Introduction

Overview

The 4megabit Video RAM is an application specific memory device designed for graphics applications. It comprises a 256k x16 DRAM memory array interfaced to a 256 x16 Serial Access Memory (SAM), or register. A half row of data, 256 columns x 16 bits, can be transferred from the DRAM array into the register in a single RAS cycle. Subsequently, the data can be read serially, 16bits(one address) at a time, starting from any arbitrary address location within the register. Once the data is transferred into the register from the DRAM array, the DRAM is free to be read or written through its random port. Concurrently and asynchronously, data can be accessed from the SAM port.

Features

- **8 x 8 (x2) Block Write allows as many as 128 bits of data to be written in every CAS cycle.**
- **Persistent and nonpersistent Write Per Bit operations allow individual write control to each DRAM I/O.**
- **Separate high and low byte write enable signals to facilitate byte write operations.**
- **Normal Fast Page mode with Extended Data Out delivers 33 MHz read/write performance on DRAM port.**
- **Pipelined page mode delivers 50 MHz peak read or write performance on DRAM port.**
- **Split serial register architecture allows one half of the register to be loaded from DRAM while the other half is read out of the SAM port.**
- **Binary stop detection in the SAM allows jumps to a new starting location while data is being accessed, facilitating the design of "tile mapped" displays and displays of any arbitrary pixel resolution.**
- **Selective Flash Write for large area fills (available at customer request).**
- **Compatible with JEDEC standards.**

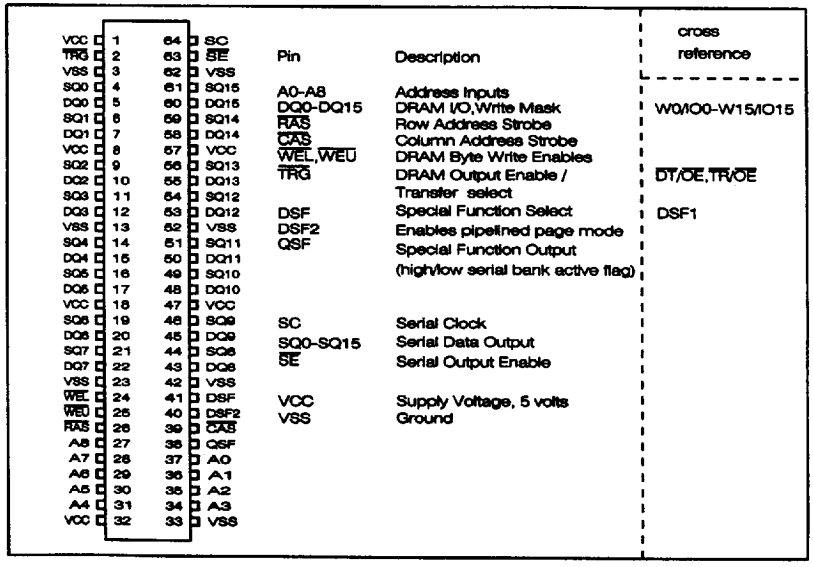


Figure 1. Pin Diagram and Nomenclature

RAS Access Time	RAS Cycle Time	Page Mode Cycle Time	Pipelined Page Mode Cycle Time	Serial Data Cycle Time	Icc, DRAM Port Read/Write	Icc,Serial Port
60 ns	101 ns	30 ns	20 ns	18 ns	140 mA	40 mA

Table 1. Major performance parameters

Definition of Terms and Conventions

RAM, or DRAM: The circuitry comprising the random access memory
SAM: The circuitry comprising the serial access memory.

IOi, DQi, Planei: The individual I/Os comprising the 16 bit DRAM port. DQi may also be used to refer to the specific package pin corresponding to I/Oi(Planei).

Low byte refers to data stored in or accessed from DQ0-DQ7.
High byte refers to data stored in or accessed from DQ8-DQ15.

Write Mask, or Write Mask Register : A register comprising 16 bits, each bit controlling a write operation to one of the 16 planes. In write per bit mode, write mask bit i enables or disables a write operation to IOi (planei). The write mask is thus a *control* register.

Write Time: The latter of $\overline{CAS}$ going low and either $\overline{WEL}$ going low, if the low byte is to be written, or $\overline{WEU}$ going low, if the high byte is to be written. This is consistent with conventional DRAM timing except that individual write controls are available for the low and high bytes.

Signal(RAS): The state of *signal* when $\overline{RAS}$ goes low.
Signal(CAS): The state of *signal* when $\overline{CAS}$ goes low.
Signal(write): The state of *signal* at write time.

Address Mask, Column Mask, or Column Address Mask : The mask formed by DQ0(write)-DQ7(write) for the lower byte and DQ8(CAS)-DQ15(CAS) for the upper byte which replaces the column address bits A0-A2 during Block Write operations. The address mask contains only address information.

A signal is said to be asserted when it is in its active state, regardless if the signal is active high or active low.
A signal is said to be negated if it is in its inactive state, regardless if the signal is active high or active low.

Tap, or Starting Location: The position within the SAM at which the SAM will begin access after a transfter cycle or bank switch.

Quad Row: One quarter of a row of DRAM, ordered as follows:

Column Addresses	Quad Row
0-127	0
128-255	1
256-383	2
384-511	3

Half Row: One half of a row of DRAM, ordered as follows:

Column Addresses	Half Row
0-255	low
255-511	high

SAM Bank, or Bank: One half of the SAM register, designated as either the low or the high bank. Mapping to column addresses will depend on whether or not binary stop bit mode is enabled.

Word: The sixteen bits of data, DQ0-DQ15 correponding to a single memory address.

Architecture

Figure 3 shows the top level block diagram of the 4 megabit Video RAM. The shaded area provides a conceptual boundary of the SAM and the supporting circuitry required to transfer data from the RAM to the SAM. The area outside of the shaded region corresponds to the RAM. The signals in boldface are external, or package pin signals.

Pin Description(per mode)

A0-A8(address); Address, Input

RAM Operations: Eighteen address bits of information are required to decode one of 256K memory addresses. The address is divided into a 9-bit row address and a 9-bit column address. The row address is presented on A0-A8 on the falling edge of $\overline{RAS}$ and the column address is presented on A0-A8 on the falling edge of $\overline{CAS}$.

Transfer Operations: In a normal transfer operation, the address of the row to be transferred to the SAM is presented on A0-A8 at the falling edge of $\overline{RAS}$. The column address within the row at which serial data will begin access from is presented on A0-A8 on the falling edge of $\overline{CAS}$. Column address A8 designates either the lower 256 columns (0-255) of the row or the upper 256 columns (256-511) of the row to be transferred to the 256 bit positions in the SAM. A8=0 selects the lower 256 columns and A8=1 selects the upper 256 columns. Column addresses A0-A7 determine which of the 256 possible start, or *tap* locations the SAM will begin reading from.

In split register mode the SAM is further divided into two banks, a low bank comprising register locations 0-127 and a high bank comprising register locations 128-255. Each bank holds one *quad row* of data. During split register transfer operations. A7 is ignored on the falling edge of $\overline{CAS}$. An internal toggle flip flop arrangement remembers which segment is active and which segment is to be reloaded.

When enabling the stop bit mode of serial operation via a CBR refresh with stop bit enable (CBRS) cycle, A4-A7 contain the stop bit information at the falling edge of $\overline{RAS}$. Strictly speaking, A7 is not used. However to maintain compatibility with 512 word SAM devices it is recommended that A7 be supplied and set to a logic low level during all CBRS cycles.

DQ0-DQ15(RAM data); Data, IO; Address (block write), Input

RAM Operations: During normal DRAM operations, DQ0-DQ15 provide the datapath into and out of RAM. During read cycles, data from the selected memory address is delivered to DQ0-DQ15 after the $\overline{RAS}$, $\overline{CAS}$, and output enable ($\overline{TRG}$) access times have been satisfied. During write cycles, DQ0-DQ15 provide the data to be written to RAM.
DQ0-DQ15 will go to low impedance during all read, late write, and read-modify-write cycles when the following events and conditions occur.

1. $\overline{TRG}$ is brought low after $\overline{RAS}$ goes low
2. $\overline{CAS}$ goes low while $\overline{RAS}$ is low and $\overline{WEL}$, $\overline{WEU}$ are high

DQ0-DQ15 will return to high impedance when any of the following events occur.

