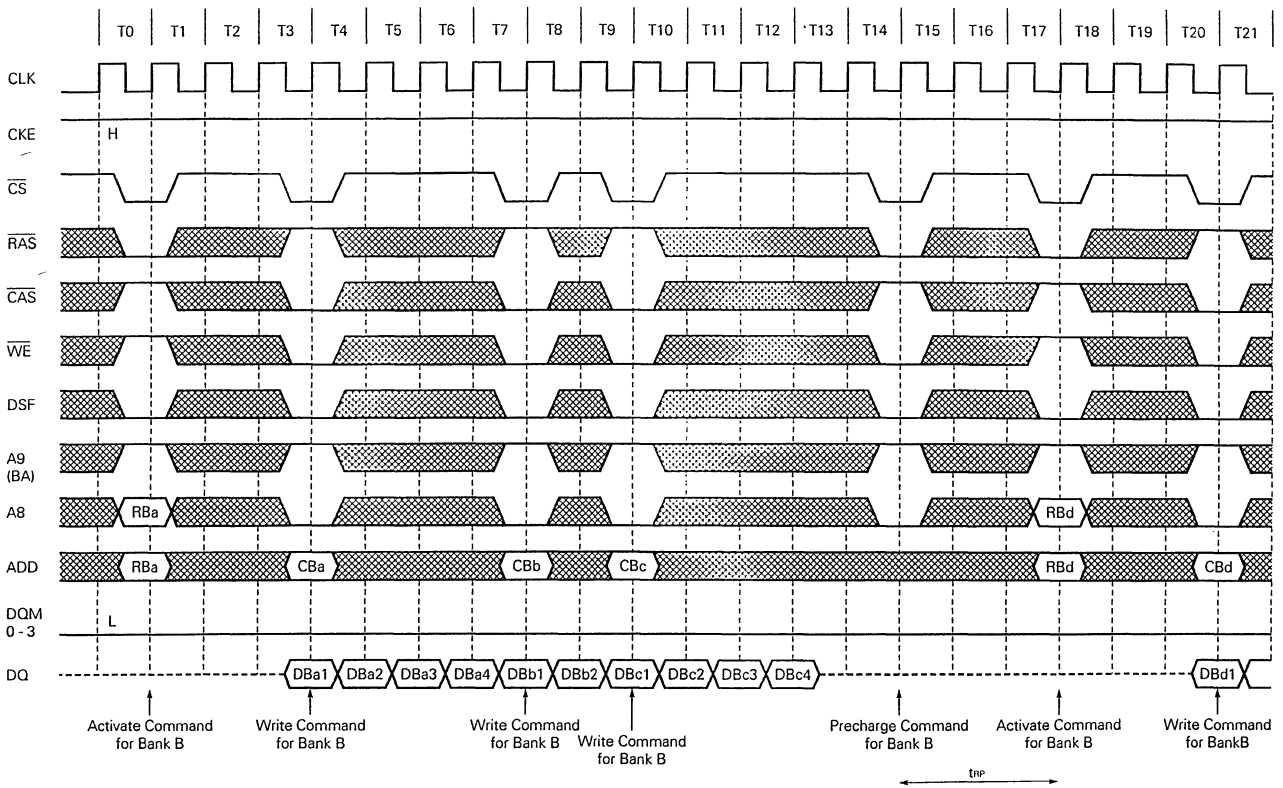
**Random Column Write (Page with same bank) (1/3)** (Burst length = 4, CAS latency = 1)



