128KX8 Bit CMOS Video RAM

FEATURES

- **Dual port Architecture**
  128K x 8 bits RAM port
  256 x 8 bits SAM port
- **Performance**

Parameter \ Speed	-6	-7	-8
RAM access time (t_{RAC})	60ns	70ns	80ns
RAM access time (t_{CAC})	20ns	20ns	20ns
RAM cycle time (t_{RC})	110ns	130ns	150ns
RAM page mode cycle (t_{PC})	40ns	45ns	50ns
SAM access time	18ns	20ns	20ns
SAM cycle time	20ns	25ns	25ns
RAM active current	90mA	85mA	80mA
SAM active current	50mA	45mA	40mA

- **Fast Page Mode**
- **RAM Read, Write, Read-Modify-Write**
- **Serial Read and Serial Write**
- **Read, Real Time Read and Split Read Transfer (RAM→SAM)**
- **Write, Split Write Transfer with Masking operation (New Mask)**
- **Block Write, Flash Write and Write per bit with Masking operation (New Mask)**
- **$\overline{CAS}$-before-$\overline{RAS}$, $\overline{RAS}$-only and Hidden Refresh**
- **Common Data I/O Using three state RAM Output Control**
- **All Inputs and Outputs TTL Compatible**
- **Refresh: 512 Cycle/8ms**
- **Single +5V±10% Supply Voltage**
- **Plastic 40-PIN 400 mil SOJ**

GENERAL DESCRIPTION

The Samsung KM428C128 is a CMOS 128K×8 bit Dual Port DRAM. It consists of a 128K×8 dynamic random access memory (RAM) port and 256×8 static serial access memory (SAM) port. The RAM and SAM ports operate asynchronously except during data transfer between the ports.

The RAM array consists of 512 bit rows of 2048 bits. It operates like a conventional 128K×8 CMOS DRAM. The RAM port has a write per bit mask capability.

The SAM port consists of eight 256 bit high speed shift registers that are connected to the RAM array through a 2048 bit data transfer gate. The SAM port has serial read and write capabilities.

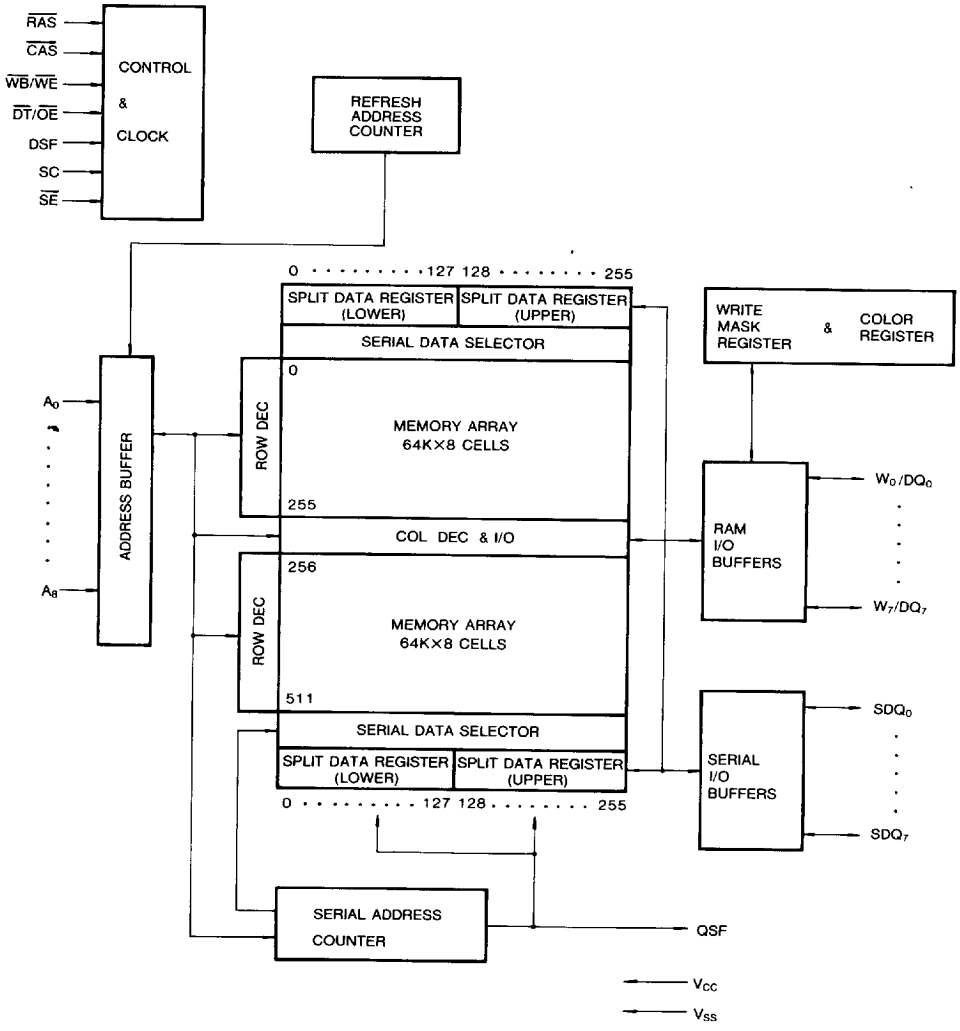
Data may be internally transferred bi-directionally between the RAM and SAM ports using read or write transfers

Refresh is accomplished by familiar DRAM refresh modes. The KM428C128 supports $\overline{RAS}$-only, Hidden, and $\overline{CAS}$-before-$\overline{RAS}$ refresh for the RAM port. The SAM port does not require refresh.

All inputs and I/O's are TTL level compatible. All address lines and Data inputs are latched on chip to simplify system design. The outputs are unlatched to allow greater system flexibility.

Pin Name	Pin Function
SC	Serial Clock
SDQ_0-SDQ_7	Serial Data Input/Output
$\overline{DT}/\overline{OE}$	Data Transfer/Output Enable
$\overline{WB}/\overline{WE}$	Write Per Bit/Write Enable
$\overline{RAS}$	Row Address Strobe
$\overline{CAS}$	Column Address Strobe
DSF	Special Function Control
W_0/DQ_0-W_7/DQ_7	Data Write Mask/Input/Output
$\overline{SE}$	Serial Enable
A_0-A_8	Address Inputs
QSF	Special Flag Output
V_{CC}	Power (+5V)
V_{SS}	Ground
N.C.	No Connection

FUNCTIONAL BLOCK DIAGRAM



DEVICE OPERATION

The KM428C128 contains 1,048,576 memory locations Seventeen address bits are required to address a particular 8-bit word in the memory array. Since the KM428C128 has only 0 address input pins, time multiplexed addressing is used to input 9 row and 8 column addresses. The multiplexing is controlled by the timing relationship between the row address strobe($\overline{RAS}$), the column address strobe($\overline{CAS}$)and the valid row and column address inputs.

Operation of the KM428C128 begins by strobing in a valid row address with $\overline{RAS}$ while $\overline{CAS}$ remains high. Then the address on the 9 address input pins is changed from a row address to a column address and is strobed in by $\overline{CAS}$. This is the beginning of any KM428C128 cycle in which a memory location is accessed. The specific type of cycle is determined by the state of the write enable pin and various timing relationships. The cycle is terminated when both $\overline{RAS}$ and $\overline{CAS}$ have returned to the high state. Another cycle can be initiated after $\overline{RAS}$ remains high long enough to satisfy the $\overline{RAS}$ precharge time (t_{RP}) requirement.

$\overline{RAS}$ and $\overline{CAS}$ Timing

The minimum $\overline{RAS}$ and $\overline{CAS}$ pulse widths are specified by t_{RAS}(min) and t_{CAS}(min) respectively. These minimum pulse widths must be satisfied for proper device operation and data integrity. Once a cycle is initiated by bringing $\overline{RAS}$ low, it must not be aborted prior to satisfying the minimum $\overline{RAS}$ and $\overline{CAS}$ pulse widths. In addition, a new cycle must not begin until the minimum $\overline{RAS}$ precharge time, t_{RP}, has been satisfied. Once a cycle begins, internal clocks and other circuits within the KM428C128 begin a complex sequence of events. If the sequence is broken by violating minimum timing requirements, loss of data integrity can occur.

Read

A read cycle is achieved by maintaining $\overline{WB}/\overline{WE}$ high during a $\overline{RAS}/\overline{CAS}$ cycle. The access time is normally specified with respect to the falling edge of $\overline{RAS}$. But the access time also depends on the falling edge of $\overline{CAS}$ and on the valid column address transition. If $\overline{CAS}$ goes low before t_{RCD}(max) and if the column address is valid before t_{RAD}(max) then the access time to valid data is specified by t_{RAC}(min) However, if $\overline{CAS}$ goes low after t_{RCD}(max) or if the column address becomes valid after t_{RAD}(max), access is specified by t_{CAC} or t_{AA}

The KM428C128 has common data I/O pins The $\overline{DT}/\overline{OE}$ has been provided so the output buffer can be precisely controlled. For data to appear at the outputs, $\overline{DT}/\overline{OE}$ must be low for the period of time defined by t_{OEA}.

Write

The KM428C128 can perform early write and read-modify-write cycles The difference between these cycles is in the state of data-out and is determined by the timing relationship between $\overline{WB}/\overline{WE}$, $\overline{DT}/\overline{OE}$ and $\overline{CAS}$. In any type of write cycle, Data-in must be valid at or before the falling edge of $\overline{WB}/\overline{WE}$, whichever is later.

Fast Page Mode

Fast page mode provides high speed read, write or read-modify-write access to all memory cells within a selected row. These cycles may be mixed in any order. A fast page mode cycle begins with a normal cycle Then, while $\overline{RAS}$ is kept low to maintain the row address, $\overline{CAS}$ is cycled to strobe in additional column addresses This eliminates the time required to set up and strobe sequential row addresses for the same page.

Write-Per-Bit

The write-per-bit function selectively controls the internal write-enable circuits of the RAM port. When $\overline{WB}/\overline{WE}$ is held 'low' at the falling edge of $\overline{RAS}$, during a random access operation, the write-mask is enabled. At the same time, the mask data on the Wi/DQi pins is latched onto the write-mask register (WM1). When a '0' is sensed on any of the Wi/DQi pins, their corresponding write circuits are disabled and new data will not be written

When a '1' is sensed on any of the Wi/DQi pins, their corresponding write circuits will remain enabled so that new data is written The write mask data is valid for only one cycle the truth table of the write-per-bit function are shown in Table 1.

Table 1. Truth table for write-per-bit function

$\overline{RAS}$	$\overline{CAS}$	$\overline{DT}/\overline{OE}$	$\overline{WB}/\overline{WE}$	Wi/DQi	FUNCTION
⎺_	H	H	H	*	WRITE ENABLE
	H	H	L	1	WRITE ENABLE
				0	INHIBIT WRITE

DEVICE OPERATION (Continued)

Block Write

A block write cycle is performed by holding $\overline{CAS}$, $\overline{DT}/\overline{OE}$ "high" and DSF "Low" at the falling edge of $\overline{RAS}$ and by holding DSF "high" at the falling edge of $\overline{CAS}$. The state of the $\overline{WB}/\overline{WE}$ at the falling edge of $\overline{RAS}$ determines whether or not the I/O data mask is enabled as write per bit function. At the falling edge of $\overline{CAS}$, the starting column address pointer and column mask data must be provided. During a block write cycle, the 2 least significant column address(A_0 and A_1) are internally controlled and only the six most significant column address(A_2-A_7)are latched at the falling edge of $\overline{CAS}$.

Flash Write

Flash write is mainly used for fast clear operations in frame buffer applications. A flash write cycle is performed by holding $\overline{CAS}$ "high", $\overline{WB}/\overline{WE}$ "low" and DSF "high" at the falling edge of $\overline{RAS}$. The mask data must also be provided on the Wi/DQi lines at the falling edge of $\overline{RAS}$ in order to enable the flash write operation for selected I/O blocks.