1. $\overline{TRG}$ goes high
2. $\overline{RAS}$ and $\overline{CAS}$ both go high (end of active cycle)
3. Either $\overline{WEL}$ or $\overline{WEU}$ goes low during any active read, write, or hidden refresh cycle.

$\overline{TRG}$ provides unlatched output impedance control of DQ0-DQ15. If $\overline{TRG}$ goes high and then low during an active read cycle, DQ0-DQ15 will go to high impedance and then back to low impedance. If $\overline{RAS}$ and $\overline{CAS}$ both go high, DQ0-DQ15 will go to high impedance and remain in high impedance until the next $\overline{RAS}$ activated read, late write, or read-modify write cycle. If $\overline{WEL}$ or $\overline{WEU}$ go low during any active cycle, then DQ0-DQ15 will go to high impedance and remain in high impedance until either of the following conditions occur.

a. $\overline{WEL}$, $\overline{WEU}$ return high and $\overline{CAS}$ is then brought low (eg. a page mode write-read sequence)
b. The $\overline{RAS}$ cycle ends and another read, late write, or read-modify write cycle is performed

Figure 2 below shows a simplified logic diagram for the output enable control circuitry during read, write, and hidden refresh cycles. DQ0-DQ15 remain in high impedance during $\overline{CAS}$ before $\overline{RAS}$ auto refresh cycles.

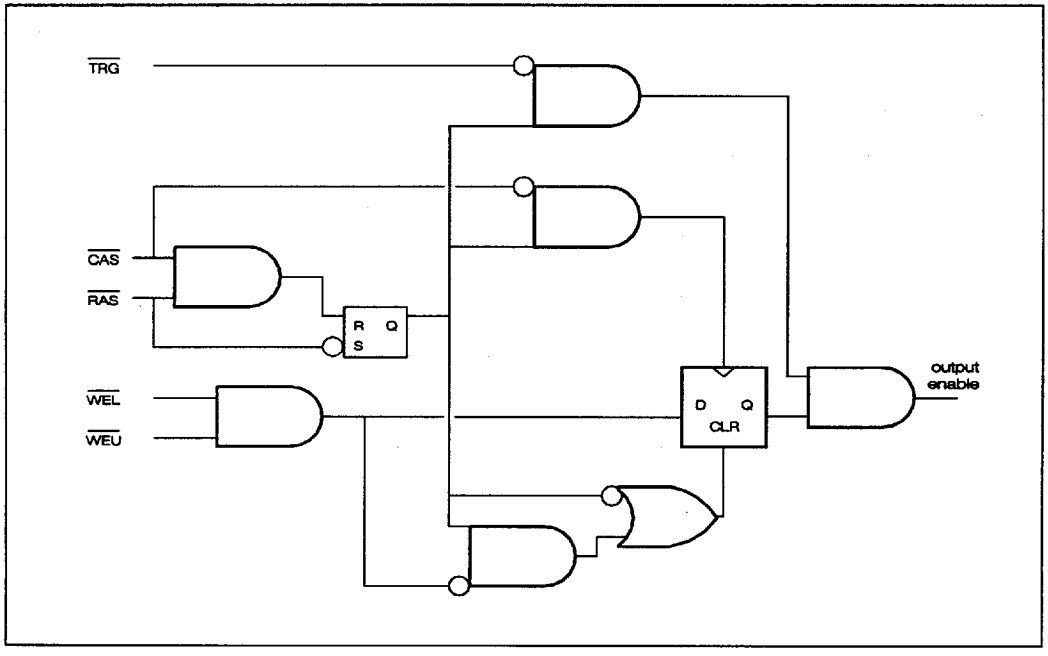


Figure2. Simplified logic of output enable control for read and write cycles

Transfer Operations: DQ0-DQ15 are not used during RAM to SAM transfer operations. They remain in high impedance for the entire transfer cycle.

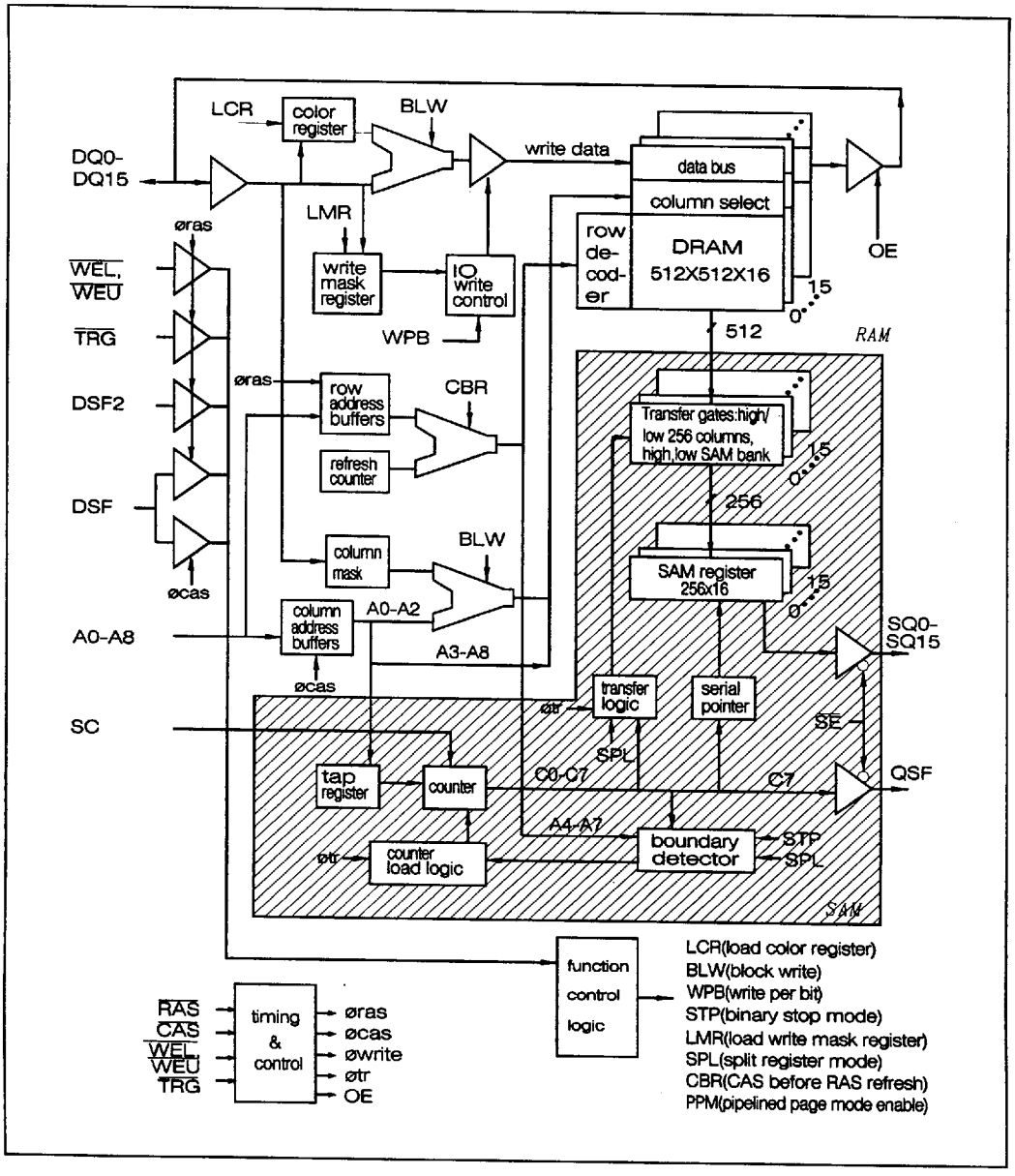


Figure 3. Top Level Functional Block Diagram

$\overline{RAS}$(Row Address Strobe); Control, Input

RAS initiates all RAM and transfer cycles. On the falling edge of $\overline{RAS}$, a memory or transfer cycle is initiated, a row address is latched on A0-A8, and the states of $\overline{CAS}$, $\overline{TRG}$, $\overline{WEU}$, $\overline{WEL}$, DSF, and DSF2 are latched to determine the type of cycle to be executed. $\overline{RAS}$ going high terminates the active portion of the memory cycle, except that any data read on DQ0-DQ15 will remain active untill $\overline{CAS}$ goes high or $\overline{TRG}$ goes high. This allows hidden refresh cycles to be supported.