**Random Column Write (Page with same bank) (2/3)** (Burst length = 4, $\overline{CAS}$ latency = 2)

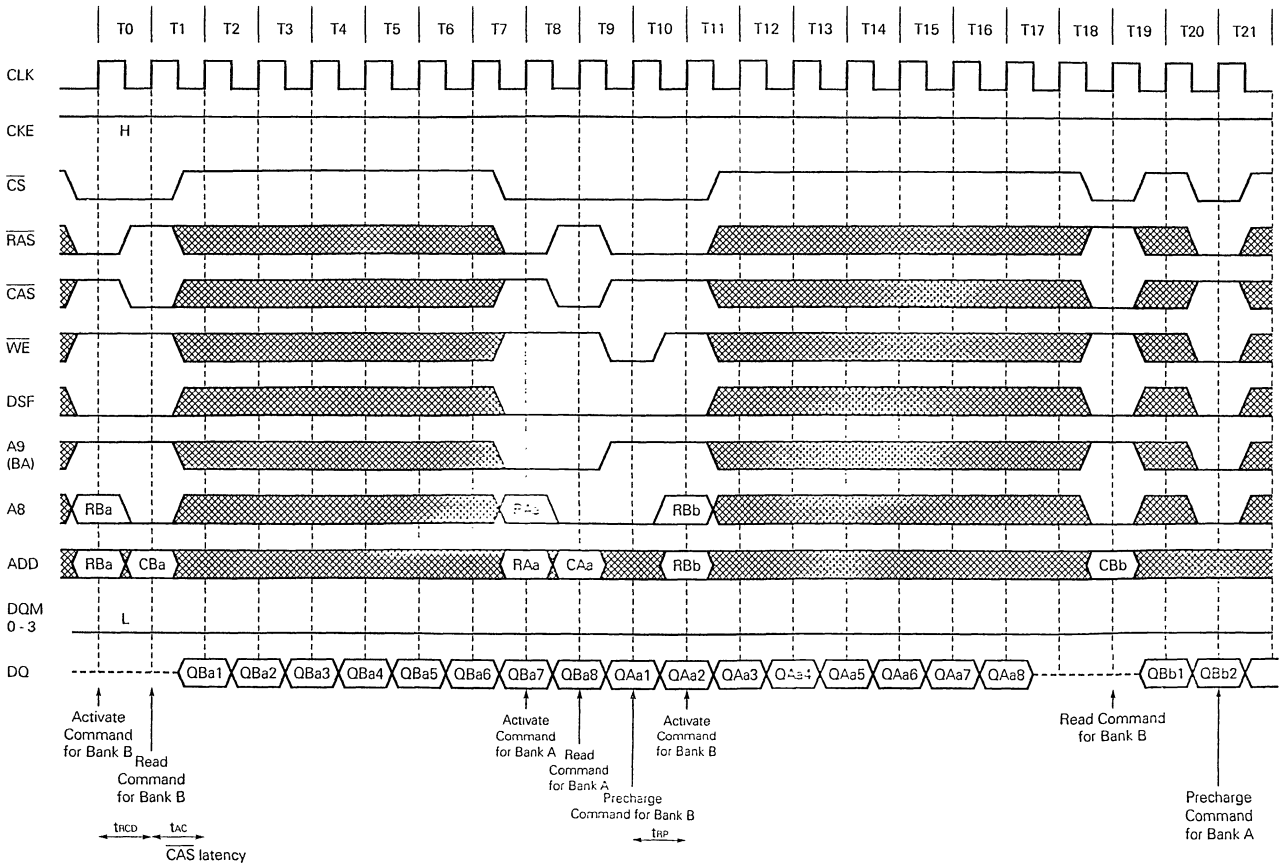


**Random Column Write (Page with same bank) (3/3)** (Burst length = 4, $\overline{CAS}$ latency = 3)

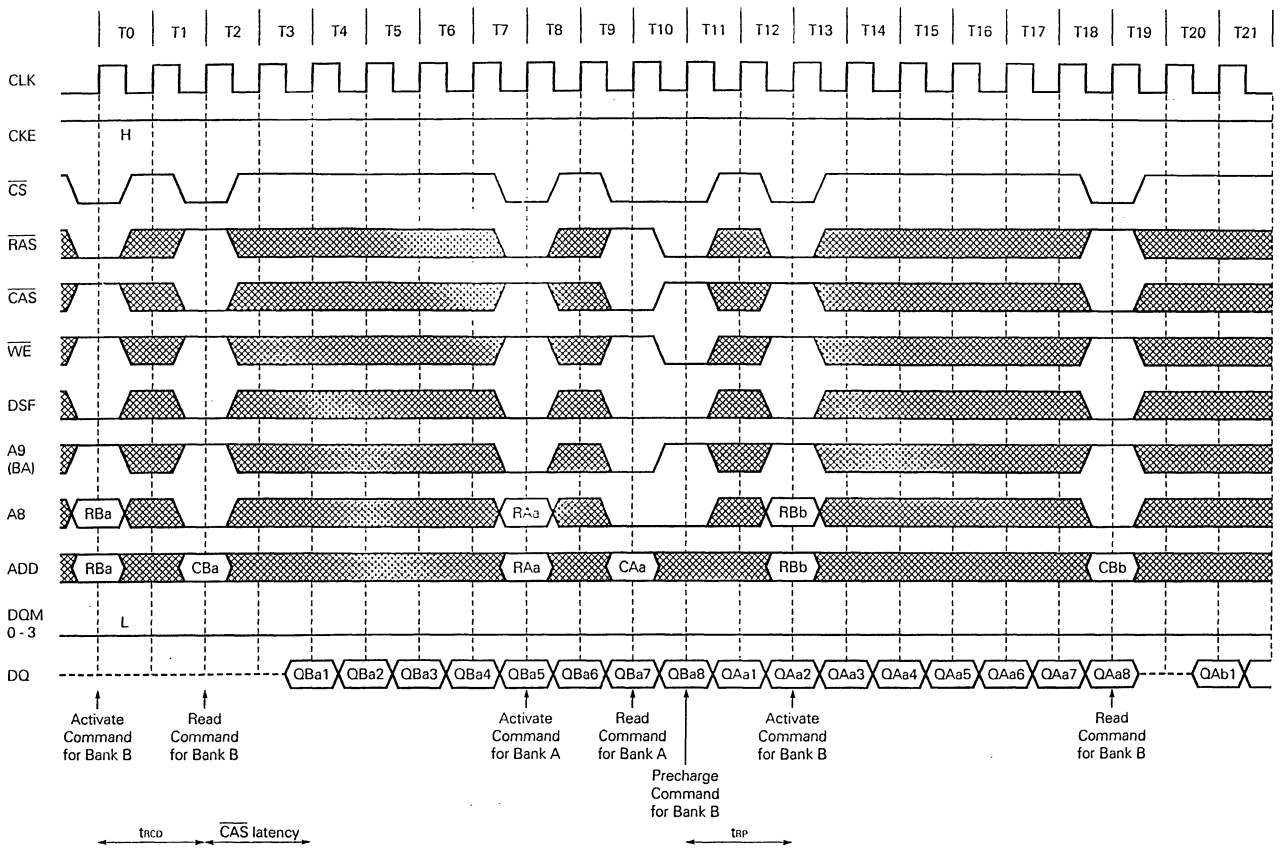


### 16.6.2 Cycles with Pingpong Banks

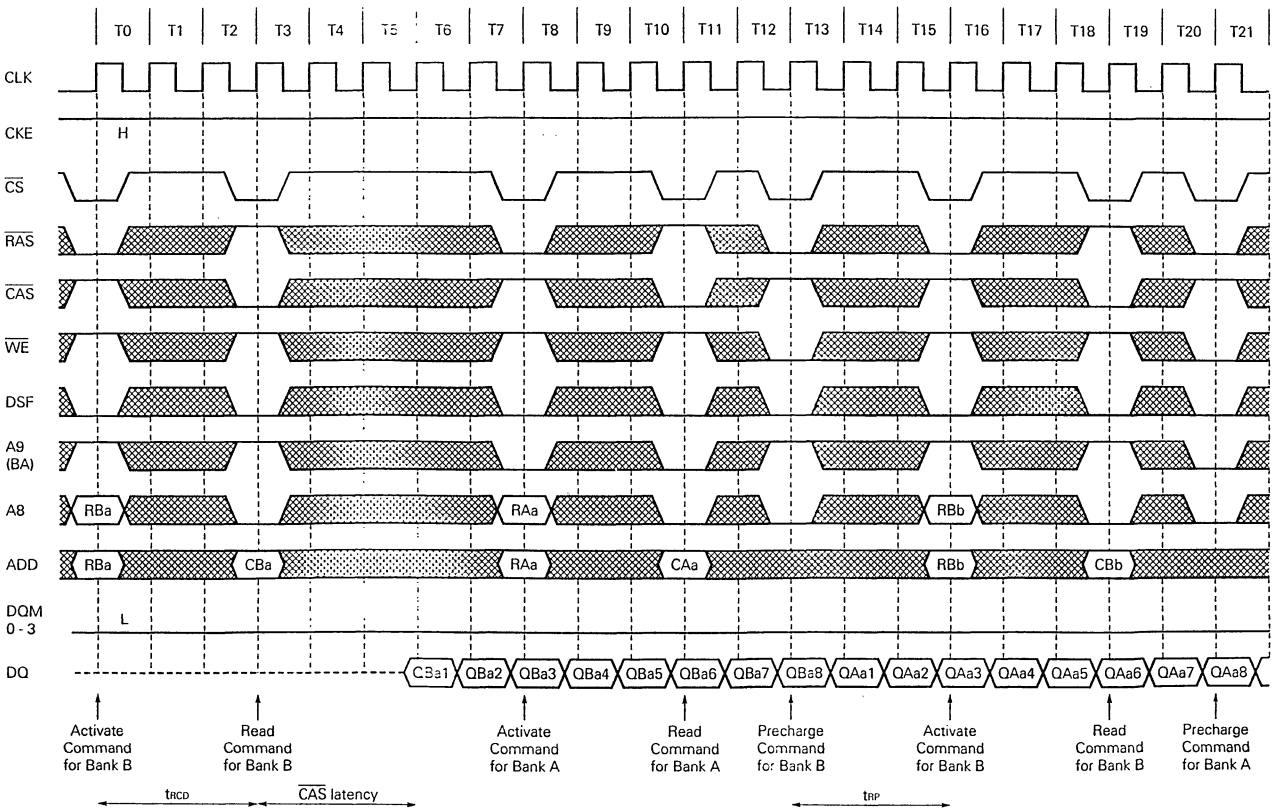
**Random Row Read (Pingpong banks) (1/3)** (Burst length = 8, $\overline{CAS}$ latency = 1)



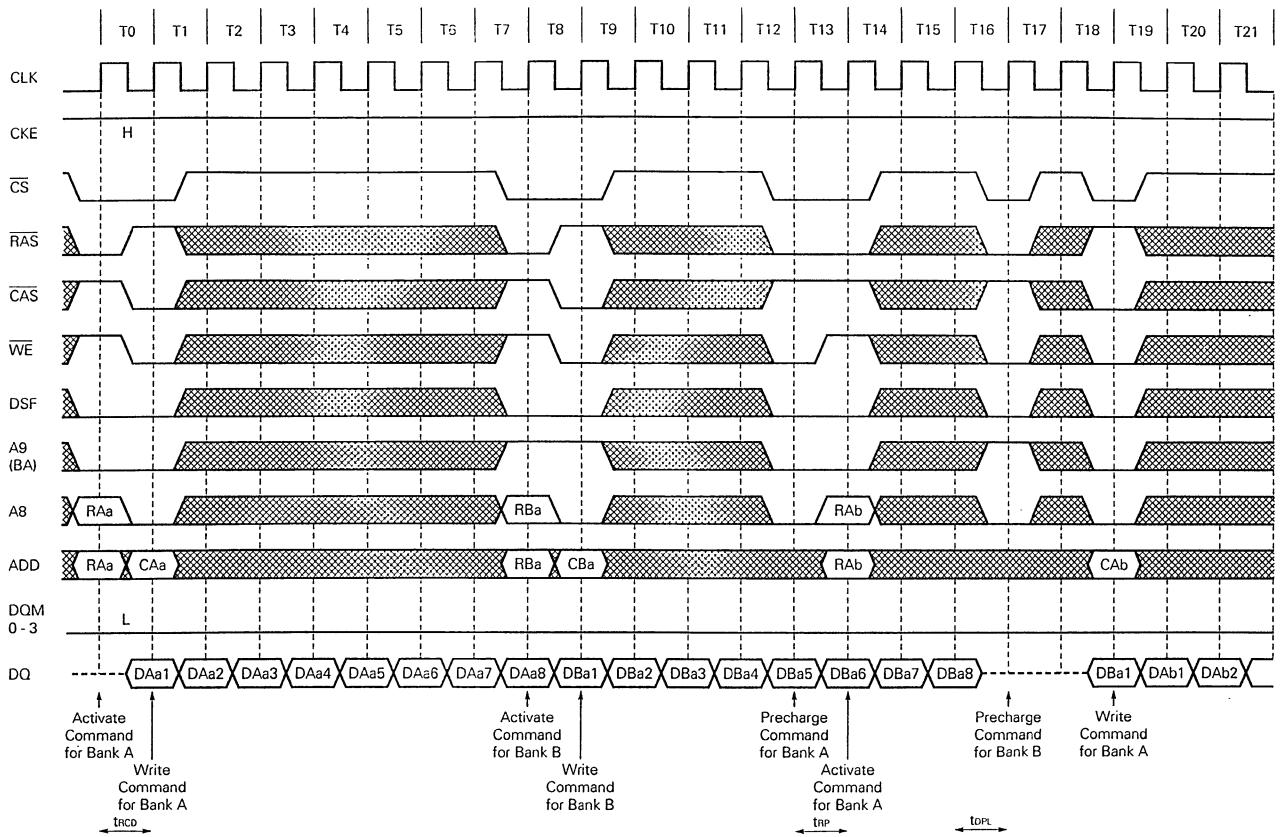
**Random Row Read (Pingpong banks) (2/3)** (Burst length = 8, $\overline{\text{CAS}}$ latency = 2)