Data Output

The KM428C128 has a three-state output buffers which are controlled by $\overline{CAS}$ and $\overline{DT}/\overline{OE}$. When either $\overline{CAS}$ or $\overline{DT}/\overline{OE}$ is high (V_{IH}) the output is in the high impedance (Hi-Z) state. In any cycle in which valid data appears at the output the output goes into the low impedance state in a time specified by t_{CLZ} after the falling edge of $\overline{CAS}$. Invalid data may be present at the output during the time after t_{CLZ} and before the valid data appears at the output. The timing parameters t_{CAC}, t_{RAC} and t_{AA} specify when the valid data will be present at the output. The valid data remains at the output until $\overline{CAS}$ returns high. This is true even if a new $\overline{RAS}$ cycle occurs (as in hidden refresh). Each of the KM428C128 operating cycles is listed below after the corresponding output state produced by the cycle.

Valid Output Data: Read, Read-Modity-Write, Hidden Refresh, Fast Page Mode Read, Fast Page Mode Read-Modify-Write, Read Color Register.

Refresh

The data in the KM428C128 is stored on a tiny capacitor within each memory cell. Due to leakage the data may leak off after a period of time. To maintain data integrity it is necessary to refresh each of the 512 rows every 8 ms. Any operation cycle performed in the RAM port refreshes the 2048 bits selected by the row addresses or an on-chip refresh address counter. Either a burst refresh or distributed refresh may be used. There are several ways to accomplish this.

$\overline{RAS}$-Only Refresh: This is the most common method for performing refresh. It is performed by strobing in a row address with $\overline{RAS}$ while $\overline{CAS}$ remains high. This cycle must be repeated for each of the 512 row address, (A_0-A_8).

$\overline{CAS}$ before-$\overline{RAS}$ Refresh: The KM428C128 has $\overline{CAS}$-before-$\overline{RAS}$ on-chip refresh capability that eliminates the need for external refresh addresses. If $\overline{CAS}$ is held low for the specified set up time(t_{CSR}) before $\overline{RAS}$ goes low the on-chip refresh circuitry is enabled. An internal refresh operation automatically occurs. The refresh address is supplied by the on-chip refresh address counter which is then internally incremented in preparation for the next $\overline{CAS}$-before-$\overline{RAS}$ refresh cycle.

Hidden Refresh: A hidden refresh cycle may be performed while maintaining the latest valid data at the output by extending the $\overline{CAS}$ active time and cycling $\overline{RAS}$. The KM428C128 hidden refresh cycle is actually a $\overline{CAS}$-before-$\overline{RAS}$ refresh cycle within an extended read cycle. The refresh row address is provided by the on-chip refresh address counter.

Other Refresh Methods: It is also possible to refresh the KM428C128 by using read, write or read-modify-write cycles. Whenever a row is accessed, all the cells in that row are automatically refreshed. There are certain applications in which it might be advantageous to perform refresh in this manner but in general $\overline{RAS}$-only refresh or $\overline{CAS}$-before-$\overline{RAS}$ refresh is the preferred method.

Transfer Operation

1. Normal Write/Read Transfer (SAM→RAM/RAM→SAM.).
2. Pseudo Write Transfer (Switches serial port from serial Read to serial Write. No actual data transfer takes place between the RAM and the SAM.).
3. Real Time Read Transfer (On the fly Read Transfer operation).
4. Split Write/Read Transfer (Divides the SAM into a high and a low half. Only one half is transferred from/to the SAM while the other half is write to/read from the SDQ pins.).

Read-Transfer Cycle

A read-transfer consists of loading a selected row of data from the RAM array into the SAM register. A read-transfer is accomplished by holding $\overline{CAS}$ high, $\overline{DT}/\overline{OE}$ low and $\overline{WB}/\overline{WE}$ high at the falling edge of $\overline{RAS}$. The row address

DEVICE OPERATION (Continued)

selected at the falling edge of $\overline{RAS}$ determines the RAM row to be trasferred into the SAM

The actual data transfer completed at the rising edge of $\overline{DT}/\overline{OE}$. When the transfer is completed, the SDQ lines are set into the otuput mode. In a read/real-time read-transfer cycle, the transfer of a new row of data is completed at the rising edge of $\overline{DT}/\overline{OE}$ and becomes valid on the $\overline{SDQ}$ lines after the specified access time t_{SCA} from the rising edge of the subsequent serial clock (SC) cycle. The start address of the serial pointer of the SAM is determined by the column address selected at the falling edge of $\overline{CAS}$.

Table 2. Truth table for Transfer operation

$\overline{RAS}$ Falling Edge					Function	Transfer Direction	Transfer Data Bits	Sam port Mode
$\overline{CAS}$	$\overline{DT}/\overline{OE}$	$\overline{WB}/\overline{WE}$	$\overline{SE}$	DSF				
H	L	H	*	L	Read Transfer	RAM→SAM	256×8	Input→Output
H	L	L	L	L	Masked Write Transfer	SAM→RAM	256×8	Output→Input
H	L	L	H	L	Pseudo Write Transfer	—	—	Output→Input
H	L	H	*	H	Split Read Transfer	RAM→SAM	128×8	Not Changed
H	L	L	*	H	Split Write Transfer	SAM→RAM	128×8	Not Changed

*: Don't Care

Write Transfer Cycle

A write transfer cycle consists of loading the content of the SAM data register into a selected row of RAM array A write transfer is accomplished by $\overline{CAS}$ high, $\overline{DT}/\overline{OE}$ low, $\overline{WB}/\overline{WE}$ low and $\overline{SE}$ low at the falling edge of $\overline{RAS}$. The row address selected at the falling edge of $\overline{RAS}$ determines the RAM row address into which the data will be transfered. The column address selected at the falling edge of $\overline{CAS}$ determines the start address of the serial pointer of the SAM. After the write transfer is completed, the SDQ lines are in the input mode so that serial data synchronized with SC can be loaded. When two consecutive write transfer operations are performed, there is a delay in availability between the last bit of the previous row and the first bit of the new row. Consequently the SC clock must be held at a constant V_{IL} or V_{IH} after the SC precharge time t_{SCP} has seen satisfied, a rising edge of the SC clock until after a specified delay t_{SRD} from the rising edge of $\overline{RAS}$.

Pseudo Write Transfer Cycle

The pseudo write transfer cycle switches SDQ lines from serial read mode to serial write mode. It doesn't perform data transfer. A psuedo write transfer is accomplished by holding $\overline{CAS}$ high, $\overline{DT}/\overline{OE}$ low, $\overline{WB}/\overline{WE}$ low and $\overline{SE}$ high at the falling edge of $\overline{RAS}$. The pseudo write transfer cycle must be performed after a read transfer cycle if the subsequent operation is a write transfer cycle. There is a timing delay associated with the switching of the SDQ lines from serial output mode to serial input mode. During this period, the SC clock must be held at a constant V_{IL} or V_{IH} after the t_{SC} precharge time has been satisfied. A rising edge of the SC clock must not occur until after the specified delay t_{SRD} from the rising edge of $\overline{RAS}$.

Special Function Input (DSF)

In read transfer mode, holding DSF high on the falling edge of $\overline{RAS}$ selects the split register mode read transfer operation. This mode divides the serial data register into a high order half and a low order half; one active, and one inactive. When the cycle is initiated, a transfer occurs between the memory array and either the high half or the low half register depending on the state of most significant column address bit(A_7) that is strobed in on the falling edge of $\overline{CAS}$. If A_7 is high, the transfer is to the high half of the register. If A_7 is low, the transfer is to the low half of the register. Use of the split register mode read transfer feature allows on-the-fly read transfer operation without synchronizing $\overline{DT}/\overline{OE}$ to the serial clock. The transfer can be to either the active half or the inactive half register. If the transfer is to the active register, with an uninterrupted serial data stream, then the timings t_{TSL} and t_{TSD} must be met.

In write tranfer mode, holding DSF high on the falling edge of $\overline{RAS}$ permits use of a Split Register mode of transfer write This mode allows $\overline{SE}$ to be high on the falling edge of $\overline{RAS}$ without performing a pseudo write transfer, with the serial port disabled during the entire transfer write cycle.

DEVICE OPERATION (Continued)

Split Register Active Status Output (QSF)

QSF indicates which half of the serial register in the SAM is being accessed. If QSF is low, then the serial address pointer is accessing the low (least significant) 128 bits of the SAM. If QSF is high, then the pointer is accessing the higher (most significant) 128 bits of the SAM.

Serial Clock (SC)

All operations of the SAM port are synchronized with the serial clock SC. Data is shifted in or out of the SAM registers at the rising edge of SC. In a serial read, the output data becomes valid on the SDQ pins after the maximum specified serial access time t_{SCA} from the rising edge of SC. The serial clock SC also increments the 8 bit serial pointer which is used to select the SAM address. The pointer address is incremented in a wrap around mode to select sequential locations after the starting location which is determined by the column address in the read transfer cycle.

Serial Input/Output (SDQ_0-SDQ_7)

Serial input and serial output share common I/O pins. Serial input or output mode is determined by the most recent transfer cycle. When a read transfer cycle is performed, the SAM port is in the output mode. When a pseudo write is performed, the SAM port operation is switched from output mode to input mode. During subsequent write transfer cycle, the SAM port remains in the input mode.

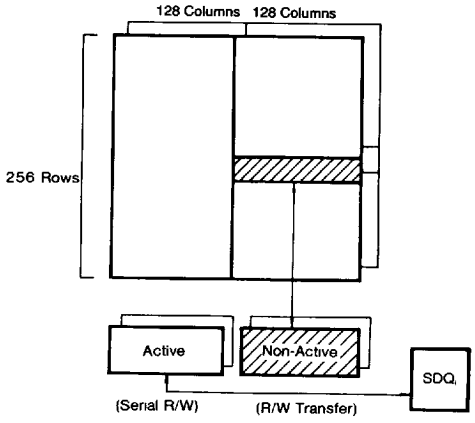
Tap Address Limitation

The Tap Address of non-split transfer cycle preceding split transfer cycle should be between 0 and 125 or between 128 and 253.

Power-up

An initial pause of 200 μsec is required after power-up followed by 8 initialization cycles before proper device operation is assured.