$\overline{CAS}$(Column Address Strobe): Control, Input

CAS is a control input which functions as a column address strobe and cloumn select.

RAM operations: $\overline{CAS}$ going low latches the column address presented on A0-A8 and enables the column circuitry. $\overline{CAS}$ going low also latches the state of DSF to enable/disable block write operation or to enable/disable a "load color register" cycle. See Table 2 for details. The 4 megabit Video RAM supports Extended Data Output Page Mode operation, which means that data remains valid after $\overline{CAS}$ goes high. This allows the address for the next page mode cycle to be set up while data from the current cycle is latched by the accessing device.

Transfer Operations: During transfer operations, $\overline{CAS}$ going low latches A0-A8 to select which columns of the selected row in RAM are to be transferred to the SAM and also to select the starting location at which the SAM will begin accessing from.

$\overline{TRG}$ (Transfer / RAM output enable): Control, Input

The $\overline{TRG}$ signal is latched on the falling input of $\overline{RAS}$ to select either a RAM or a transfer operation.

RAM operations: $\overline{TRG}$ functions as an output enable for DQ0-DQ15.

Transfer operations: On normal transfer cycles, $\overline{TRG}$ controls the physical transfer of data from the RAM to SAM and the reloading of the serial counter to the new tap address. During split register transfer cycle, holding $\overline{TRG}$ low when $\overline{RAS}$ falls *enables* a transfer from RAM to the inactive bank of the SAM. The timing of the physical transfer of data is controlled internally and is completed within the transfer cycle. Also during split register mode operation, the reloading of the serial counter, and therefore the readout of the SAM from the new tap location in the inactive bank, is deferred until after the most significant address (binary stop mode disabled) or a specified boundary address (binary stop mode enabled) is read out from the active bank of the SAM.

$\overline{WEL}$, $\overline{WEU}$ (Byte Write Enable, write per bit select): Control, Input

RAM Operations: $\overline{WEL}$, $\overline{WEU}$ are latched on the falling edge of $\overline{RAS}$ to enable or disable write per bit operation. If either $\overline{WEL}$ or $\overline{WEU}$ is held low on the falling edge of $\overline{RAS}$, write per bit mode will be enabled for all 16 DQs. If write per bit mode is disabled, then the write to the sixteen DQs is controlled exclusively by $\overline{WEL}$(for DQ0-DQ7) and $\overline{WEU}$ (for DQ8-DQ15). If write per bit mode is enabled, then the write to any DQ is enabled by setting its correponding write mask bit to 1 using either persistent or non persistent write per bit mode. Subsequently, the actual write to that DQ is performed by asserting that DQ's correponding byte write enable, $\overline{WEL}$ or $\overline{WEU}$, during write time. $\overline{WEL}$, and $\overline{WEU}$ can also be used to control the output impedance of DQi, All DQ's will go to the high impedance state whenever $\overline{WEL}$ or $\overline{WEU}$ is asserted(low).

Serial Operations: $\overline{WEL}$, $\overline{WEU}$ are not used during actual serial operation, however they are used to invoke the boundary stop mode of serial operation. This is accomplished by performing a special $\overline{CAS}$ before $\overline{RAS}$(CBRS) cycle with either $\overline{WEL}$ or $\overline{WEU}$ held low on the falling edge of $\overline{RAS}$. See Table 2 for details.

Transfer Operations: not used

DSF (Special Function Select); Control, Input

Random Operations: Enables persistent write per bit mode and loads the write mask register. Allows persistent write per bit mode to be sustained during CBR cycles. Enables block mode write operation and the loading of

the on chip color register used during block write operations. Enables selective flash write operation using the contents of the color register. (Flash write is available as a special part number at customer request.) See Table 2 for details.

Transfer Operations: Selects between split register and non split register transfer and serial operation.

DSF2(Special Function Select): Control, Input

Random Operations: Enables / disables pipelined page mode operation.

Transfer Operations: Not used.

SC(serial clock): Clock, Input

RAM Operations: not used

Transfer Operations: Not used per $\overline{SE}$, but there are important timing parameters which must be met to synchronize SC to the transfer operation. In normal transfers (split register mode disabled). SC must be synchronized with respect to $\overline{TRG}\uparrow$ to be sure that the last bit of the old register contents and the first bit of the new register contents are read properly. During split register transfers, the transfer cycle must be completed sufficiently in advance of the last bit being read out of the SAM from the active portion of the register. See Timing Diagrams for details.

SQ0-SQ15(Serial Data); Output

RAM Operations: not used

Transfer Operations: Not used.

Serial Operations: SQ0-SQ15 are the output data terminals for the SAM. Data from the SAM is normally accessed serially, module 256, from each SC↑ starting from the tap location selected during the the previous transfer cycle. In split register mode, however, the SAM will skip to the tap address in the inactive bank whenever a "boundary crossing" occurs in the active bank and a transfer to the inactive bank has occured before the boundary in the active bank is reached.

$\overline{SE}$(Serial Output and QSF Enable); Control, Input

Serial Operations: Enables or disables serial data output at SQ0-SQ15 and the bank active output QSF. Important: $\overline{SE}$ is only a serial *output* enable, not a general serial *mode* enable. Thus, if $\overline{SE}$ is negated, the counter is not prevented from advancing if the serial clock (SC) continues to toggle. The serial register and supporting circuits are static, so no refresh operations are required to sustain information in the SAM.

QSF (Special Function Output, Active Register Flag); Control, Output

Serial Operations: This signal outputs a 0 if the lower bank of the register is being accessed, a 1 if the higher bank of the register is being accessed, and is in the high impedance state if SE is negated.

Description of Functional Units

The block diagram shown in Figure 3 should be referenced for discussions of the major functional blocks.

Write Mask Register

The write mask register is a 16 bit register used to control the internal write enables to each IO, or plane in RAM. The write mask register can be loaded from the DQ inputs on a per write cycle basis, known as "nonpersistent" mode, or can be loaded during a special "Load write Mask Register" (LMR) cycle with timing and control similar to a conventional DRAM write cycle. Once a Load Write Mask cycle is executed, the Video RAM goes into "persistent" mode, and will use the contents of the write mask on every write per bit write cycle thereafter until persistent mode is disabled. While in persistent mode, the contents of the write mask cannot be altered, except

by another Load write Mask Register cycle. The Video RAM can be returned to nonpersistent mode by performing a special $\overline{CAS}$ before $\overline{RAS}$(CBRR) cycle "with option reset." See Table 2 for details.

Byte Write Input Buffers

To facilitate byte write operations, the 4Mb Video RAM features separate $\overline{WE}$ pins, denoted $\overline{WEL}$ and $\overline{WEU}$ (dual $\overline{WE}$ version)or separate $\overline{CAS}$ pins, denoted $\overline{CASL}$ and $\overline{CASU}$,(dual $\overline{CAS}$ version) to individually control write operations to DQ0-DQ7 and DQ8-DQ15 respectively. Timing for each is similar to conventional DRAM operation.
The byte write enable signals can be combined with write per bit operation. For example, if write per bit mode is enabled, a particular IO is written if and only if both its byte write enable signal and its correpoading bit in the write mask register are asserted.

Color Register

The color register is a 16 bit register used as an alternate data source during block write operations. The register is loaded from the DQi inputs during "Load Color Register" (LCR) cycles and used as data during block write cycles. Once loaded, it cannot be altered except by another LCR cycle or by removing power from the device.

Column Address Mask

During block write cycles, the column mask is used to select any arbitrary subset of eight contiguous column locations in RAM chosen by A3-A8 to be written using the contents of the color register. The column mask is loaded from the DQ inputs at write time during block write cycles. Separate 8-bit masks are loaded on DQ0-7 and DQ8-15, meaning that the subset of columns written to memory can be different for the high and low bytes. If block write mode is enabled, the column mask can be changed during each page mode write cycle, allowing the column mask to be controlled dynamically during page mode operation as A3-A8 is changed.

Special Function Logic

The special function logic is a control block that takes as input $\overline{CAS}$(RAS), $\overline{TRG}$(RAS), $\overline{WEL}$(RAS), $\overline{WEU}$(RAS), DSF(RAS), and DSF(CAS) and gives as output the control signals necessary to execute the various special function cycles. The modes can be deduced from the signals output from the special function logic in Figure 3.