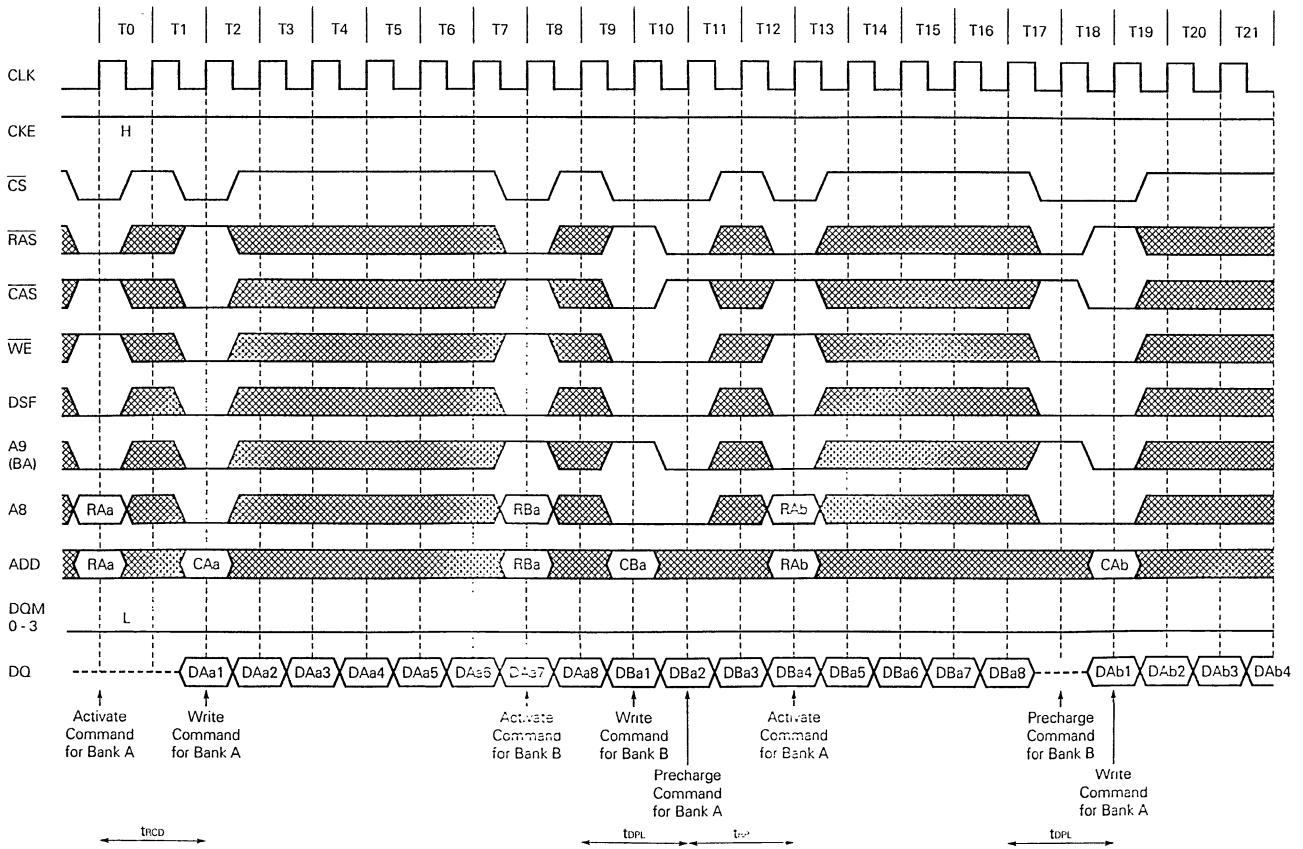
**Random Row Read (Pingpong banks) (3/3)** (Burst length = 8, $\overline{CAS}$ latency = 3)



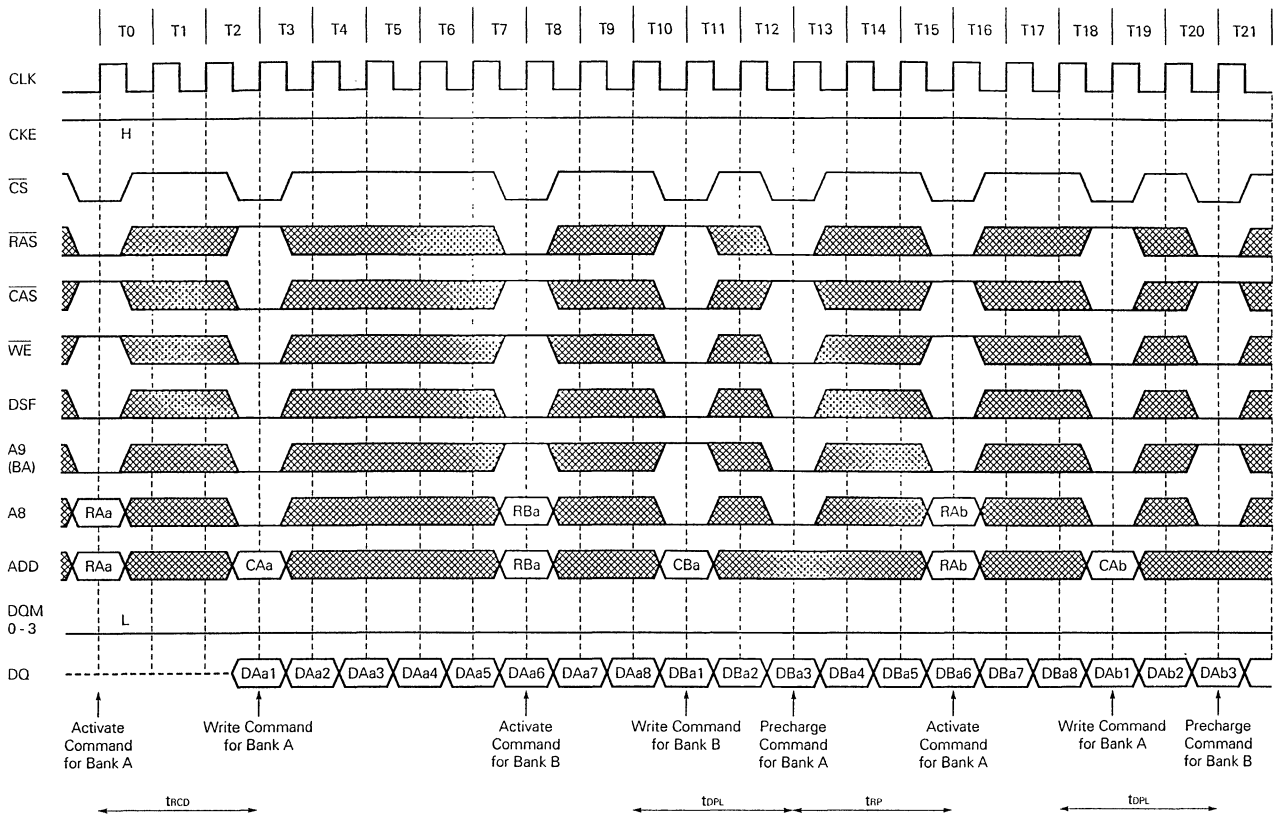
**Random Row Write (Pingpong banks) (1/3)** (Burst length = 8, CAS latency = 1)



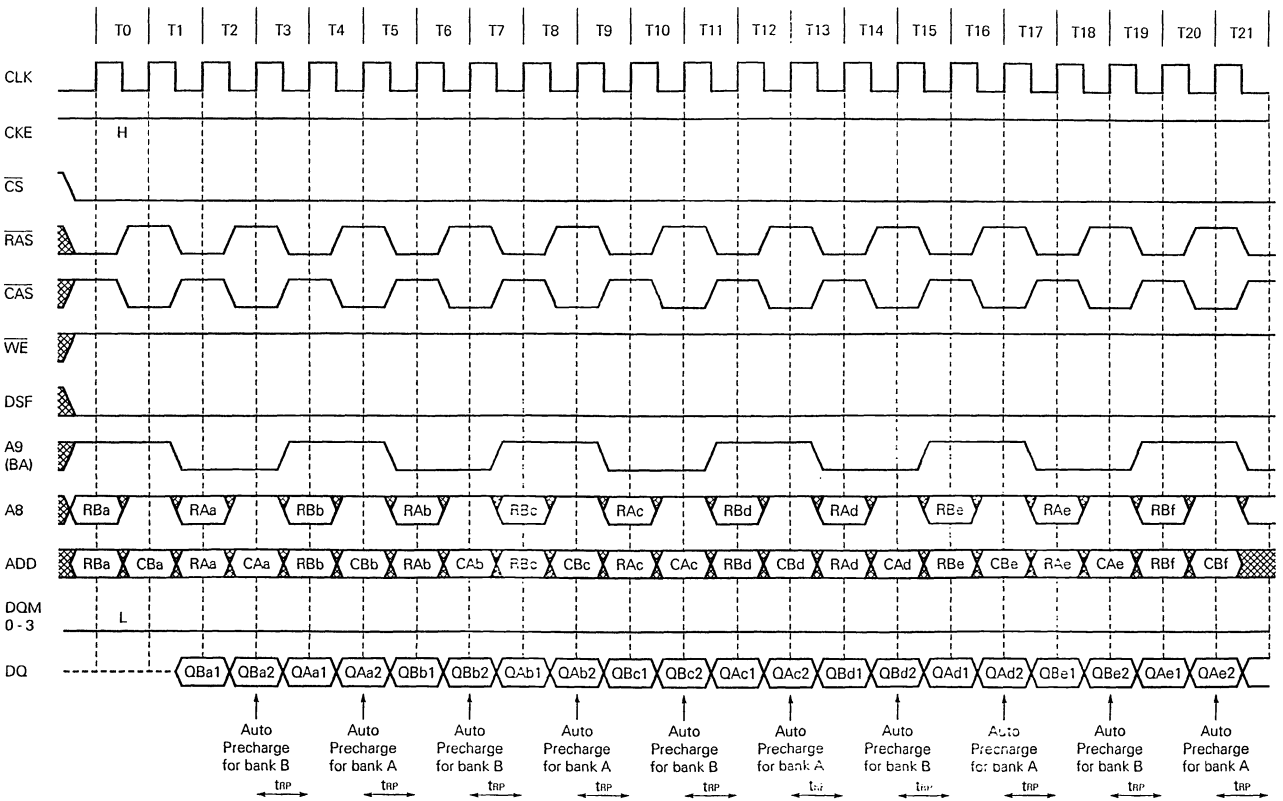
**Random Row Write (Pingpong banks) (2/3)** (Burst length = 8, $\overline{CAS}$ latency = 2)