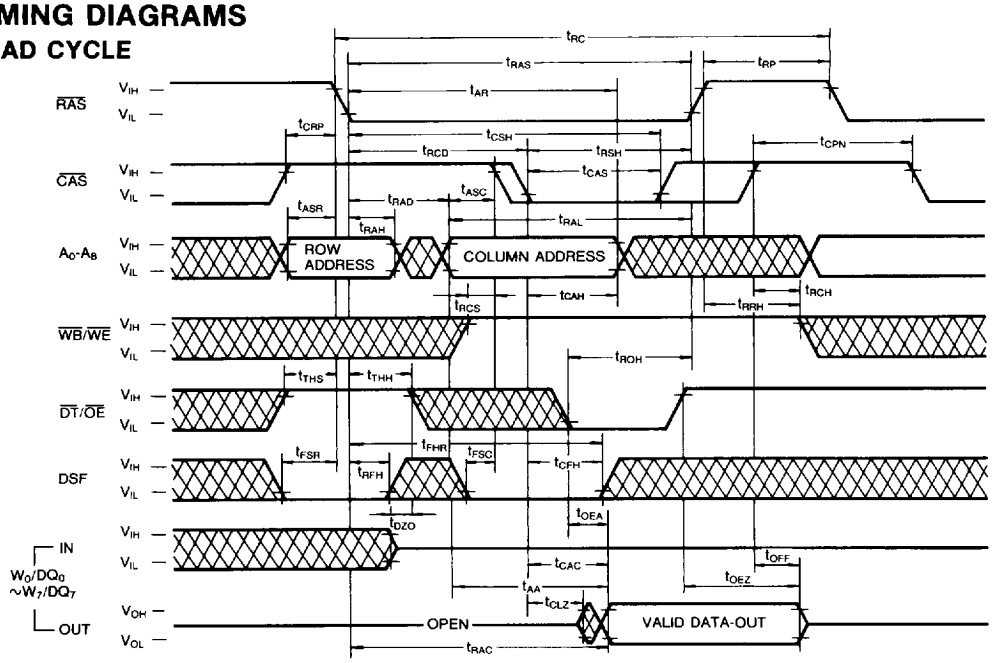
Table 3. SPLIT REGISTER MODE



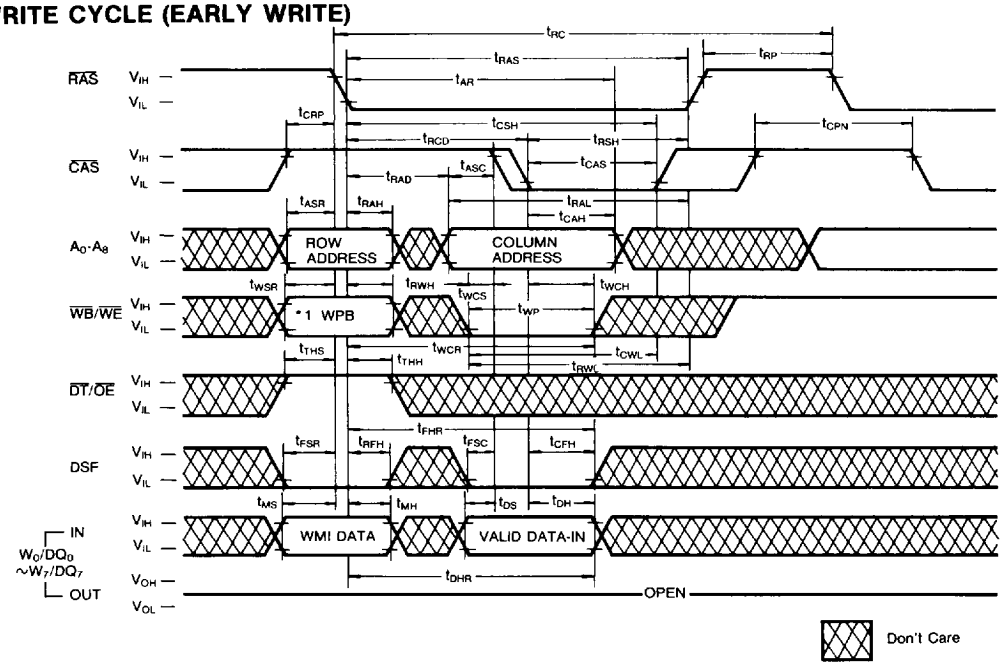
Active SAM	QSF Level
LOWER SAM	LOW
UPPER SAM	HIGH

TIMING DIAGRAMS

READ CYCLE

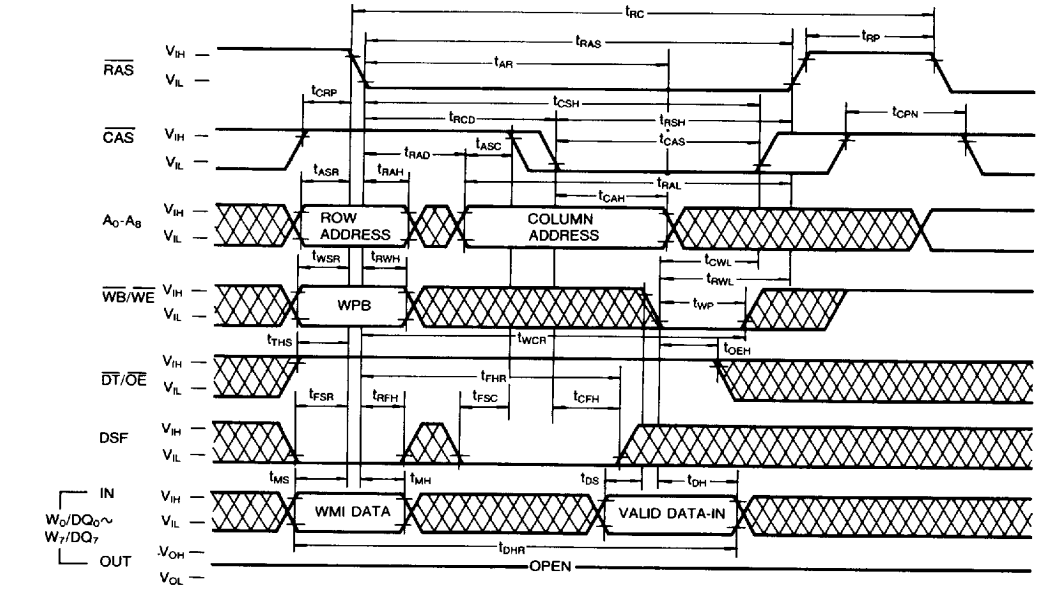


WRITE CYCLE (EARLY WRITE)

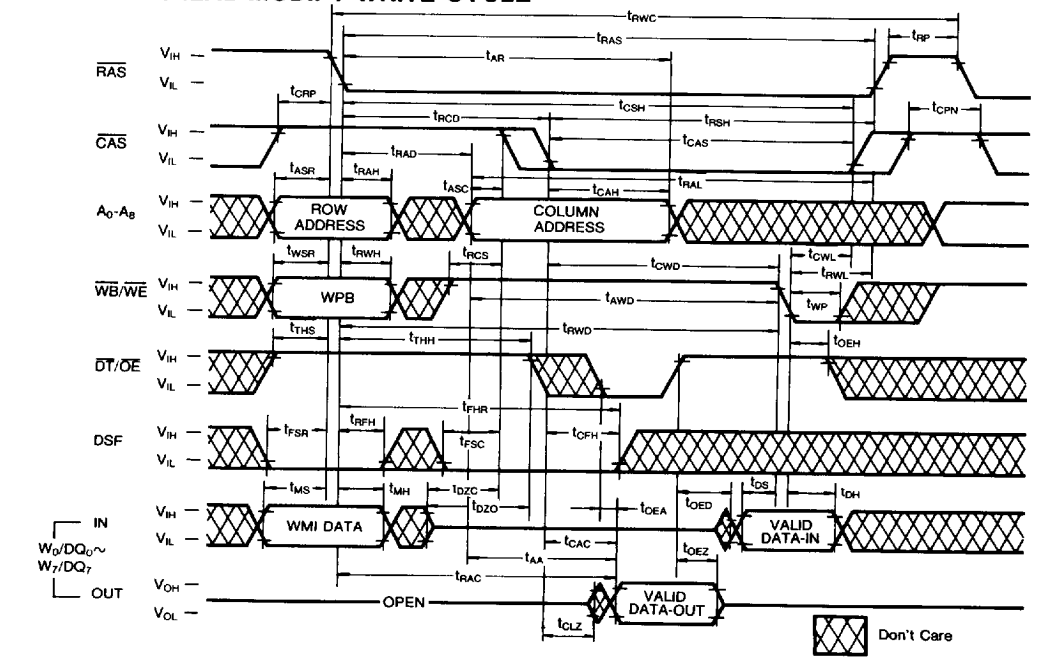


TIMING DIAGRAMS (Continued)

WRITE CYCLE ($\overline{OE}$ CONTROLLED WRITE)