Serial Access Memory(SAM)

Although the SAM is accessed serially according to the corresponding column addresses in RAM, it is implemented internally as a RAM with a counter supplying the address. The counter is incremented by each positive edge of the serial clock, SC.

Transfer Logic

Controls the logic and timing for the various operating modes of transfer operation, including real time data transfer, split register transfer, early and late load, and binary stop mode enabled and disabled.

Transfer Gates

The transfer gates are MOSFETS connected between each pair of bit lines and a register bit inside the SAM. While in split register mode the register can be loaded one bank at a time. The upper bank can be loaded while the lower bank is read out through the SAM port and vice versa.

Counter

The counter is presettable from the tap register. During normal transfers (split register mode disabled), it is loaded from the tap register while the transfer logic completes the physical transfer of data from RAM to SAM. In split register mode, the loading of the counter is deferred until the last bit is read out of the active bank.

MSB Detector

During split register mode, the MSB detector reads the outputs of the counter and issues a bank switch signal when the most significant address is read from either the high or low bank in the SAM. The MSB detector essentially detects all 1's from the seven least significant bits of the counter. When the binary stop bit mode is enabled, only some of these bits are compared for 1's, depending upon which stop bits are enabled. Enabling the stop bits is done by performing a CBRS cycle(See Table 2 for detailed logic assignments).

Timing

In general, there are three types of operations which can be occuring in the Video RAM: DRAM access, serial access, and transfer cycles. When not transferring data from RAM to SAM, the timing block can be thought of as two independent sub blocks, one controlling DRAM read, write, and refresh operations and the other controlling the advancing of the counter and the serial read path in the SAM. During transfer cycles, the two timing blocks must be synchronized, especially during real time register load cycles.

Summary of Special Functions

The special functions of the 4Mb Video RAM, are invoked by decoding the states of the control pins, including DSF and DSF2, on the falling edges of $\overline{RAS}$ and $\overline{CAS}$. The DQ pins will assume various functions, providing information other than just normal RAM data, depending on the particular cycle invoked. For example, the DQ pins serve as a write mask on write per bit cycles and as a multiple (column) address mask on block write cycles.

Figure 4 below illustrates the general timing by which the control inputs are latched and the cycle types that they select. See the timing parameters and diagrams for details.

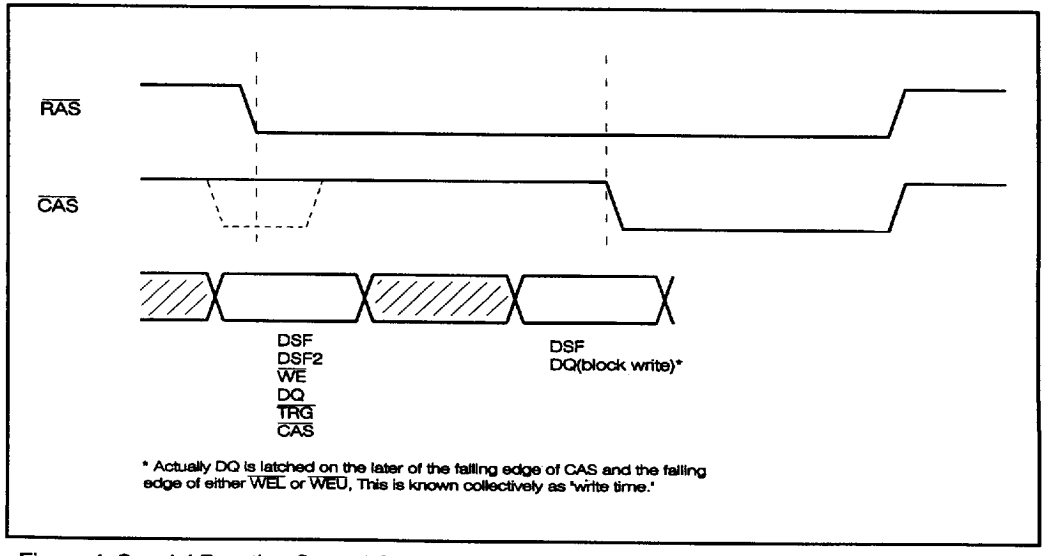


Figure 4. Special Function Control Scheme

Table 2 shows the JEDEC approved truth table describing how the various DRAM and transfer cycles are invoked. The *mne* codes are used throughout this document to refer to the particular cycles defined in the table.

Function, or Cycle Type	RAS↓ signal(RAS)							CAS↓ signal(CAS)		CAS↓ & (WEL↓ + WEU↓)[1]	mne code
	CAS	TRG	WEL, WEU[2]	DSF	DSF2	Address	DQi[3]	DSF	Address	DQi	
CBR, reset to nonpersistent & disable stop bit mode	0	x	1	0	0	x	x	x	x	x	CBRR
CBR, Enable, set Stop bit	0	x	0	1	0	stop A4-A7	x	x	x	x	CBRS
CBR,No reset to nonpersistent or disable of stop bit	0	x	1	1	0	x	x	x	x	x	CBRN
Read Transfer	1	0	1	0	0	row	x	x	tap	x	RT
Split Register Read Transfer	1	0	1	1	0	row	x	x	tap	x	SRT
DRAM write per bit, nonpersistent, use new mask	1	1	0	0	note4	row	Write-mask	0	column	data	RWM
DRAM bolck write with write per bit, nonpersistent, use new mask	1	1	0	0	note4	row	write-mask	1	block addr A3-A8	col addr mask	BWM
DRAM write per bit, persistent, use old mask	1	1	0	0	note4	row	x	0	column	data	RWM
DRAM block write with write per bit, persistent, use old mask	1	1	0	0	note4	row	x	1	block addr A3-A8	col addr mask	BWM
Masked Flash Write, nonpersistent, use new mask	1	1	0	1	0	row	write-mask	x	x	x	FWT[5]
Masked Flash Write, persistent, use old mask	1	1	0	1	0	row	x	x	x	x	FWT[5]
DRAM write, no write per bit, no mask	1	1	1	0	note4	row	x	0	column	data	RW
DRAM block write, no write per bit, no mask	1	1	1	0	note4	row	x	1	block addr A3-A8	col addr mask	BW
Load Write Mask Register	1	1	1	1	0	x	x	0	x	write mask	LMR
Load Color Register	1	1	1	1	0	x	x	1	x	color data	LCR

Table 2. Special Function Truth Table

Notes;

[1] The signals on all DQ pins are strobed in on the later of the falling edge of CAS and the falling edge of either WEL and WEU, However, the low (high)byte is not enabled for write operation unless WEL (WEU) is asserted.

[2] The 0's in this column denote that WEL=0 or WEU=0. The 1's in this column denote that WEL=1 and WEU=1.

[3] DQi refers to DQ0 - DQ15 collectively.

[4] Available in extended data out page mode (DSF2 low when RAS↓) and pipelined page mode with extended data out (DSF2 high when RAS↓).

[5] Available by special part number at customer request.

Parametric specifications

Absolute Maximum Ratings

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating Temperature (Ambient)	0 to 70	°C
T_{STG}	Storage Temperature(Plastic)	-65 to 150	°C
V_{IN}, V_{OUT}	Voltage on Vcc Supply Pin Relative to Vss	-1.0 to 7.0	V
I_{SO}	Short Circuit Output Current	50	mA
P_D	Power Dissipation	1.1	W

NOTE: Stresses greater than those listed above may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

Recommended Operating Conditions

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
Vcc	Supply Voltage	4.5	5.0	5.5	V
Vss	Supply Voltage	–	0	–	V
V_{IH}	Input High Voltage	2.4	–	6.5	V
V_{IL}	Input Low Voltage	-1.0	–	0.8	V

Capacitance

(T_A=25°C, Vcc=5V+10%, f=1MHz)

SYMBOL	PARAMETER	MAX.	UNIT
C_{IC}	Input Capacitance($\overline{RAS}$,$\overline{CAS}$, $\overline{WEL}$,U,$\overline{TRG}$, SC,$\overline{SE}$, DSF,DSF2)	7	pf
C_{IA}	Input Capacitance(A0 - A8)	5	pf
$C_{I/O}$	Input/Output Capacitance(DQ)	7	pf
C_O	Output Capacitance (SQ, QSF)	7	pf

DC Characteristics

(T_A=0°C to 70°C, V_{CC}=5V+10%, V_{SS}=0V, unless otherwise noted.)