**Random Row Write (Pingpong banks) (3/3)** (Burst length = 8, $\overline{CAS}$ latency = 3)

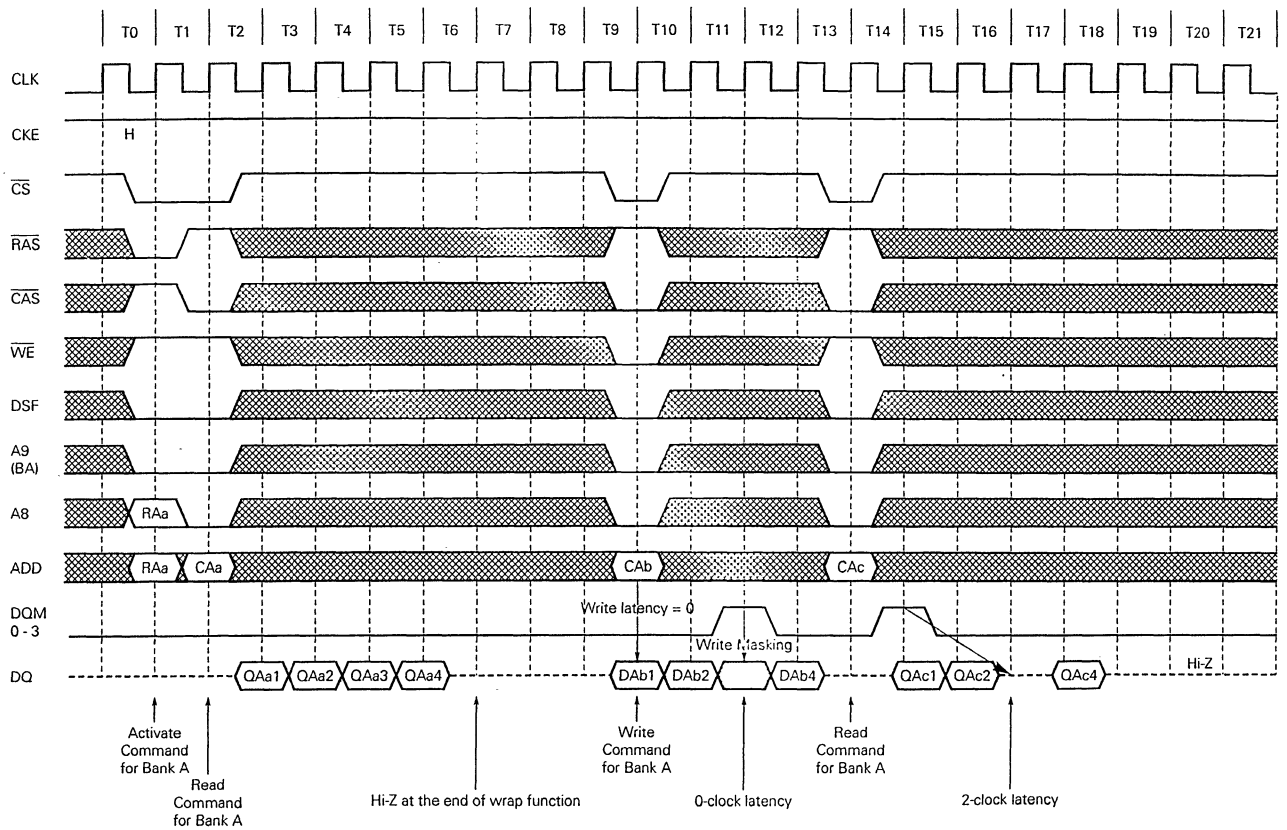


**Random Row READ (Pingpong banks)** (Burst length = 2, $\overline{CAS}$ latency = 1)

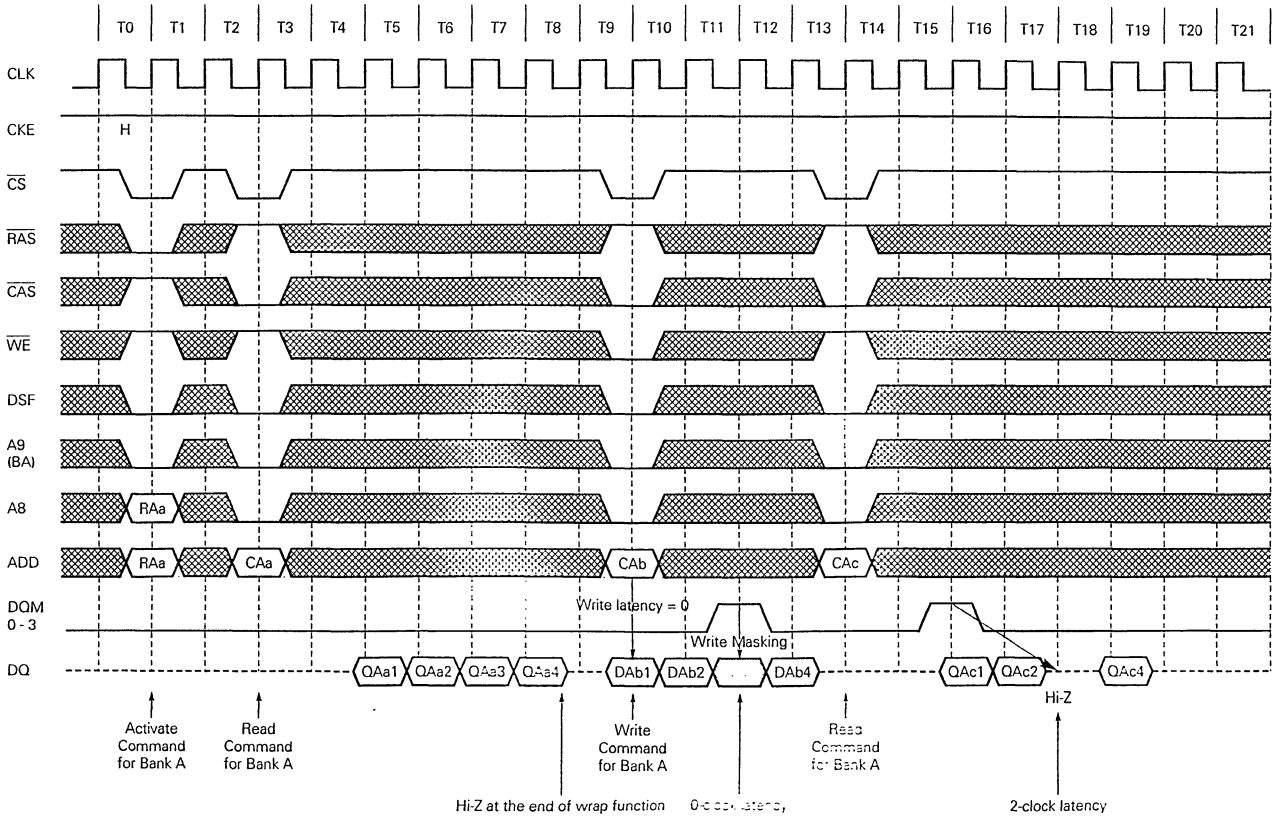


### 16.6.3 READ and WRITE Cycles

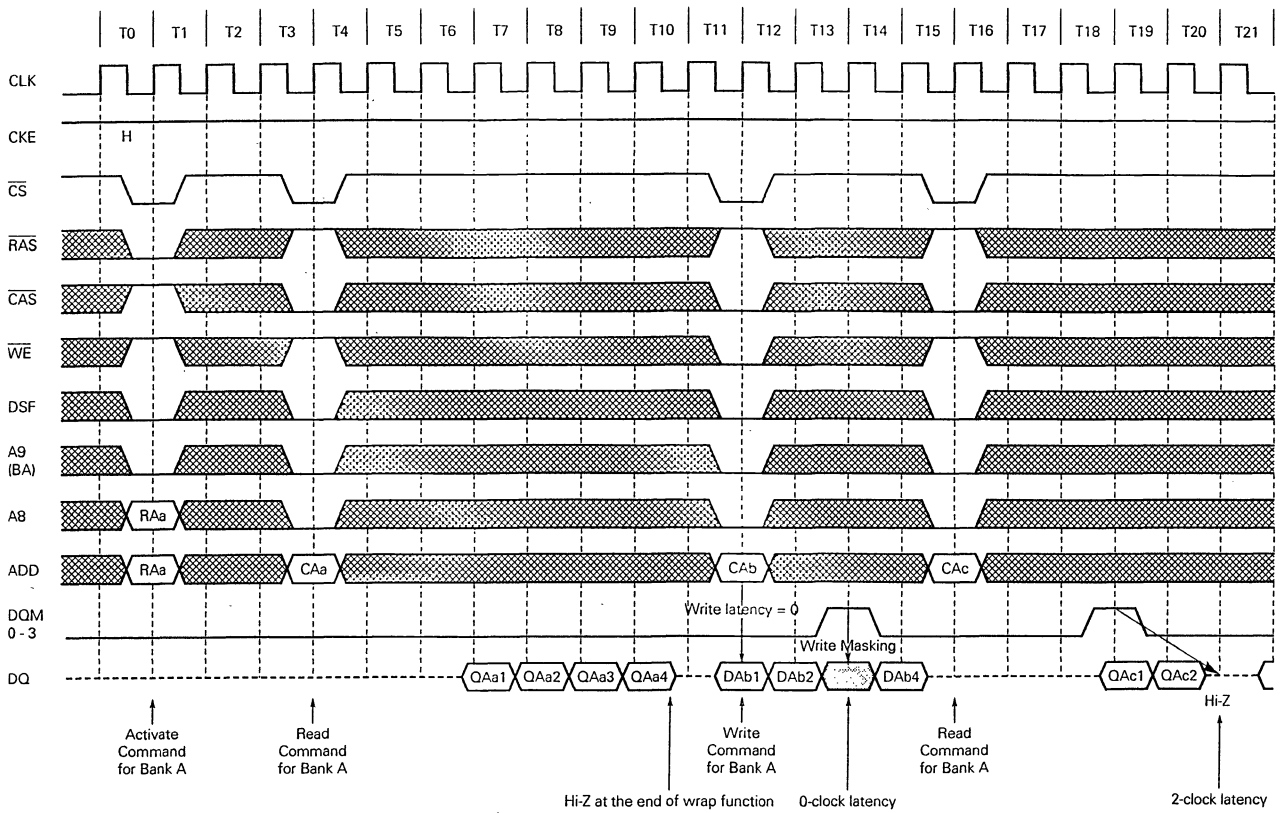
**READ and WRITE (1/3)** (Burst length = 4, $\overline{CAS}$ latency = 1)