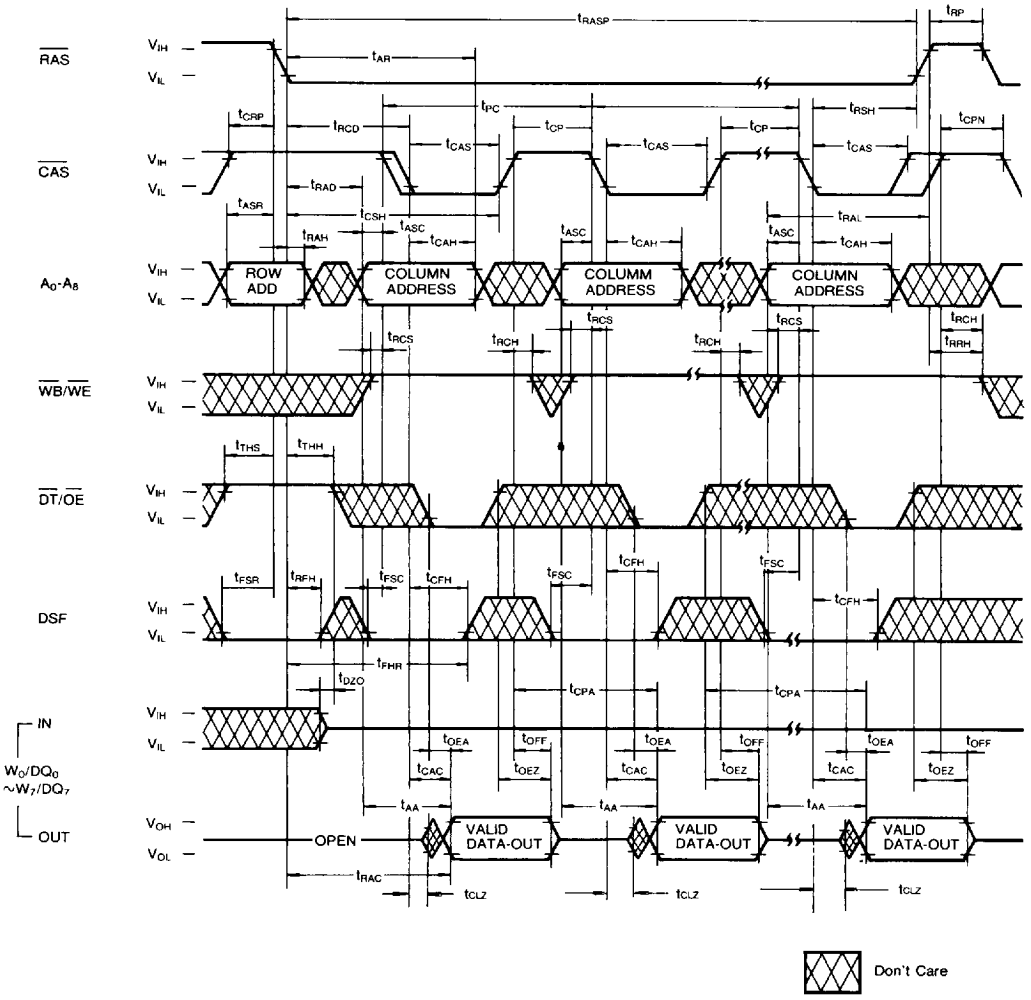


READ-WRITE/READ-MODIFY-WRITE CYCLE



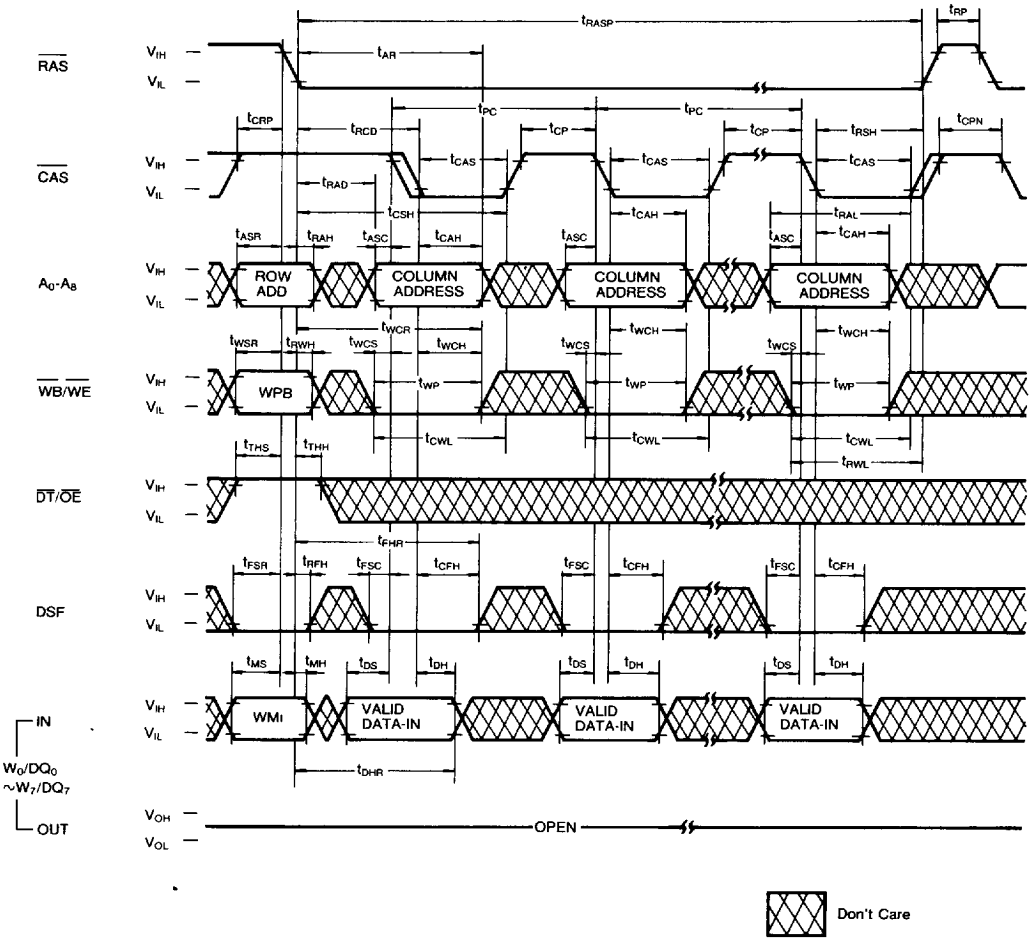
TIMING DIAGRAMS (Continued)

PAGE MODE READ CYCLE



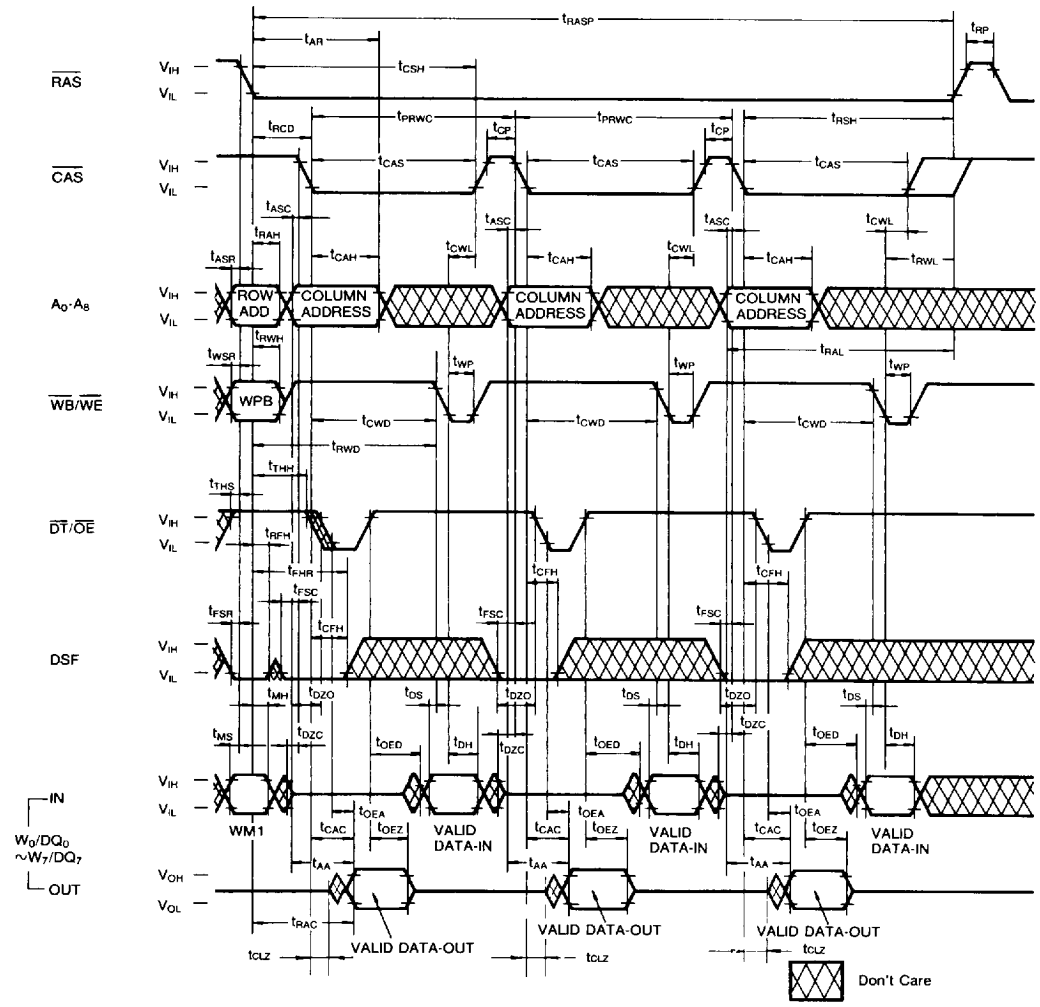
TIMING DIAGRAMS (Continued)

PAGE MODE WRITE CYCLE (EARLY WRITE)



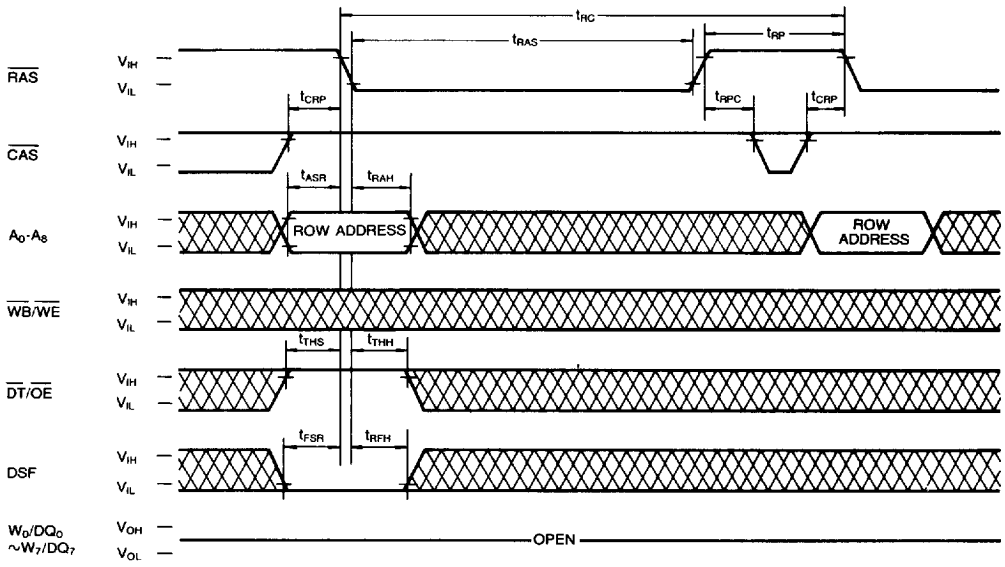
TIMING DIAGRAMS (Continued)

PAGE MODE READ-MODIFY-WRITE CYCLE

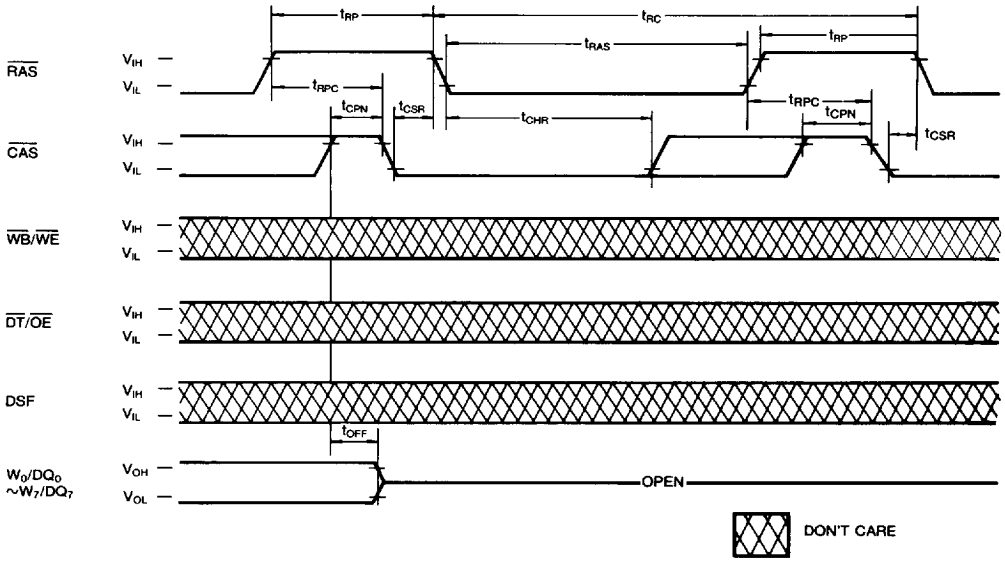


TIMING DIAGRAMS (Continued)