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	MAX.	UNIT
I_{LI}	Input Leakage Current	0V ≤ V_{IN} ≤ 5.5V All other pins not under test=V_{SS}	-10	10	μA
I_{LO}	Output Leakage Current	0V < Vout < 5.5V SQ, DQ pins disabled	-10	10	μA
V_{OL}	Output Low Voltage	I_{OL}=2mA	-	0.4	V
V_{OH}	Output High Voltage	I_{OH}=-1mA	2.4	-	V

ICC Characteristics

(T_A= 0°C TO 70°C, V_{CC}=5V+10%, V_{SS}=0V, unless otherwise noted.)

SYMBOL	PARAMETER	TEST CONDITIONS	SAM	HY5216257 -60 MAX.	HY5216257 -70 MAX.	UNIT	NOTE
I_{CC1}	Operating Current -Read/Write/LMR/LCR	$\overline{RAS}$ & $\overline{CAS}$ Cycling, t_{RC}= t_{RC} (min.)	Standby	140	115	mA	1
I_{CC1A}			Active	185	135	mA	2
I_{CC2}	Standby Current -TTL Input Levels	$\overline{RAS}$ & $\overline{CAS}$=V_{IH}, Other Pins V_{SS}	Standby	5	5	mA	1
I_{CC2A}			Active	45	40	mA	2
I_{CC3}	RAS Only Refresh Current	$\overline{RAS}$ Cycling, $\overline{CAS}$=V_{IH}, t_{RC}=t_{RC} (min.)	Standby	140	115	mA	1
I_{CC3A}			Active	185	150	mA	2
I_{CC4}	Fast Page Current	$\overline{RAS}$=V_{IL}, $\overline{CAS}$ Cycling, t_{PC}=t_{PC} (min.)	Standby	110	85	mA	1
I_{CC4A}			Active	140	115	mA	2
I_{CC5}	$\overline{CAS}$ Before $\overline{RAS}$ Auto Refresh Current	$\overline{RAS}$ &V_{IL}, $\overline{CAS}$ Cycling, $t\overline{RC}$=$t\overline{RC}$ (min.)	Standby	140	115	mA	1
I_{CC5A}			Active	185	155	mA	2
I_{CC6}	Data Transfer Current	$\overline{RAS}$ & $\overline{CAS}$ Cycling, t_{RC}=t_{RC}(min.)	Standby	140	115	mA	1
I_{CC6A}			Active	185	155	mA	2
I_{CC7}	Flash Write Current	$\overline{RAS}$ & $\overline{CAS}$ Cycling, t_{RC}=t_{RC}(min.)	Standby	140	115	mA	1
I_{CC7A}			Active	170	142	mA	2
I_{CC8}	Page Block Write Current	$\overline{RAS}$=V_{IL}, $\overline{CAS}$ Cycling, t_{PC}=t_{PC}(min.)	Standby	140	115	mA	1
I_{CC8A}			Active	185	130	mA	2
I_{CC9}	Pipelined Page Current	$\overline{RAS}$=V_{IL}, $\overline{CAS}$ Cycling, t_{PCP}=t_{PCP} (min.)	Standby	140	115	mA	1
I_{CC9A}			Active	185	150	mA	2
I_{CC10}	Pipelined Page Block Write Current	$\overline{RAS}$=V_{IL}, $\overline{CAS}$ Cycling, t_{PCP}=t_{PCP} (min.)	Standby	140	115	mA	1
I_{CC10A}			Active	185	150	mA	2

Notes:

1. SAM Standby: $\overline{SE}$ at V_{IH} and SC stable at either V_{IH} OR V_{IL}

2. SAM Active: $\overline{SE}$ at V_{IL} and tScc=tScc(min)

3. An initial pause of 200μs is required after power-up followed by 8 refresh cycles, 8 transfer cycles and 8 pulses or more of SC clock before proper device operation is achieved. In case of using internal refresh counter, a minimum of 8 $\overline{CAS}$-Before-$\overline{RAS}$ initialization cycles are required (Within the power-up and pause time, all control and address signals should be fixed high or low, or should be turned on in compliance with V_{CC}.)

AC Characteristics

#	SYMBOL	PARAMETER	HY5216257GE				UNIT	NOTE
			-60		-70			
			MIN.	MAX.	MIN.	MAX.		
1	tAA	Acess time from column address	-	30	-	35	ns	5
2	tASC	Cojumn address setup time	0	-	0	-	ns	
3	tASD	Column address to first SC high after $\overline{TRG}$ high (Read Transfer)	25	-	25	-	ns	12
4	tASR	$\overline{ROW}$ address setup time	0	-	0	-	ns	
5	tATH	$\overline{TRG}$ hold time from column address (real time data transfer)	18	-	18	-	ns	
6	tAWD	Column address to $\overline{WE}$ low delay time (RMW cycles)	50	-	60	-	ns	6
7	tCAC	Access time from $\overline{CAS}$ low (Extended Data Out mode)	-	17	-	20	ns	3,5
8	tCAC2	Access time, 1st bit from $\overline{CAS}$ low (pipelined page mode)	-	18	-	22	ns	
9	tCAH	Column address hold time	10	-	10	-	ns	
10	tCAL	Column address to $\overline{CAS}$ high delay time (Extended Data Out)	25	-	35	-	ns	
11	tCAS	$\overline{CAS}$ low pulse width	10	10,000	10	10,000	ns	
12	tCASP	$\overline{CAS}$ low pulse width (piplined page mode)	7	10,000	10	10,000	ns	
13	tCFH	DSF hold time from $\overline{CAS}$ low	10	-	10	-	ns	
14	tCHR	$\overline{CAS}$ hold time from $\overline{RAS}$ low (CBR cycles)	10	-	10	-	ns	
15	tCLZ	Output enable delay from $\overline{CAS}$ low	3	-	3	-	ns	3
16	tCOH	DQ output form $\overline{CAS}$ low	4	-	5	-	ns	
17	tCP	$\overline{CAS}$ precharge time	10	-	10	-	ns	
18	tCPP	$\overline{CAS}$ precharge time (pipelined page mode)	7	-	10	-	ns	
19	tCPA	Access time from previous $\overline{CAS}$ high (page mode)	-	35	-	40	ns	5
20	tCQD	$\overline{CAS}$ low QSF propagation delay time (early load)	-	30	-	30	ns	11
21	tCSD	$\overline{CAS}$ low to first SC high after $\overline{TRG}$ high (Read Transfer)	20	-	20	-	ns	12
22	tCSH	$\overline{CAS}$ low hold time from $\overline{RAS}$ low	53	-	60	-	ns	

AC Characteristics(continued))

#	SYMBOL	PARAMETER	HY5216257GE				UNIT	NOTE
			-60		-70			
			MIN.	MAX.	MIN.	MAX.		
23	tCSR	$\overline{CAS}$ setup time before /RAS low	0	-	0	-	ns	13
24	tCTH	$\overline{TRG}$ hold time from $\overline{CAS}$ low (Real Time Data Transfer	15	-	15	-	ns	
25	tCWD	$\overline{CAS}$ low to $\overline{WE}$ low delay time (RMW cycles)	37	-	45	-	ns	6
26	tCWL	Write low to $\overline{CAS}$ high delay (lead) time	15	-	15	-	ns	
27	tCWLP	Write low to $\overline{CAS}$ high delay (lead) time, pipelined page mode	7	-	10	-	ns	
28	tDH	Data hold time	12	-	15	-	ns	
29	tDS	Data setup time	0	-	0	-	ns	
30	tDZC	Data bus tristate to $\overline{CAS}$ low delay time	0	-	0	-	ns	10
31	tDZO	Data bus tristate (no input data) to $\overline{TRG}$ low delay time	0	-	0	-	ns	10
32	tFSC	DSF setup time before $\overline{CAS}$ low	0	-	0	-	ns	
33	tFSR	DSF setup time before $\overline{RAS}$ low	0	-	0	-	ns	
34	tMH	Write mask hold time $\overline{RAS}$ low	10	-	10	-	ns	
35	tMS	Write mask setup time before $\overline{RAS}$ low	0	-	0	-	ns	
36	tOEA	Access time from $\overline{TRG}$ low	-	15	-	20	ns	5
37	tOED	$\overline{TRG}$ high to data applied on data bus delay time	10	-	15	-	ns	
38	tOEH	$\overline{TRG}$ high time after $\overline{WE}$ low (to maintain tristate)	10	-	10	-	ns	
39	tOELZ	Output enable from $\overline{TRG}$ low	3	-	3	-	ns	
40	tOEZ	Output disable from $\overline{TRG}$ high	3	15	3	20	ns	6
41	tOFC	Output disable from $\overline{CAS}$ high	3	15	3	20	ns	
42	tOFR	Output disable from $\overline{RAS}$ high	3	15	3	20	ns	
43	tPC	Read or write page mode cycle time.	30	-	30	-	ns	
44	tPCP	Read or write page mode cycle time (pipelined page mode)	20	-	25	-	ns	
45	tPRWC	Read-modify-write page mode delay time	75	-	80	-		
46	tRAC	Access time from $\overline{RAS}$ low	-	60	-	70		