**READ and WRITE (2/3)** (Burst length = 4, $\overline{CAS}$ latency = 2)

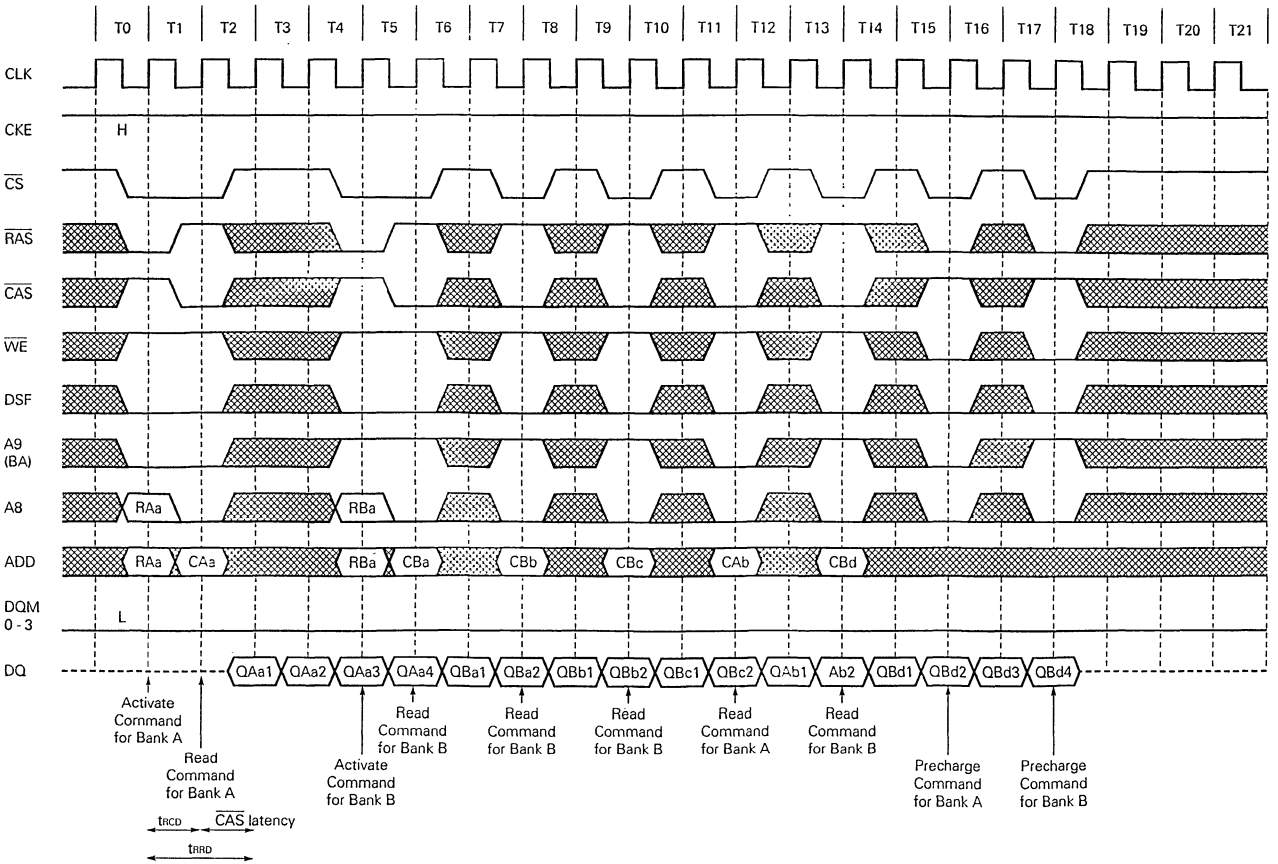


**READ and WRITE (3/3)** (Burst length = 4, $\overline{CAS}$ latency = 3)

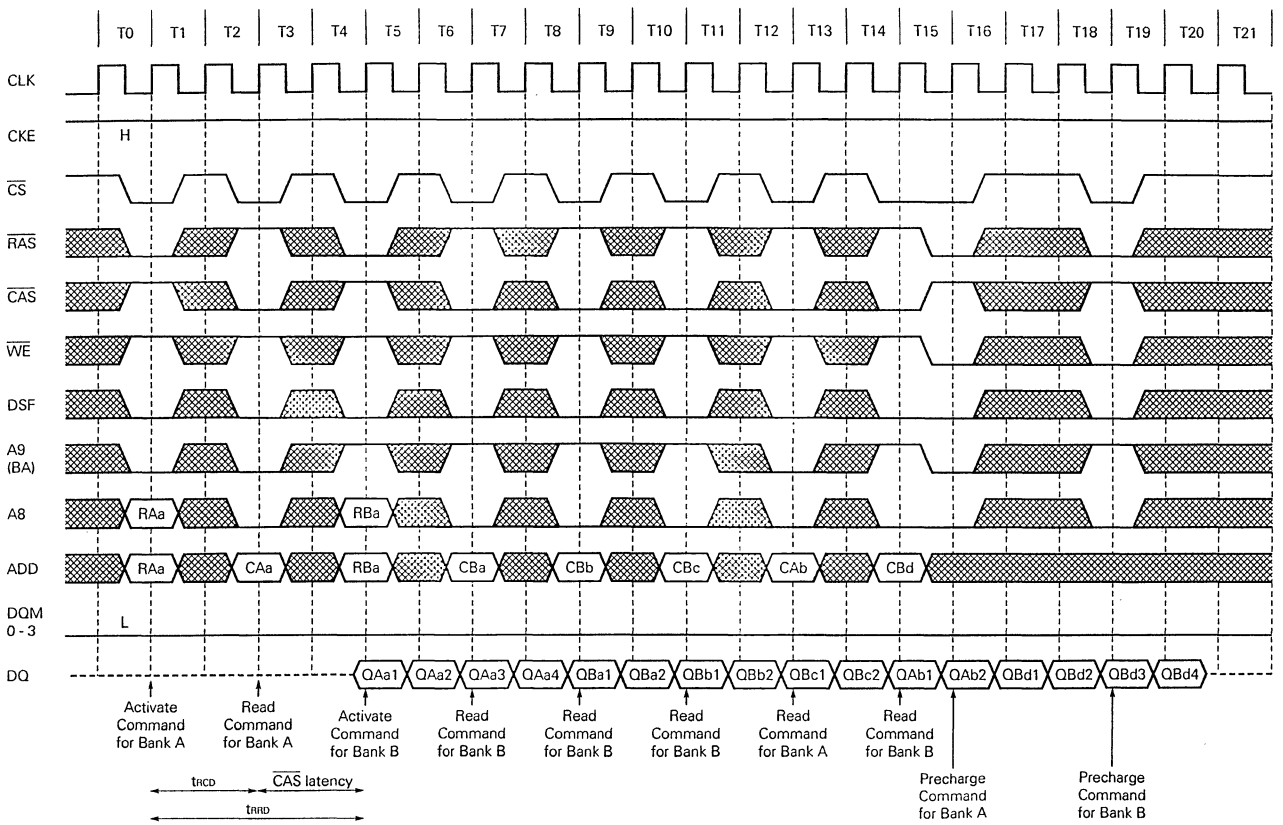