$\overline{RAS}$ ONLY REFRESH CYCLE

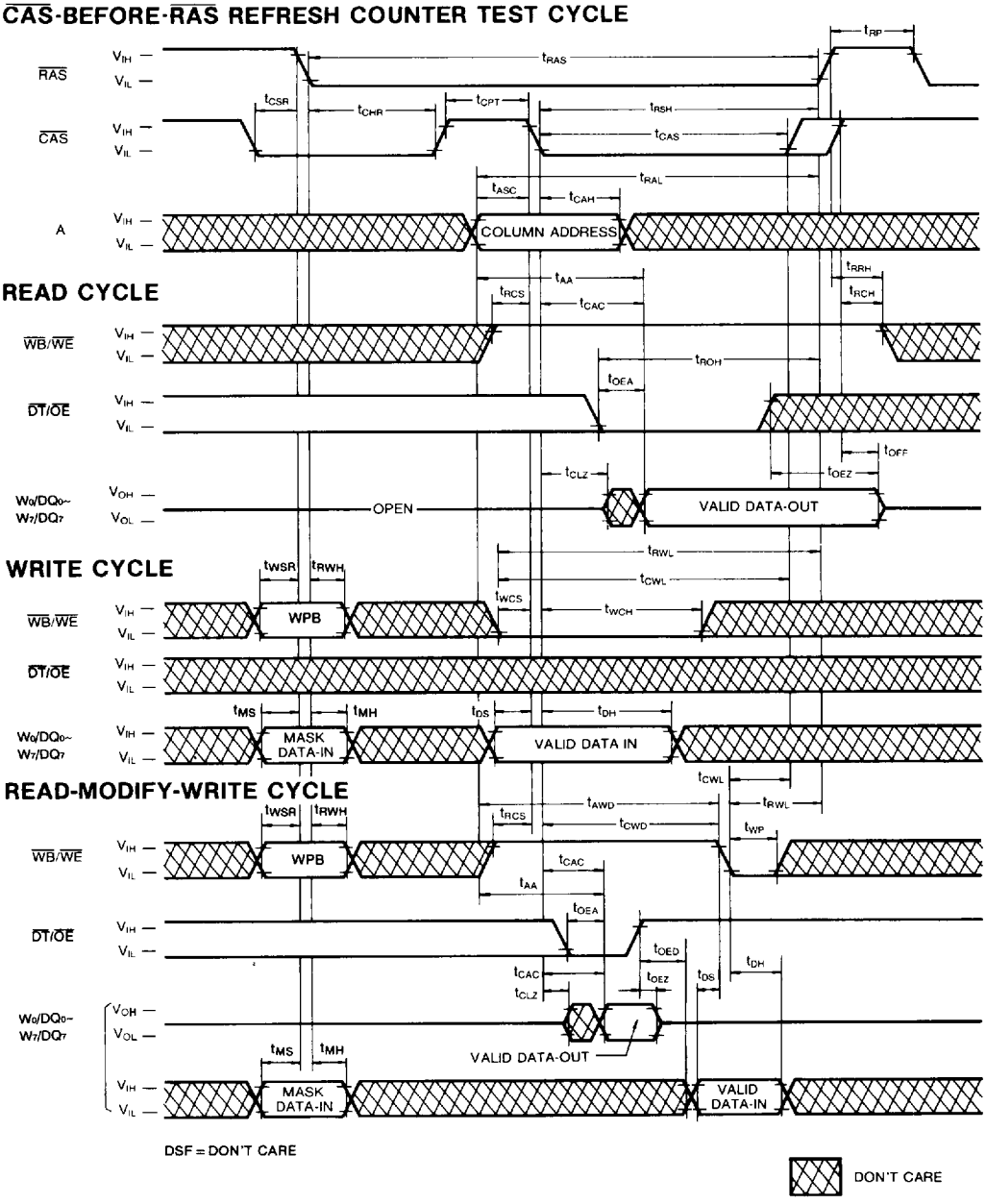


$\overline{CAS}$ BEFORE $\overline{RAS}$ REFRESH



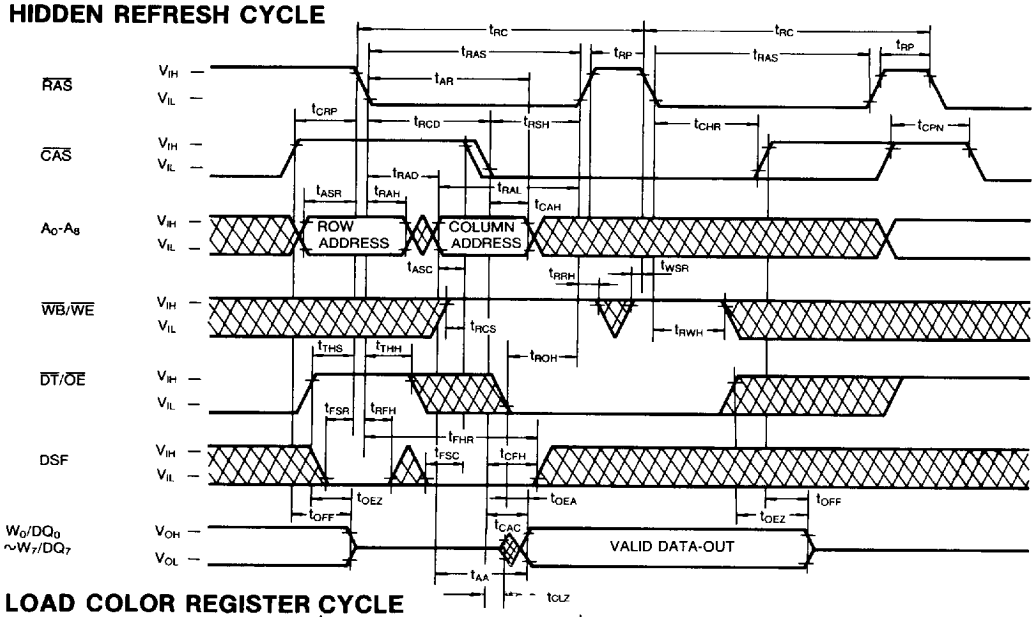
TIMING DIAGRAMS (Continued)

CAS-BEFORE-RAS REFRESH COUNTER TEST CYCLE

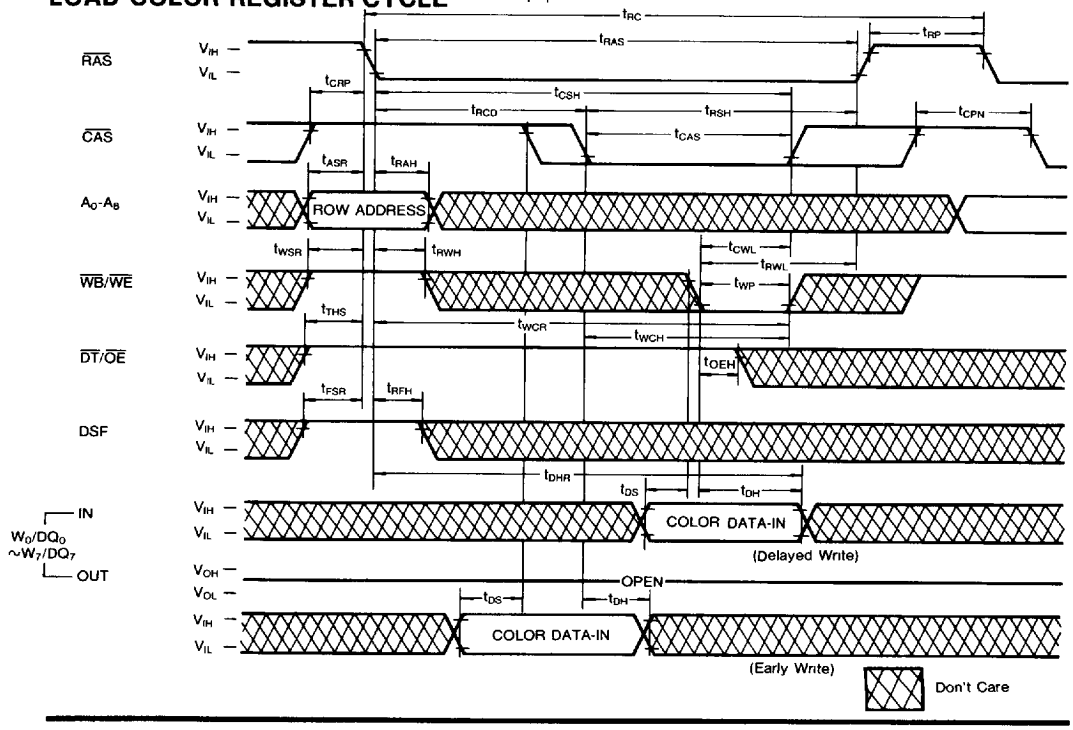


TIMING DIAGRAMS (Continued)

HIDDEN REFRESH CYCLE



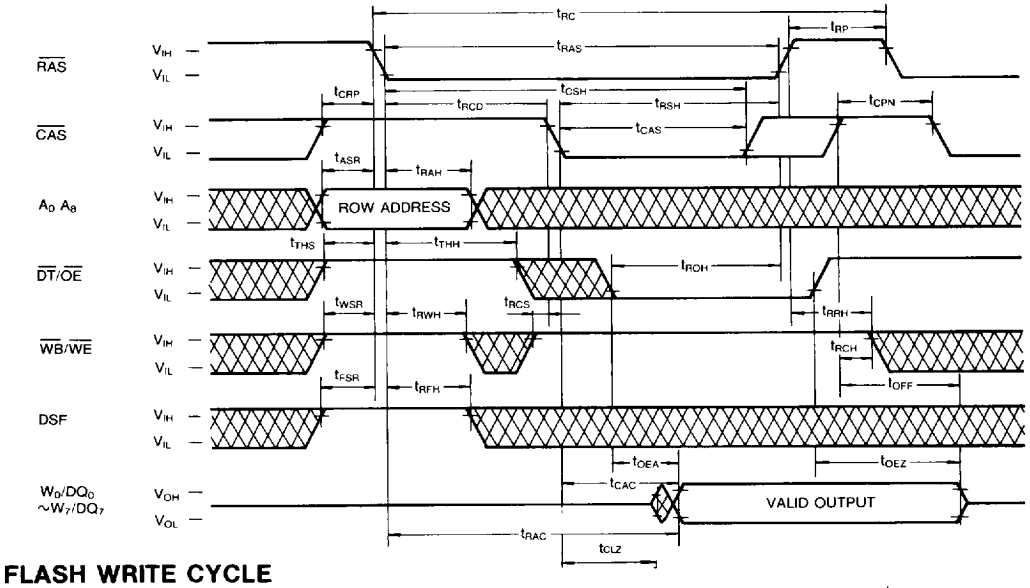
LOAD COLOR REGISTER CYCLE



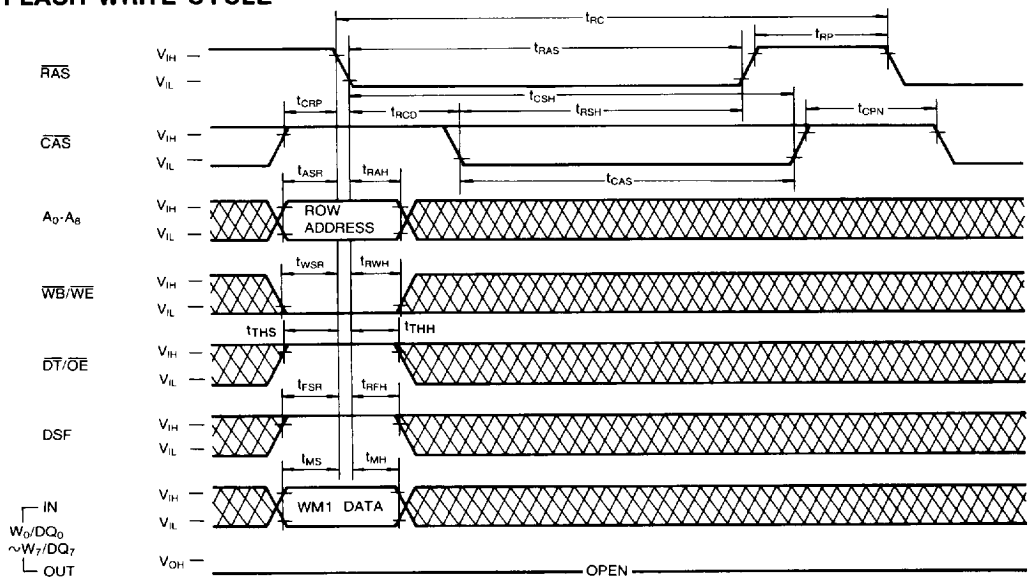
SAMSUNG ELECTRONICS

TIMING DIAGRAMS (Continued)

READ COLOR REGISTER CYCLE



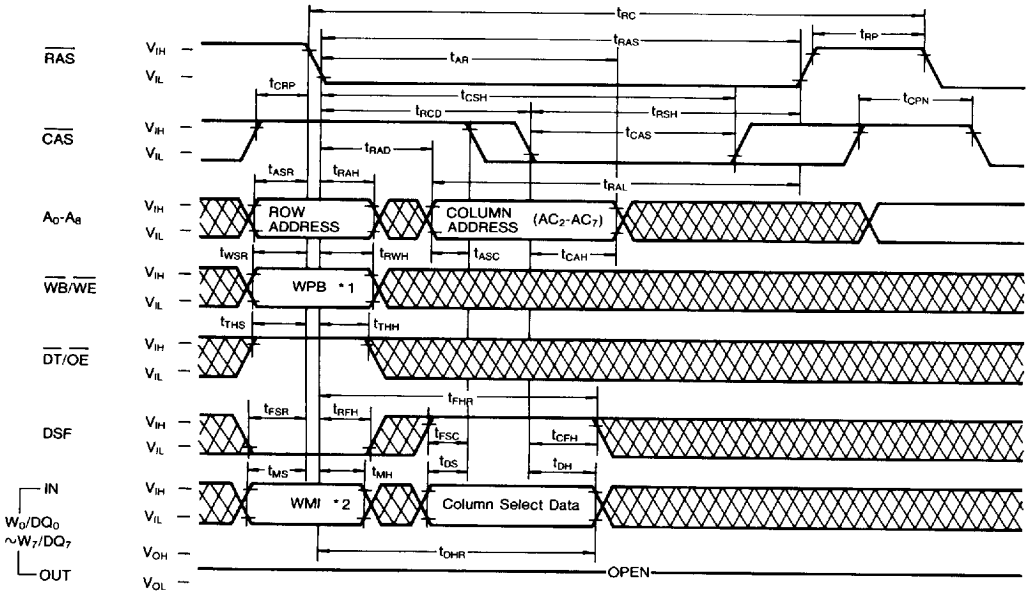
FLASH WRITE CYCLE



WM1 DATA	CYCLE
O	Flash write Disable
I	Flash write Enable

TIMING DIAGRAMS (Continued)