AC Characteristics(continued)

#	SYMBOL	PARAMETER	HY5216257GE				UNIT	NOTE
			-60		-70			
			MIN.	MAX.	MIN.	MAX.		
47	tRAD	$\overline{RAS}$ low to column address delay time	15	30	15	35	ns	3
48	tRAH	Row address hold time	10	-	10	-	ns	
49	tRAL	Column address to $\overline{RAS}$ high lead time	30	-	35	-	ns	
50	tRAS	$\overline{RAS}$ pulse width	60	10,000	70	10,000	ns	
51	tRASP	$\overline{RAS}$ pulse width (page or pipelined page mode)	60	100,000	70	100,000	ns	
52	tRC	Read or write cycle time	100	-	120	-	ns	
53	tRCD	$\overline{RAS}$ low to $\overline{CAS}$ low delay time	20	43	20	50	ns	7
54	tRCH	Read command hold time after $\overline{CAS}$ high	0	-	0	-	ns	8
55	tRCS	Read command setup time	0	-	0	-	ns	
56	tREF	Refresh period	-	8	-	8	ms	
57	tRFH	DSF hold time from $\overline{RAS}$ low	10	-	10	-	ns	
58	tROH	$\overline{RAS}$ low hold time from $\overline{TRG}$ low	15	-	15	-	ns	
59	tRP	$\overline{RAS}$ precharge time	35	-	45	-	ns	
60	tRPC	$\overline{RAS}$ high(precharge)to $\overline{CAS}$ low delay time	0	-	0	-	ns	14
61	tRQD	$\overline{RAS}$ low to QSF delay time (early load)	-	60	-	70	ns	11
62	tRRH	Read command hold time after $\overline{RAS}$ high	0	-	0	-	ns	8
63	tRSD	$\overline{RAS}$ low to first SC high after $\overline{TRG}$ high (Read Transfer)	60	-	70	-	ns	12
64	tRSH	$\overline{RAS}$ low hold time from $\overline{CAS}$ low	15	-	20	-	ns	
65	tRTH	$\overline{TRG}$ hold time from $\overline{RAS}$ low (real time data transfer)	45	-	55	-	ns	
66	tRWC	Read-modify-write cycle time	140	-	165	-	ns	
67	tRWD	$\overline{RAS}$ low to write $\overline{WE}$ low delay time (RMW cycles)	80	-	95	-	ns	6
68	tRWH	$\overline{WE}$ hold time from $\overline{RAS}$ low	10	-	10	-	ns	
69	tRWL	$\overline{WE}$ low to $\overline{RAS}$ high delay (lead) time	15	-	15	-	ns	
70	tSC	SC high pulse width	5	-	8	-	ns	
71	tSCA	SQ access time from SC high	-	15	-	20	ns	11

AC Characteristics(continued)

#	SYMBOL	PARAMETER	HY5216257GE				UNIT	NOTE
			-60		-70			
			MIN.	MAX.	MIN.	MAX.		
72	tSCC	SC serial cycle time	18	-	22	-	ns	
73	tSCP	SC slow pulse width	5	-	8	-	ns	
74	tSEA	Access time of SQ and QSF from /SE low	-	12	-	15	ns	11
75	tSEZ	Serial output buffer turn off delay from $\overline{SE}$ high	3	10	3	15	ns	
76	tSOH	Serial output hold time from SC high	4	-	5	-	ns	
77	tSOO	Serial output turn on delay from $\overline{SE}$ low	3	-	-	-	ns	
78	tSQD	SC high to QSF delay time	-	20	-	25	ns	11
79	tSRS	SC high to $\overline{RAS}$ low (CBRS, CBRR cycles only)	10	-	10	-	ns	
80	tSTH	Split transfer hold time ($\overline{RAS}$ high to SC high of active bank boundary location)	15	-	20	-	ns	
81	tSTS	Split transfer set up time (SC high of active bank boundary location to $\overline{RAS}$ low)	15	-	20	-	ns	
82	tTH	$\overline{TRG}$ hold time from $\overline{RAS}$ low	10	-	10	-	ns	
83	tT	Transition time, rise and fall	3	50	3	50	ns	1
84	tTP	$\overline{TRG}$ high pulse width	15	-	20	-	ns	
85	tTQD	$\overline{TRG}$ high to QSF delay time	-	25	-	25	ns	11
86	tTRP	$\overline{TRG}$ high to $\overline{RAS}$ low precharge time (late load data transfer)	tRP+5	-	tRP+5	-	ns	
87	tTS	$\overline{TRG}$ setup time before $\overline{RAS}$ low	0	-	0	-	ns	
88	tTSD	$\overline{TRG}$ high to first SC high delay time (Read transfer)	20	-	25	-	ns	12
89	tTSL	Last SC high to $\overline{TRG}$ high delay (lead) time (Read Transfer)	5	-	5	-	ns	12
90	tTRRH	$\overline{TRG}$ high to $\overline{RAS}$ high delay time	-10	-	-10	-	ns	
91	tWCH	Write command setup time	10	-	15	-	ns	
92	tWEZ	Output disable time from $\overline{WE}$ low (while $\overline{CAS}$ is high)	3	15	3	20	ns	
93	tWCS	Write command setup time	0	-	0	-	ns	9
94	tWP	Write command pulse width	10	-	10	-	ns	
95	tWSR	$\overline{WE}$ setup time before $\overline{RAS}$ low	0	-	0	-	ns	

Notes:

1. AC measurements assume t_T=3ns.

2. V_{IH}(min.) and V_{IL}(max.) are reference levels for measuring timing of input signals. Also, transition times are measured between V_{IH} and V_{IL}.

3. Operation within the t_{RAD}(max.), t_{RCD}(max.) limit ensures that t_{RAC}(max.) can be met. The t_{RAD}(max.) limit is specified as a reference point only. If t_{RAD} is greater than the specified t_{RAD}(max.) limit, then access time is controlled by t_{AA}.

4. If $t_{ASC} > t_{AA} - t_{CAC} - t_T$ and $t_{RAD} > t_{RAD}$(max.), the access time is controlled by the column address.

5. Measured with a load equivalent to 1 TTL loads plus 50pF and output reference level is V_{OH}/V_{OL} = 2V/0.8V.

6. t_{OEZ} define the time at which the output achieves the open circuit condition and is not referenced to output voltages levels.

7. t_{RCD}(min.) = t_{RAH}(min.)+2t_T+t_{ASC}(min.)

8. Either t_{RCH} or t_{RRH} must be satisfied for a read cycle.

9. t_{WCS} indicates early write operation, and not an operational limit point of the device. If $t_{WCS} \geq t_{WCS}$(min.), the cycle is an early is an early write cycle and data out pin will remain open circuit(high impedance) through the entire cycle.

10. Operation is generated whichever timing requirement t_{DZC} or t_{DZO} is met.

11. Measured with a load equivalent to 1 TTL load plus 30pF serial data out reference level is V_{OH}/V_{OL} = 2V/0.8V.

12. Except split register transfers.

13. t_{CSR} replaces both t_{CSR} and t_{CRP} quoted in specifications of other VRAM manufacturers.

14. Needed to avoid invoking a hidden refresh cycle.

Operation (Cycle Description, Timing Overview)

RAM Port

Read, Write, RMW with EDO mode

Read, write, and rdad-modify-write cycles are executed similarly to normal DRAM operations and are described by the detailed timing diagrams. However it is instructive to define here the mechanism of extended data output page mode.