### 16.6.4 Interleaved Cycles

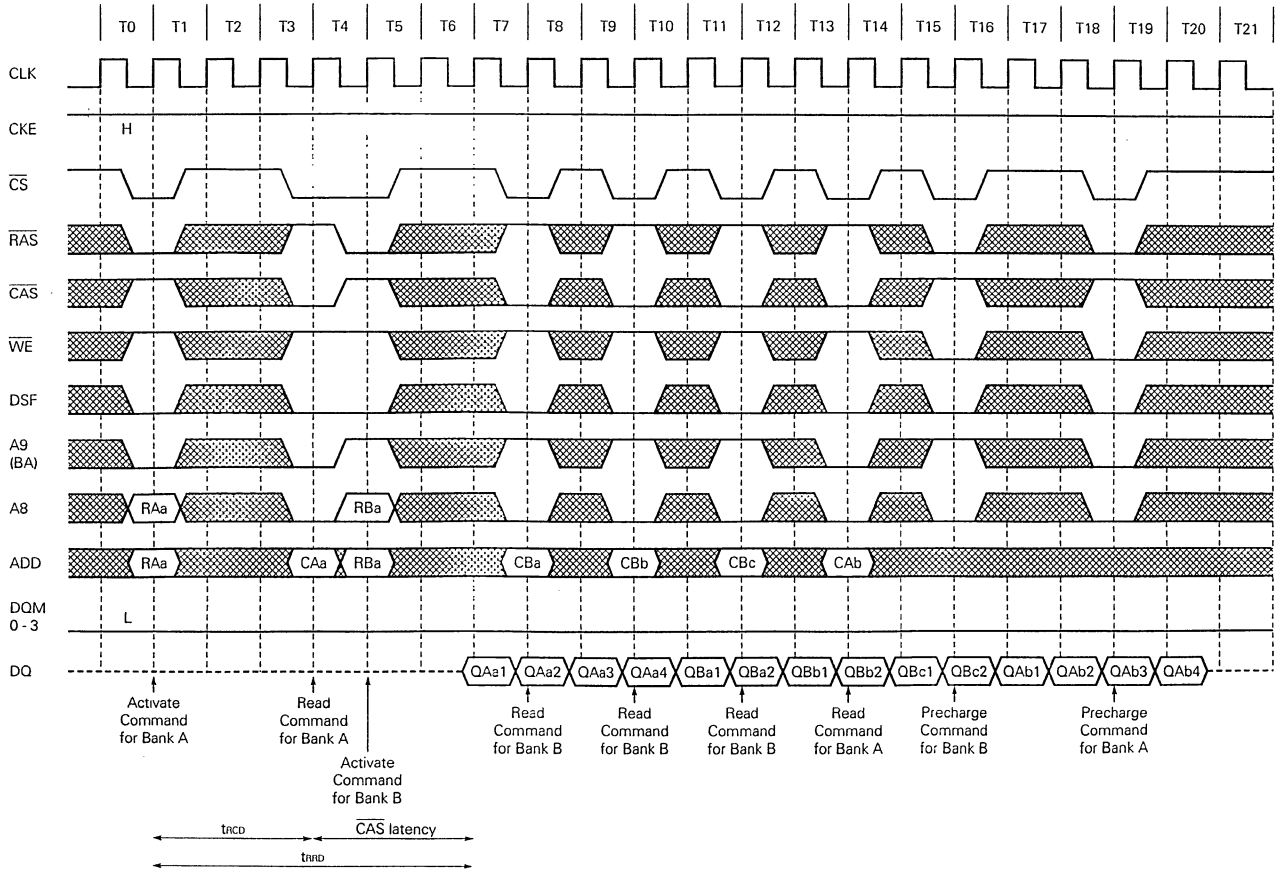
**Interleaved Column READ Cycle (1/3)** (Burst length = 4, $\overline{CAS}$ latency = 1)



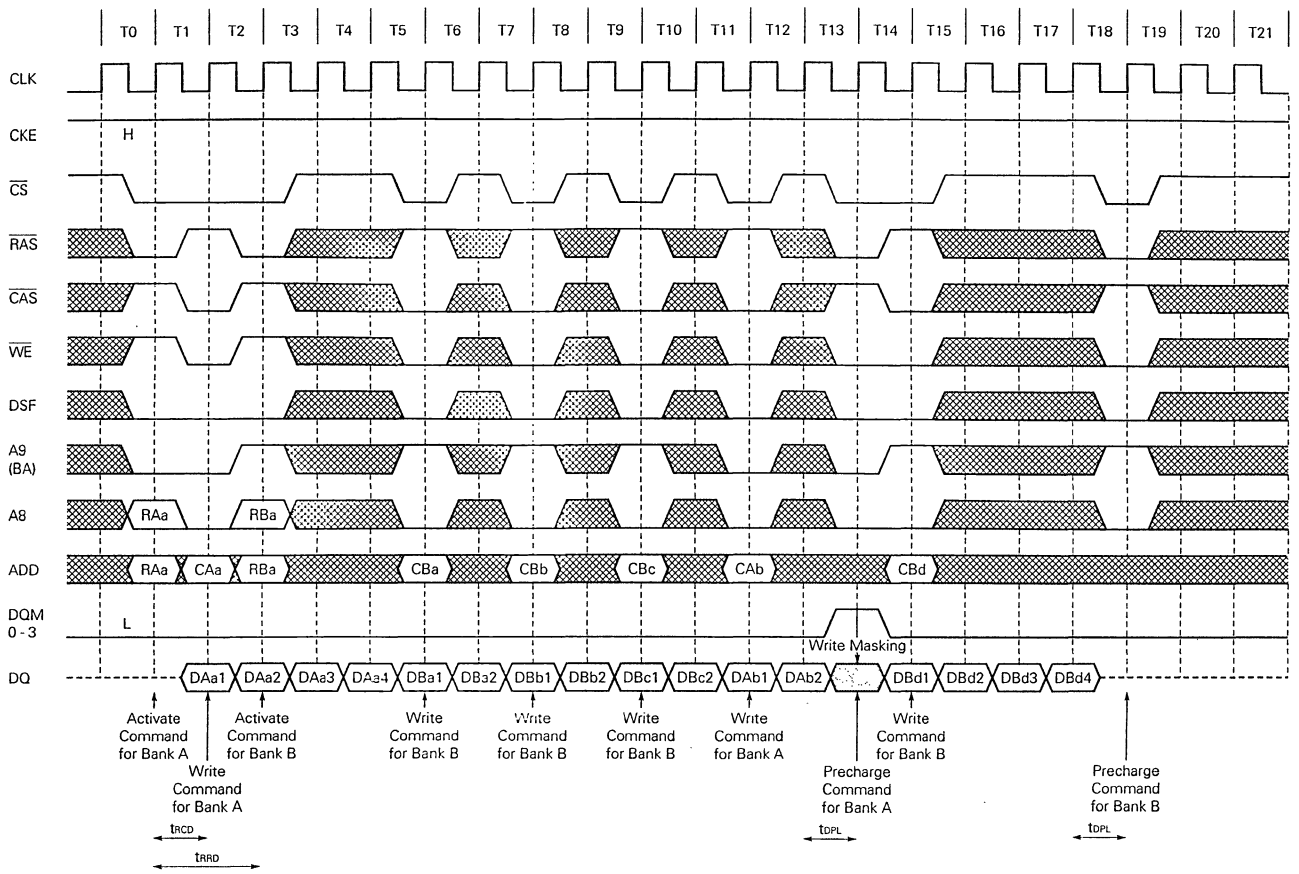
**Interleaved Column READ Cycle (2/3)** (Burst length = 4, $\overline{CAS}$ latency = 2)