BLOCK WRITE CYCLE



*1 $\overline{WB}/\overline{WE}$	*2 W_0/DQ_0-W_7/DQ_7	CYCLE
0	WM1 Data	Masked Block Write
1	Don't Care	Block Write (Non Mask)

WM1 Data: 0: Write Disable
1: Write Enable

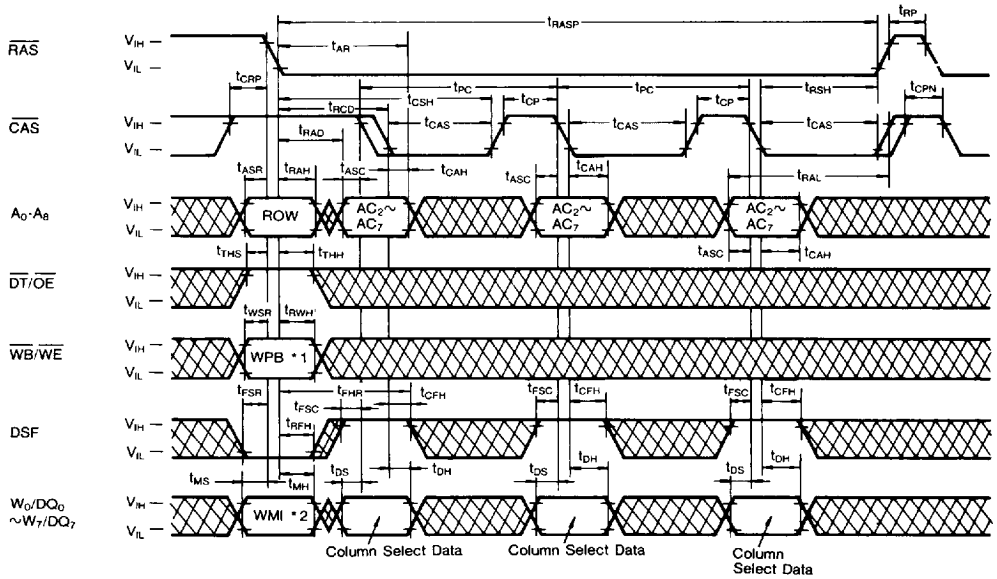
COLUMN SELECT DATA

W_0/DQ_0 — Column 0 (A_{IC}=0, A_{OC}=0)
W_1/DQ_1 — Column 1 (A_{IC}=0, A_{OC}=1)
W_2/DQ_2 — Column 2 (A_{IC}=1, A_{OC}=0)
W_3/DQ_3 — Column 3 (A_{IC}=1, A_{OC}=1)

Wn/DQn
= 0: Disable
= 1: Enable

TIMING DIAGRAMS (Continued)

PAGE MODE BLOCK WRITE CYCLE



*1 $\overline{WB}/\overline{WE}$	*2 W_0/DQ_0-W_7/DQ_7	CYCLE
0	WM1 Data	Masked Block Write
1	Don't Care	Block Write (Non Mask)

WM1 Data: 0: Write Disable
1: Write Enable

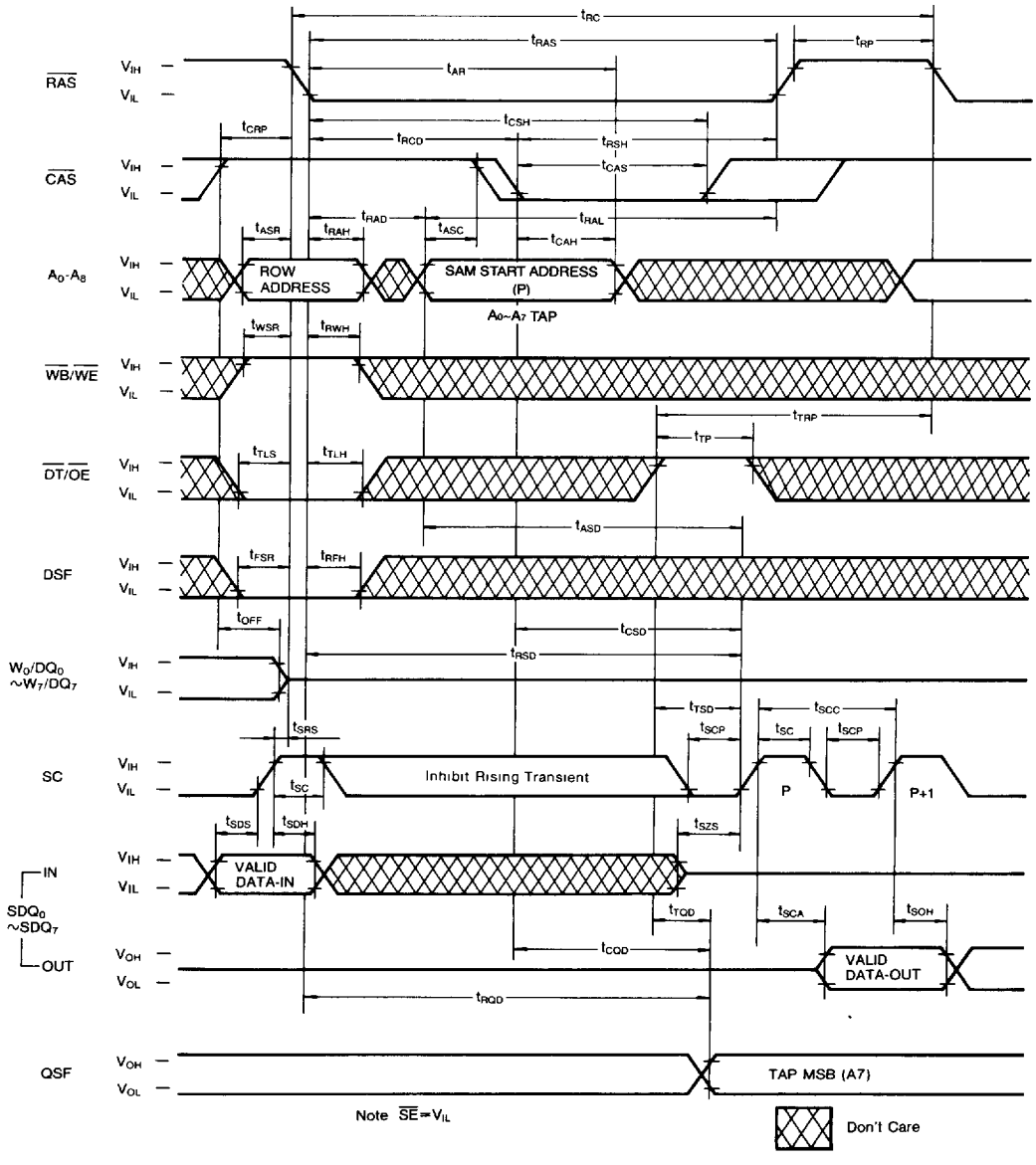
COLUMN SELECT DATA

W_0/DQ_0 — Column 0 (A_{IC}=0, A_{OC}=0)
W_1/DQ_1 — Column 1 (A_{IC}=0, A_{OC}=1)
W_2/DQ_2 — Column 2 (A_{IC}=1, A_{OC}=0)
W_3/DQ_3 — Column 3 (A_{IC}=1, A_{OC}=1)

Wn/DQn
= 0: Disable
= 1: Enable

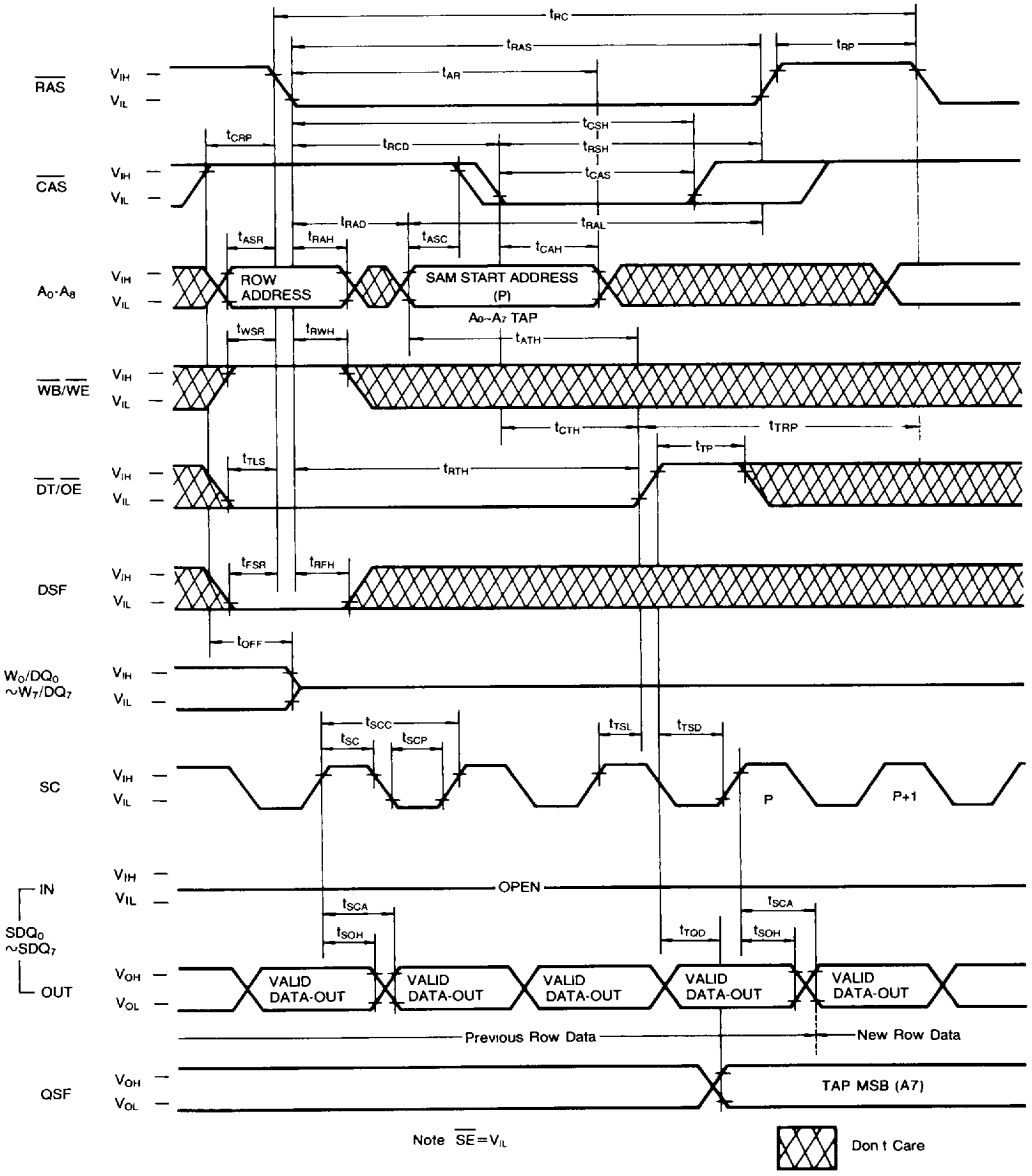
TIMING DIAGRAMS (Continued)