Extended data output page mode differs from traditional fast page mode in that once $\overline{RAS}$ and $\overline{CAS}$ are brought low and $\overline{RAS}$ remains low, $\overline{CAS}$ has no tristate control over the random outputs (DQ pins). When $\overline{CAS}$ goes high, data corresponding to the previous $\overline{CAS}$ low remains valid on DQi. During page mode, this allows the accessing device more time to latch the data while $\overline{CAS}$ goes back high to set up the next page mode access.

WEL and $\overline{WEU}$ provide tristate control over the DQ pins and latch the data presented on the DQ pins and control the write operation. Whenever $\overline{WEL}$ or $\overline{WEU}$ is asserted, the all sixteen DRAM output bufffers will go to tristate.

Read, Write, with pipelined page mode

Pipelined page mode is selected using the DSF2 control input. In this mode, a pipeline register is activated in the page mode read data path. This substantially accelerates the peak data rate at the DRAM outputs while introducing one additional page mode cycle of latency before the first address is read out. Thus, to read out n column addresses, a total of n+1 $\overline{CAS}$ cycles must be performed. Pipelined page mode is ideal for performing long, high speed bursts of reads or bursts of writes within a page. During write operations, data is actually written during the page mode cycle in which it and its corresponding column address are latched, so no additional clock cycles are necessary to complete the last write operation. However to maintain compatibility with other implementations of pipelined page mode, it is recommended that one additional page mode cycle be performed at the end of each burst of pipelined page mode write operations. Pipelined page mode read cycles can be intermixed with pipelined page mode write cycles during the same $\overline{RAS}$ low period, however each burst of n reads, consuming n+1 $\overline{CAS}$ cycles, must be completed before initiating a burst write sequence. Likewise, when pipelined page mode reads follow a burst of n pipelined page mode writes, n+1 $\overline{CAS}$ cycles are recommended to complete the write operations. Following the n+1 $\overline{CAS}$ cycle, pipelined page mode read operation can begin, and the first data read out will occur on the second $\overline{CAS}$ cycle following the n+1 $\overline{CAS}$ cycle which completed the write operation. Figure 5 illustrates intermixed read and write cycles during pipelined page mode operation. Pipelined page mode is available for all DRAM read and write operation, including byte write, block write, and write per bit cycles. For details, see the timing diagrams.

Refresh

Refresh Requirements

The 4Mb Video RAM contains 512 row addresses which must be refreshed within each eight millisecond interval. On the 4Mb Video RAM, certain $\overline{CAS}$ before $\overline{RAS}$ cycles (CBRR and CBRS) are also used to enable or disable useful special functions as indicated in Table 2 and described below.

RAS Only Refresh

RAS only Refresh is performed as in conventional DRAMs, by supplying the address of the row to be refreshed on the falling edge of $\overline{RAS}$. Generally, $\overline{CAS}$, $\overline{WEL}$, $\overline{WEU}$, and $\overline{TRG}$ are negated throughout the RAS only refresh cycle. There are no specific timing constraints beyond normal DRAM refresh cycles.

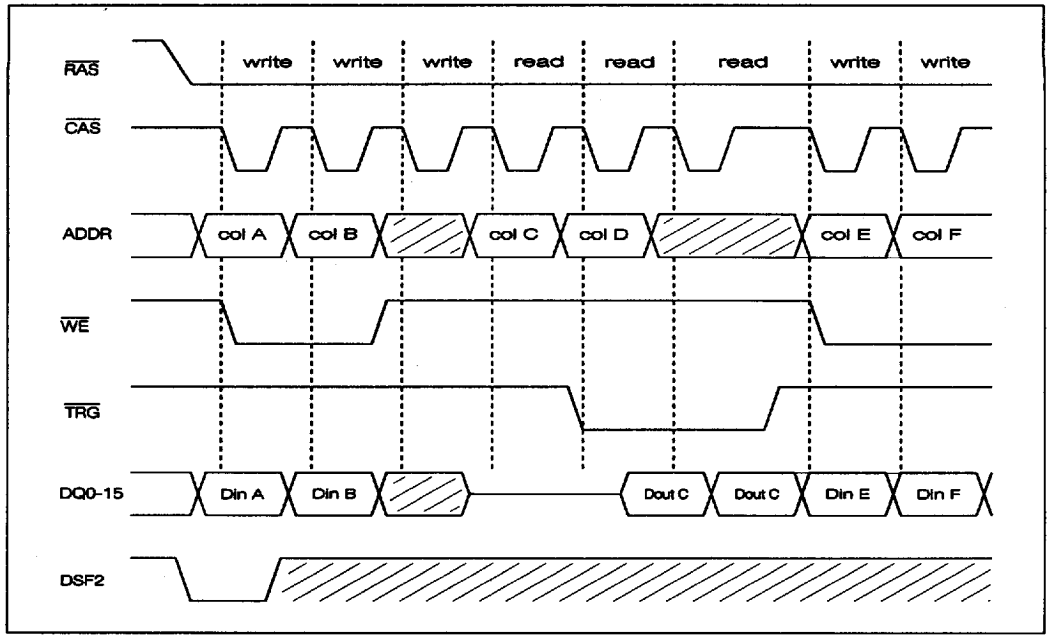


Figure 5. Write-Read-Write burst sequence in Pipelined Page Mode

Hidden Refresh

This cycle is performed identically to conventional DRAMs. Prior to a hidden refresh cycle, $\overline{RAS}$, $\overline{CAS}$, and $\overline{TRG}$ are asserted, causing data to appear at the DQ outputs. Subsequently, $\overline{RAS}$ is negated while both $\overline{CAS}$ and $\overline{TRG}$ remain asserted. $\overline{RAS}$ is again asserted and the row addressed by the internal CBR counter is refreshed. The hidden refresh cycle allows a useful refresh operation to be performed while data is waiting to be latched by the accessing device.

CBR Refresh, Normal Mode with Option Reset.(CBRR)

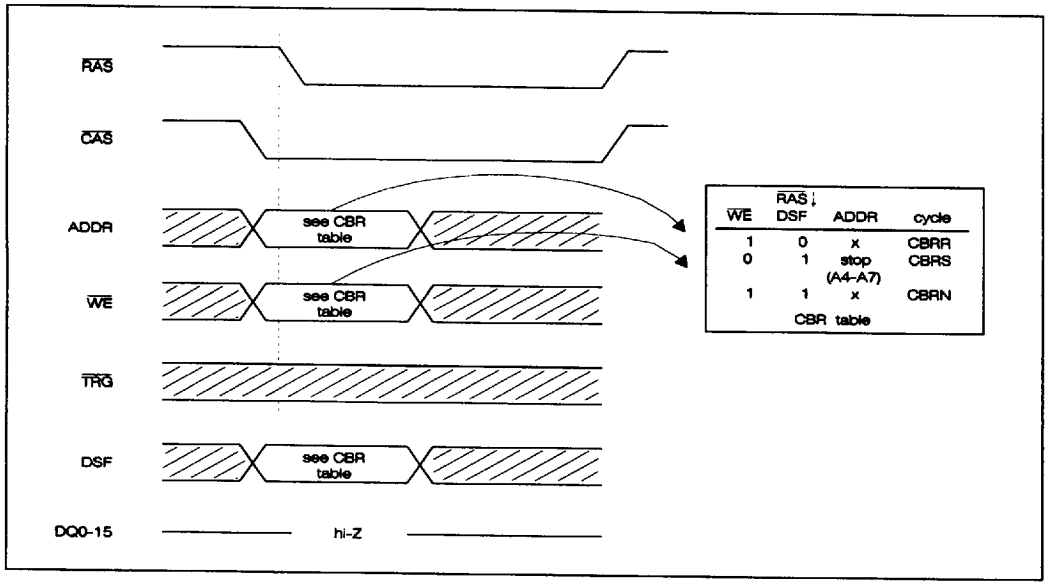
The 4Mb Video RAM has an internal 9-bit row address counter which is incremented on every CBR refresh cycle. During CBRS cycles, DSF(RAS) is low, so that two functions are accomplished. First, the row addressed by the CBR counter is refreshed. Second, the Video RAM unconditionally disables persistent write per bit and serial binary stop bit modes of operation. Persistent write per bit mode can be re-enabled by performing a Load Write Mask (LMR) cycle. Stop bit mode can be re-enabled by performing a stop mode enable CBR cycle(CBRS). Typically, a system will use either persistent or non persistent write per bit mode exclusively. Likewise, a system will typically use or not use stop bit mode all of the time. During power up, a CBRR cycle can be executed to clear both persistent mode and stop bit enable mode. From there, each mode can be selected individually by executing the LMR and CBRS cycles according to Table 2.