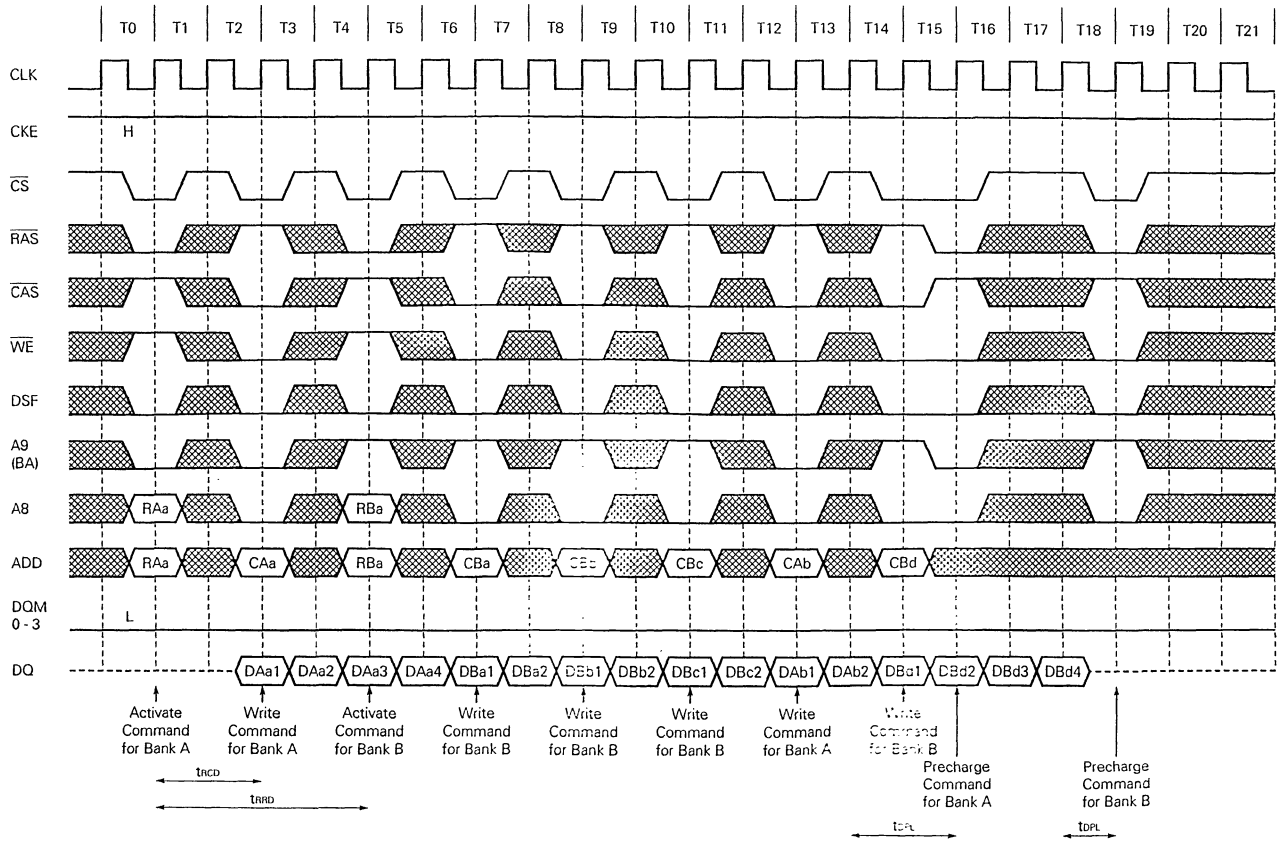
**Interleaved Column READ Cycle (3/3)** (Burst length = 4, $\overline{CAS}$ latency = 3)



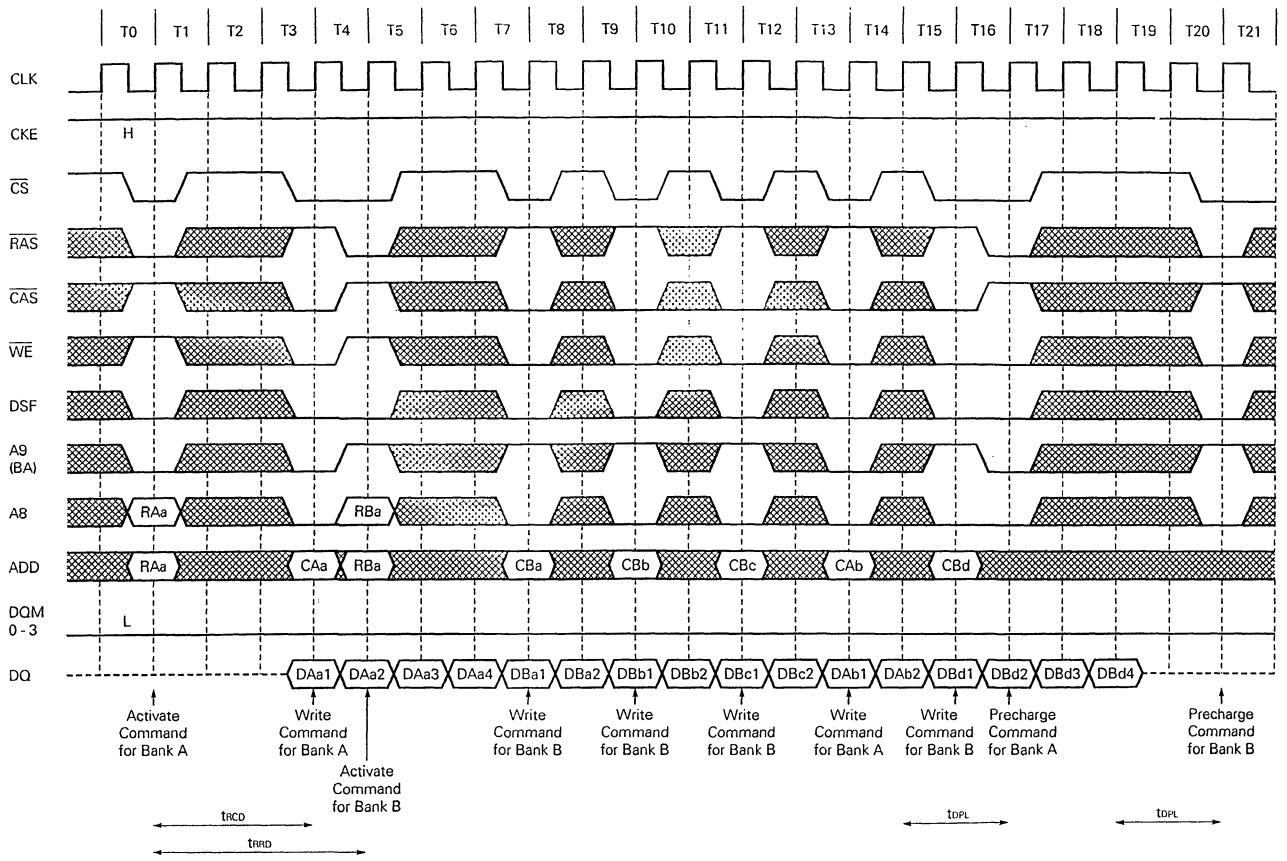
**Interleaved Column WRITE Cycle (1/3)** (Burst length = 4, $\overline{CAS}$ latency = 1)