READ TRANSFER CYCLE



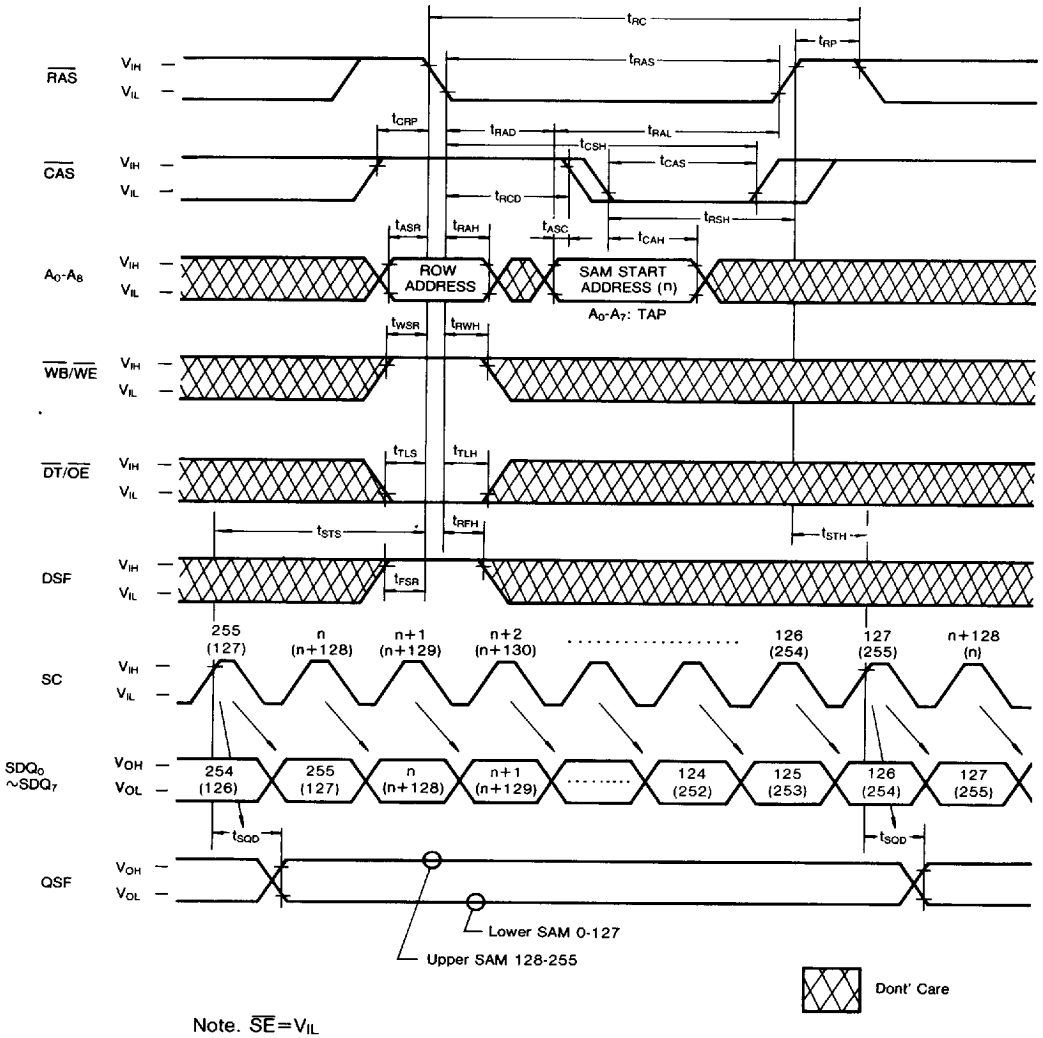
TIMING DIAGRAMS (Continued)

REAL TIME READ TRANSFER CYCLE



TIMING DIAGRAMS (Continued)

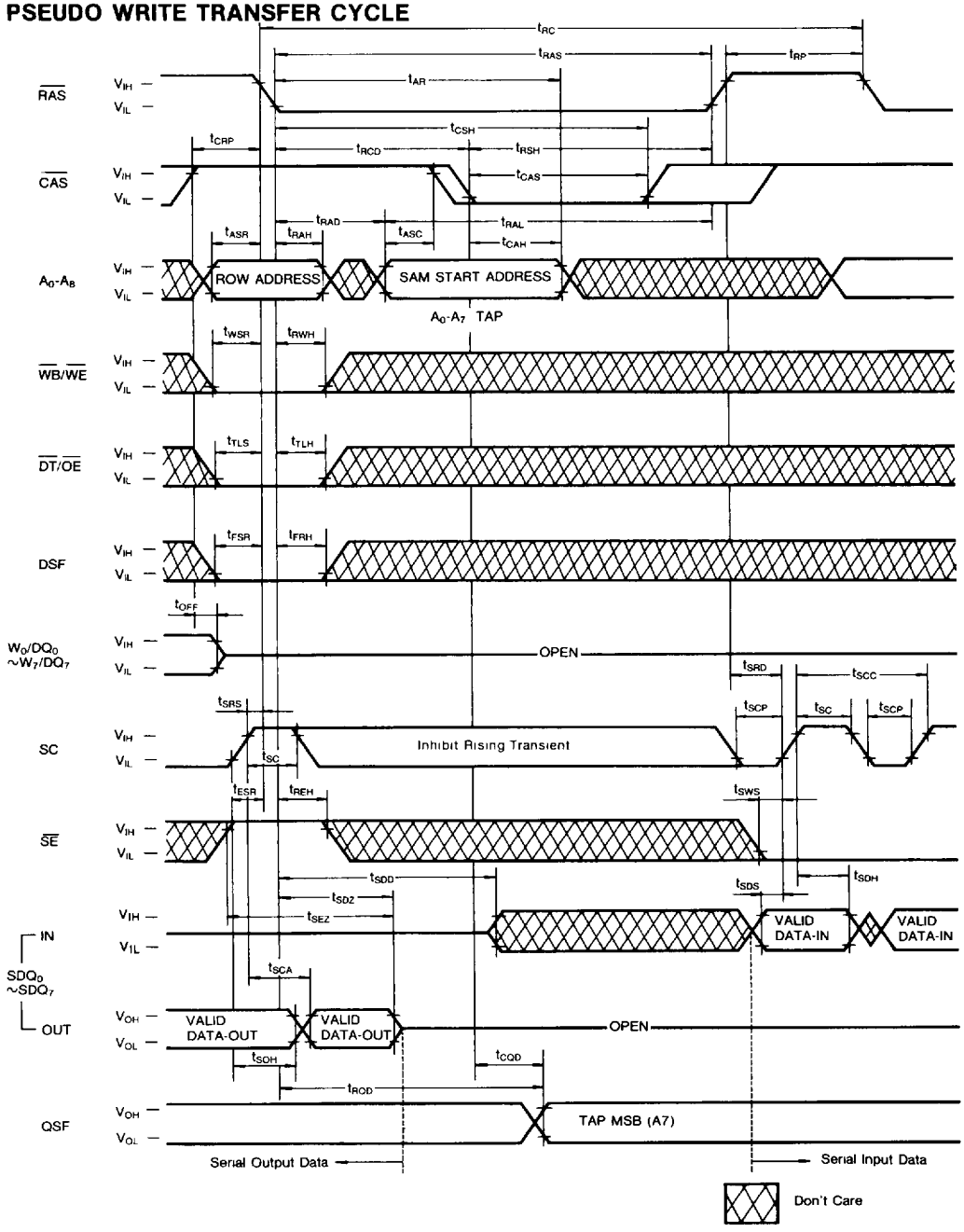
SPLIT READ TRANSFER CYCLE



Note. $\overline{SE}=V_{IL}$

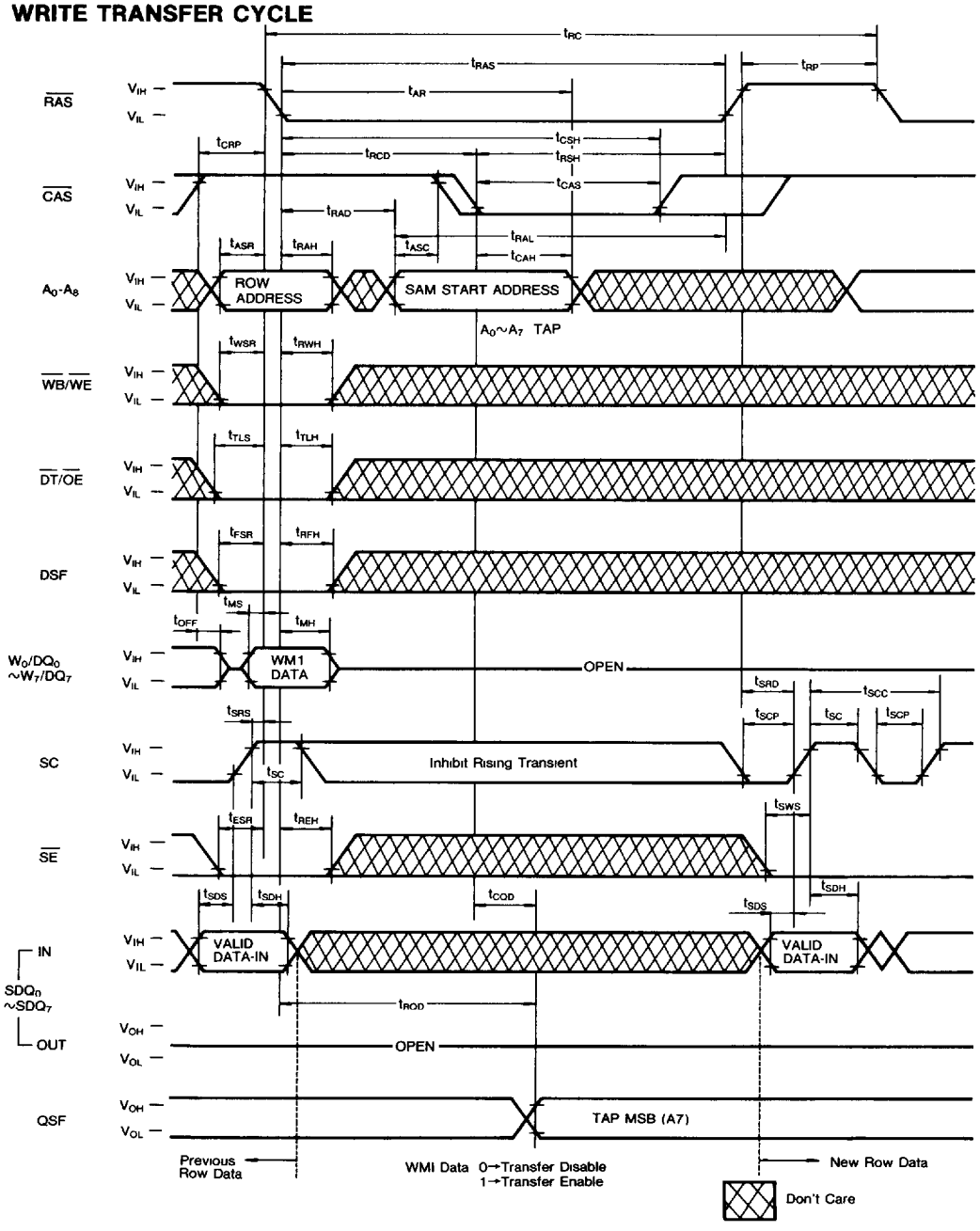
TIMING DIAGRAMS (Continued)

PSEUDO WRITE TRANSFER CYCLE



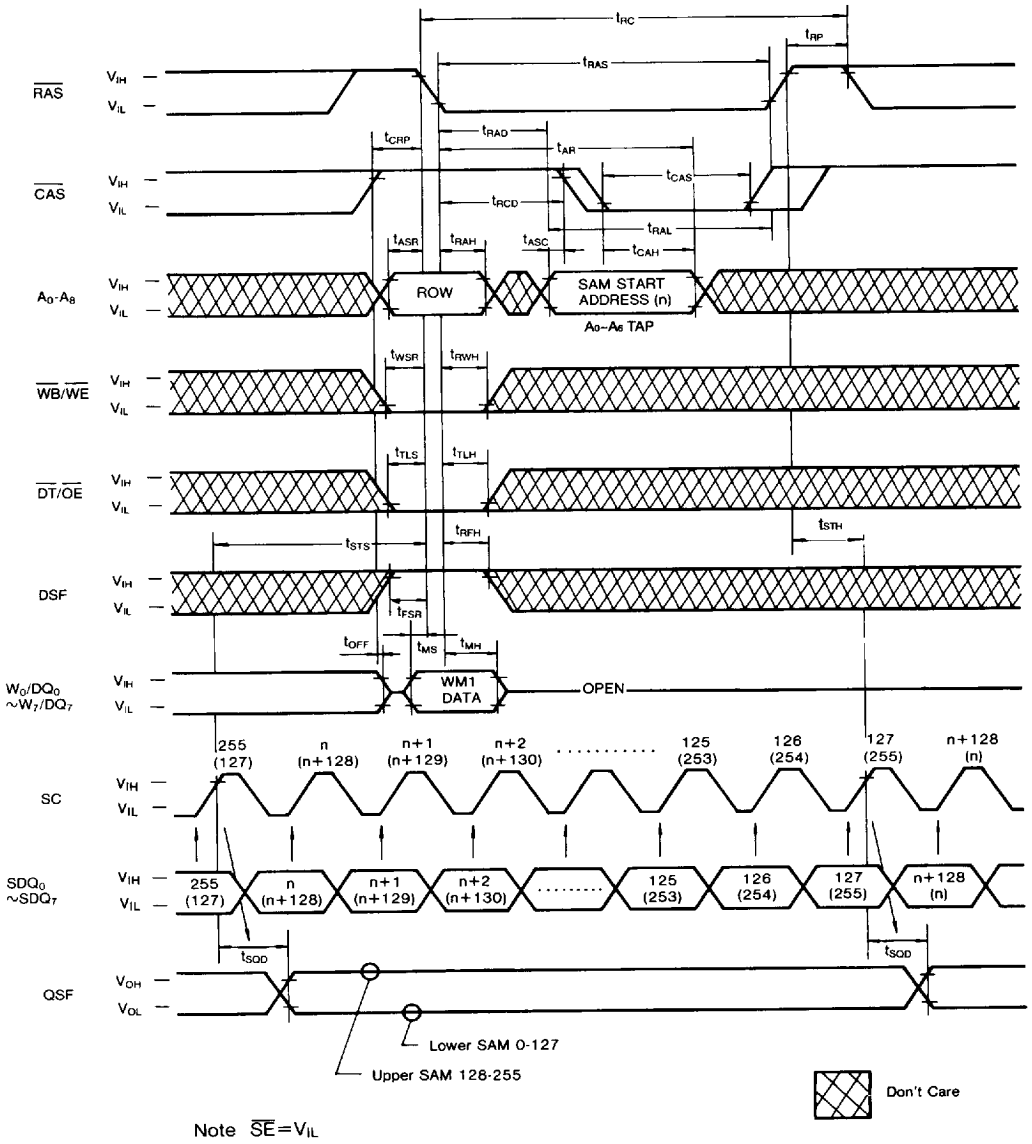
TIMING DIAGRAMS (Continued)

WRITE TRANSFER CYCLE



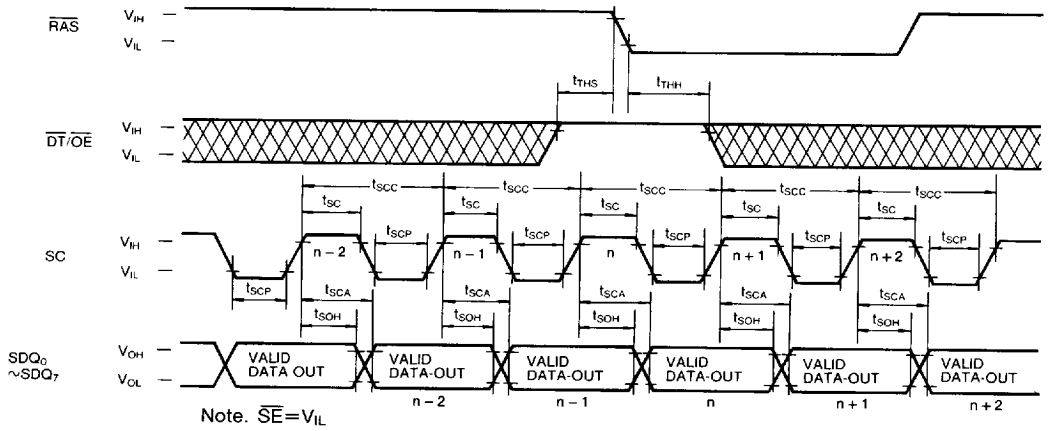
TIMING DIAGRAMS (Continued)

SPLIT WRITE TRANSFER CYCLE



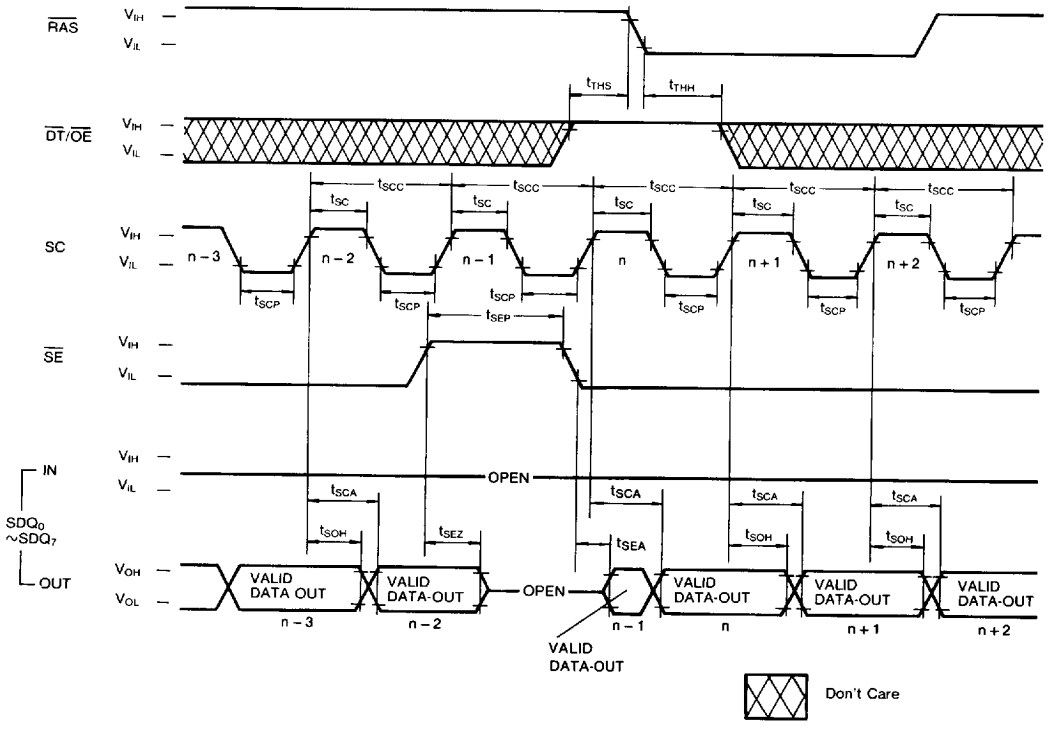
Note $\overline{SE}=V_{IL}$

SERIAL READ CYCLE ($\overline{SE}$ = V_{IL})



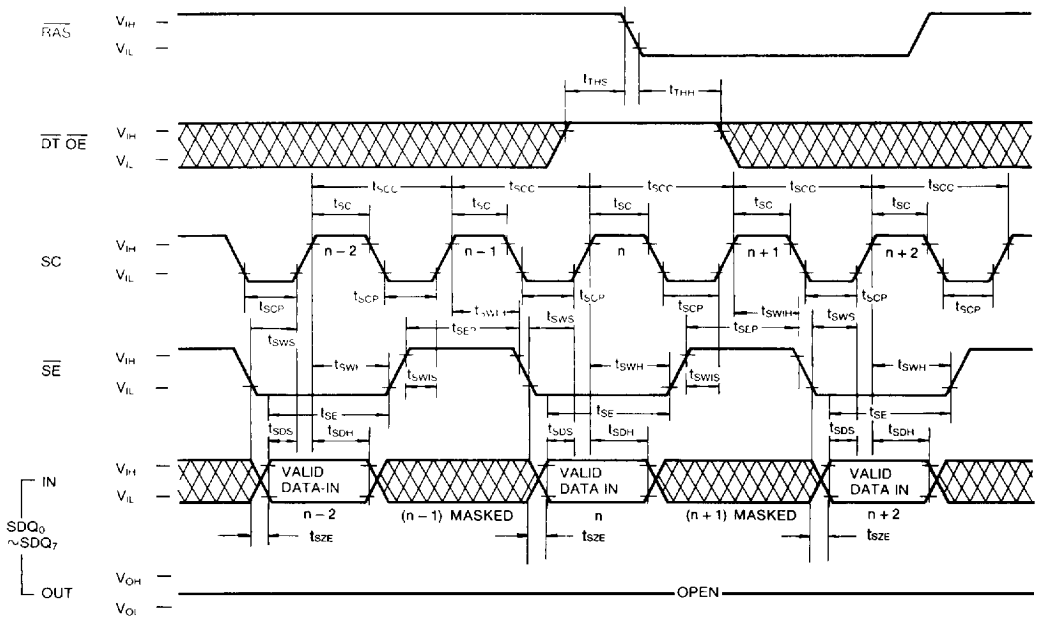
Note. $\overline{SE}$=V_{IL}

SERIAL READ CYCLE ($\overline{SE}$ Controlled Outputs)

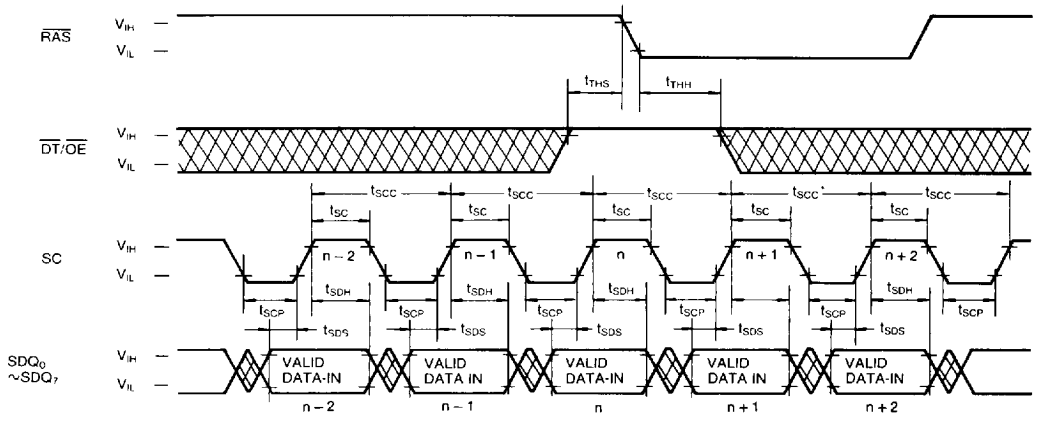


TIMING DIAGRAMS (Continued)

SERIAL WRITE CYCLE ($\overline{SE}$ Controlled Inputs)



SERIAL WRITE CYCLE ($\overline{SE}$ = V_{IL})



Note $\overline{SE}$=V_{IL}

Don't Care