**Interleaved Column WRITE Cycle (2/3)** (Burst length = 4, $\overline{CAS}$ latency = 2)

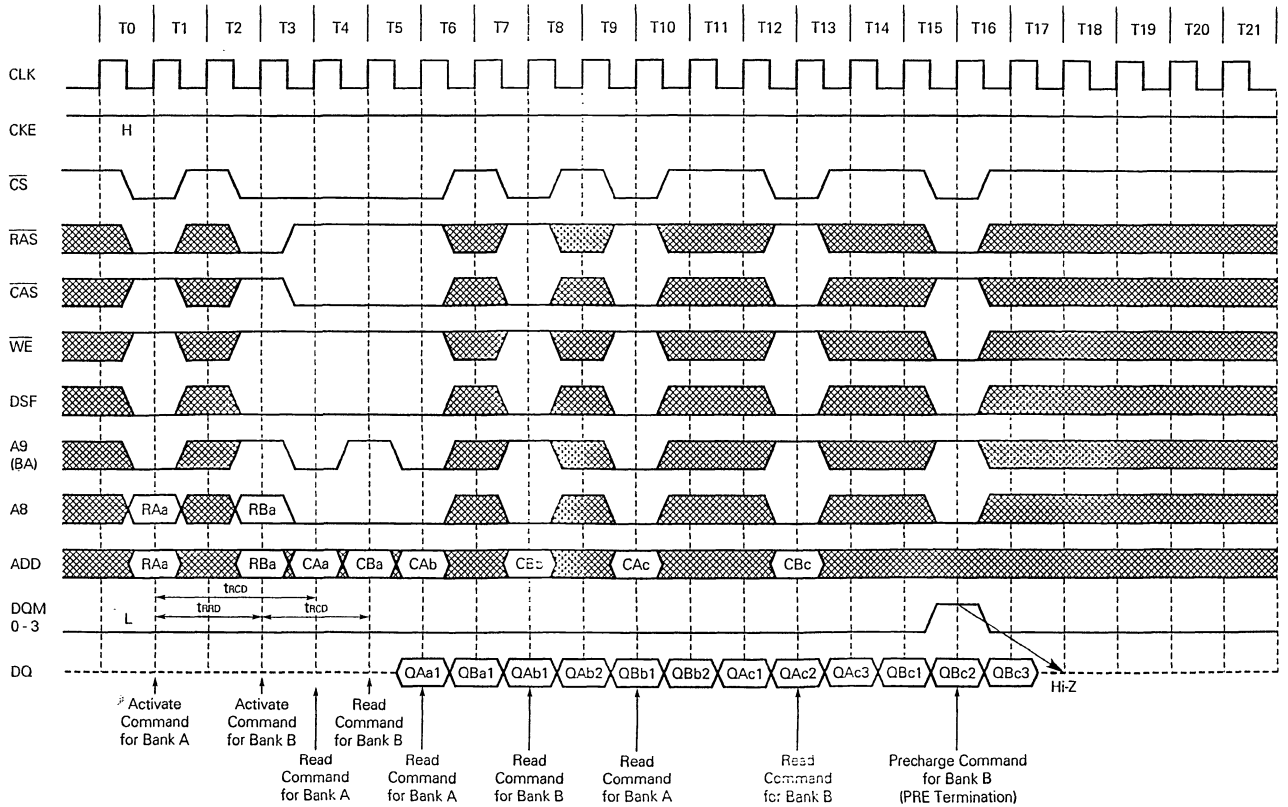


**Interleaved Column WRITE Cycle (3/3)** (Burst length = 4, $\overline{CAS}$ latency = 3)

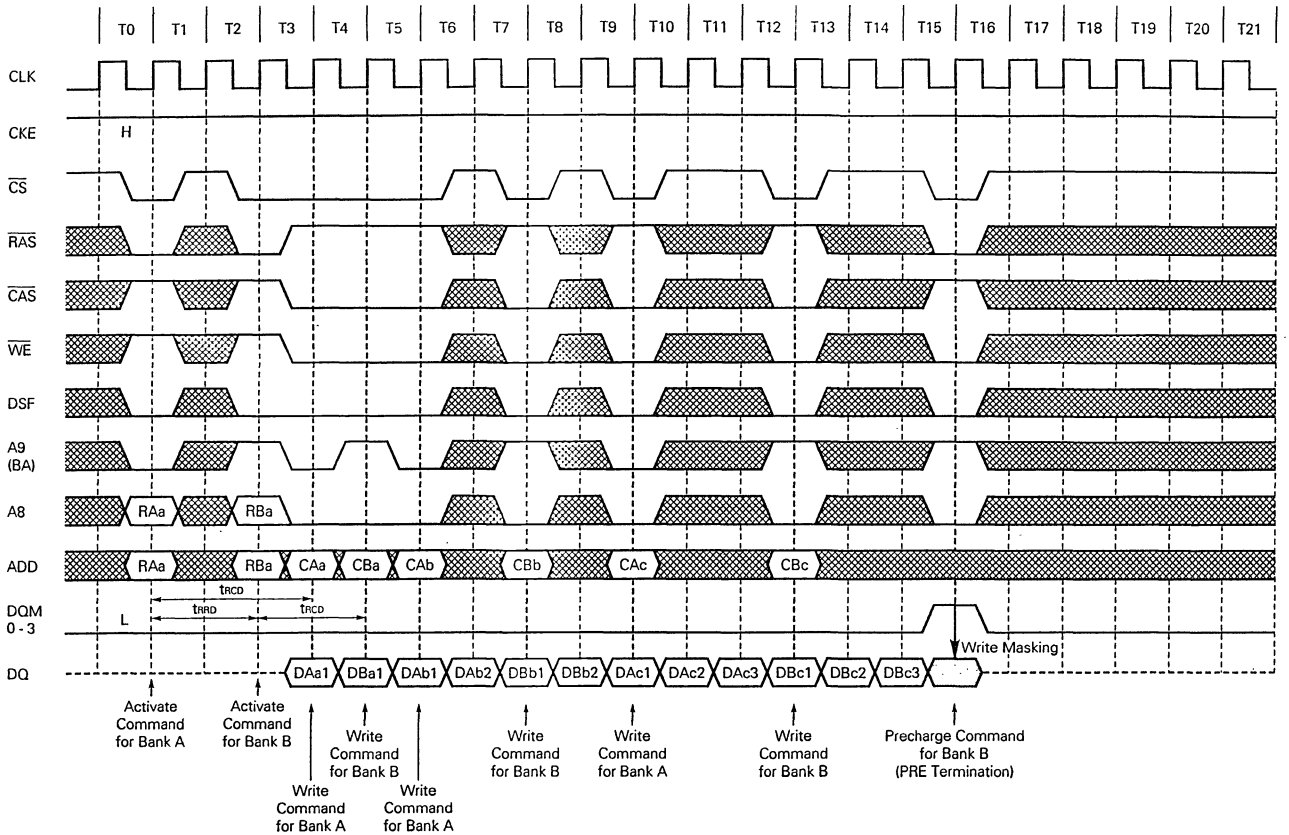


### 16.6.5 Full Page Random Cycles

**Full Page Random Column Read** (Burst length = Full Page, $\overline{CAS}$ latency = 2)

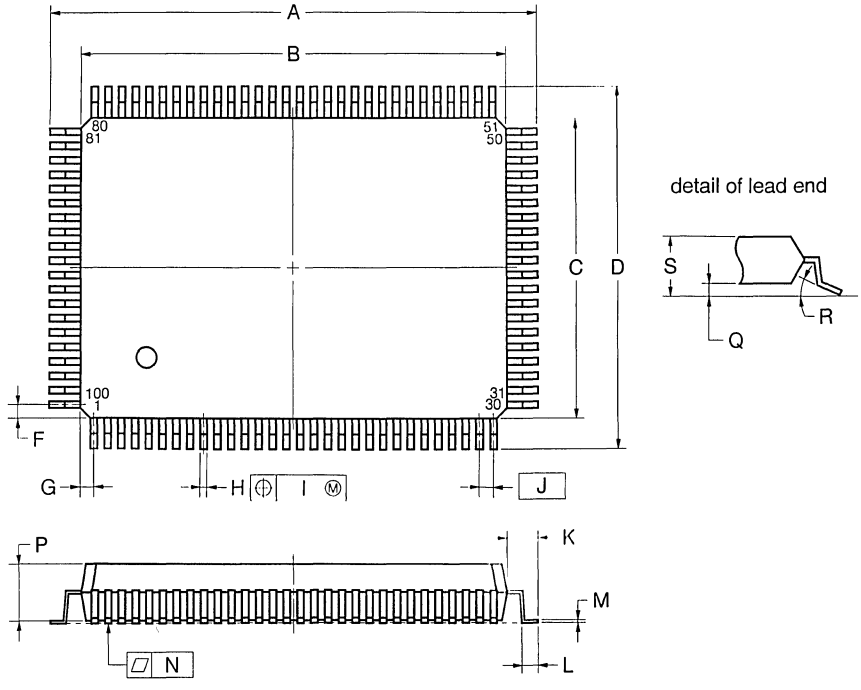


**Full Page Random Column Write** (Burst length = Full Page, $\overline{CAS}$ latency = 2)



## 17. Package Drawing

### 100PIN PLASTIC QFP (14 × 20)




**NOTE**

Each lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 23.2±0.2 | $0.913^{+0.009}_{-0.008}$ |
| B | 20.0±0.2 | $0.787^{+0.009}_{-0.008}$ |
| C | 14.0±0.2 | $0.551^{+0.009}_{-0.008}$ |
| D | 17.2±0.2 | 0.677±0.008 |
| F | 0.825 | 0.032 |
| G | 0.575 | 0.023 |
| H | $0.32^{+0.08}_{-0.07}$ | 0.013±0.003 |
| I | 0.13 | 0.005 |
| J | 0.65 (T.P.) | 0.026 (T.P.) |
| K | 1.6±0.2 | 0.063±0.008 |
| L | 0.8±0.2 | $0.031^{+0.009}_{-0.008}$ |
| M | $0.17^{+0.06}_{-0.05}$ | 0.007±0.002 |
| N | 0.10 | 0.004 |
| P | 2.7 | 0.106 |
| Q | 0.125±0.075 | 0.005±0.003 |
| R | $3°^{+7°}_{-3°}$ | $3°^{+7°}_{-3°}$ |
| S | 3.0 MAX. | 0.119 MAX. |

S100GF-65-JBT

## 18. Recommended Soldering Conditions

Please consult with our sales offices for soldering conditions of the μPD481850.

**Type of Surface Mount Device**

**μPD481850GF-JBT: 100-pin Plastic QFP (14 × 20 mm)**

NEC

NEC Electronics Inc.

Corporate Headquarters
475 Ellis Street
Mountain View, CA 94043
Telephone: 415-960-6000
Fax: 415-965-6130

New Corporate Headquarters
(Spring 1996)
2880 Scott Blvd.
Santa Clara, CA 95050

For literature, call toll-free 8 a.m. to 4 p.m., Pacific time: 1-800-366-9782



Doc. No. M11002EU1V0DB00