CBR Refresh; Persistent, Stop Bit Modes Not Reset..(CBRN)

Same as CBRR, except that DSF(RAS)=1, preventing persistent mode and stop bit mode from being reset. This CBRN cycle cannot be used to change the Video RAM from nonpersistent mode to persistent mode. That is accomplished by performing Load Write Mask Register(LMR) cycles.

CBR Refresh, Set Binary Stop for Split Register.(CBRS)

The CBRS cycle performs a CBR refresh and enables stop bit mode to be used in serial split register operation. The stop bits are supplied on the A4-A7 pins on the falling edge of $\overline{RAS}$.



WE	RAS↓ DSF	ADDR	cycle
1	0	x	CBRR
0	1	stop (A4-A7)	CBRS
1	1	x	CBRN

CBR table

Figure6. Timing diagram of CBR cycles

Write Per Bit

Write per bit mode allows the write of data to each of the sixteen RAM DQs to be controlled individually. A 16 bit write mask is supplied to the video RAM on its DQ lines at the falling edge of RAS. The mask is latched and stored on chip and is used to select which DQs will be written during write per bit cycles and which will not. If the mask bit for DQi is a 1. DQi will be written at write time during write per bit cycles. If the mask bit for DQi is a 0. DQi will not be written at write time during write per bit cycles.

Nonpersistent Write Per Bit

In nonpersistent write per bit mode, write per bit mode is selected by asserting either $\overline{WEL}$ or $\overline{WEU}$ on the falling edge of $\overline{RAS}$. Also at the falling edge of $\overline{RAS}$, the state of the DQ pins are latched and stored in the Video RAMS internal write mask register. At write time during the same cycle, the write mask is used as described in Section to enable write operation only to those DQs whose corresponding write mask bits are set to 1. Figure 7 illustrates the general timing and control of the nonpersistent write per bit write cycle. During page mode operation, the write mask is not changed. The same DQs are enabled on every page mode write cycle. The advantage of the nonpersistent write per bit mode is that the write mask can be loaded on every write per bit $\overline{RAS}$ cycle, providing maximum flexibility. The disadvantages are twofold. First, the timing is somewhat unlike conventional DRAMs, since the DQ lines must be driven with write control information as $\overline{RAS}$ is asserted. Second, and related to the first, the DQ lines cannot be "trimultiplexed" with the address lines, since the write mask would collide with the row address.

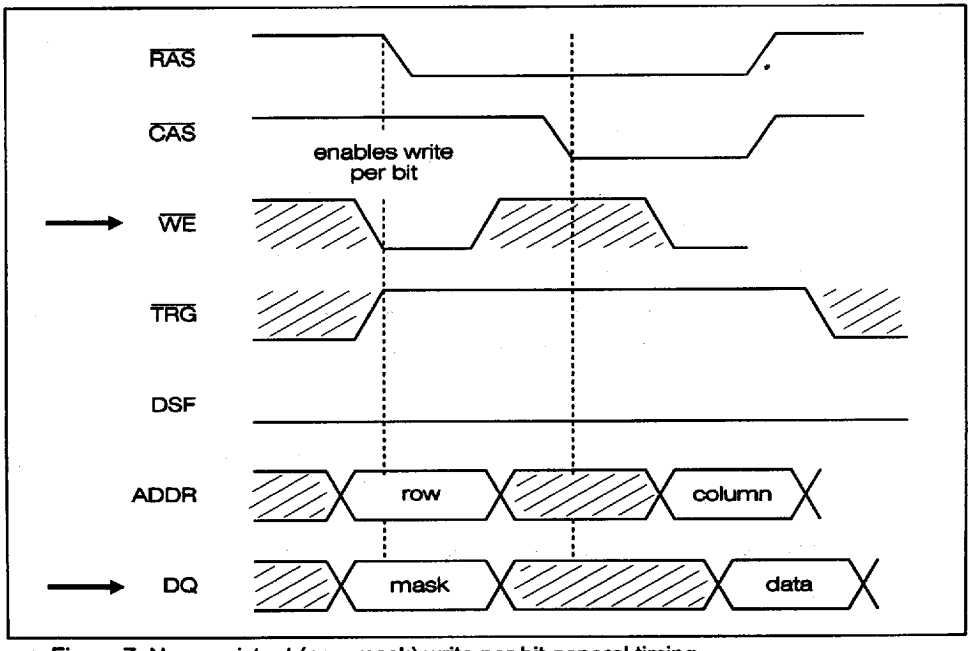


Figure 7. Nonpersistent (new mask) write per bit general timing

Persistent Write Per Bit

Often, the Video RAM user maintains the same write mask for many cycles. The persistent write per bit operation exploits this and resolves the two chief problems with nonpersistent write per bit operation. The general scheme is to split the write per bit function into two separate cycle types. The first cycle type (LMR) is used to load the write mask, using conventional DRAM timing and control. This enters the 4Mb Video RAM into the "persistent" mode.

Subsequent write per bit cycles are performed similar to nonpersistent mode, except that the write mask is not reloaded from the DQ lines when $\overline{RAS}$ falls. The write mask maintains the value it was written to during the previous write mask load (LMR) cycle. To change the write mask, another LMR cycle must be performed. Figure 8 illustrates the basic timing and control for the LMR cycle. Note that the write mask is written using conventional DRAM timing, except there is no required address. The destination of the write operation is the write mask register, not an address in RAM. The disadvantage of persistent write per bit mode is that an additional memory cycle is required each time the write mask is changed.

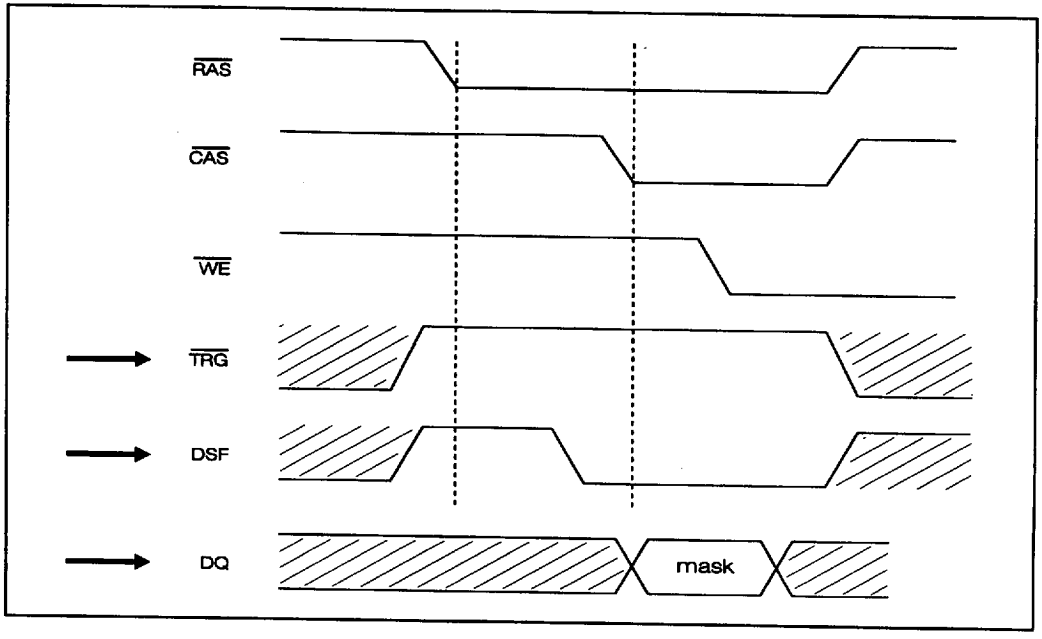


Figure 8. Persistent mode write per bit mask load (LMR) cycle.

Figure 9 shows a write per bit cycle being performed in persistent mode. Note that the state of the DQ lines is a don't care when RAS is asserted. There is no conflict between the row address and write mask, so the address and DQ lines can be tied together.

Persistent write per bit mode can be reset to nonpersistent mode by executing a CBRR.

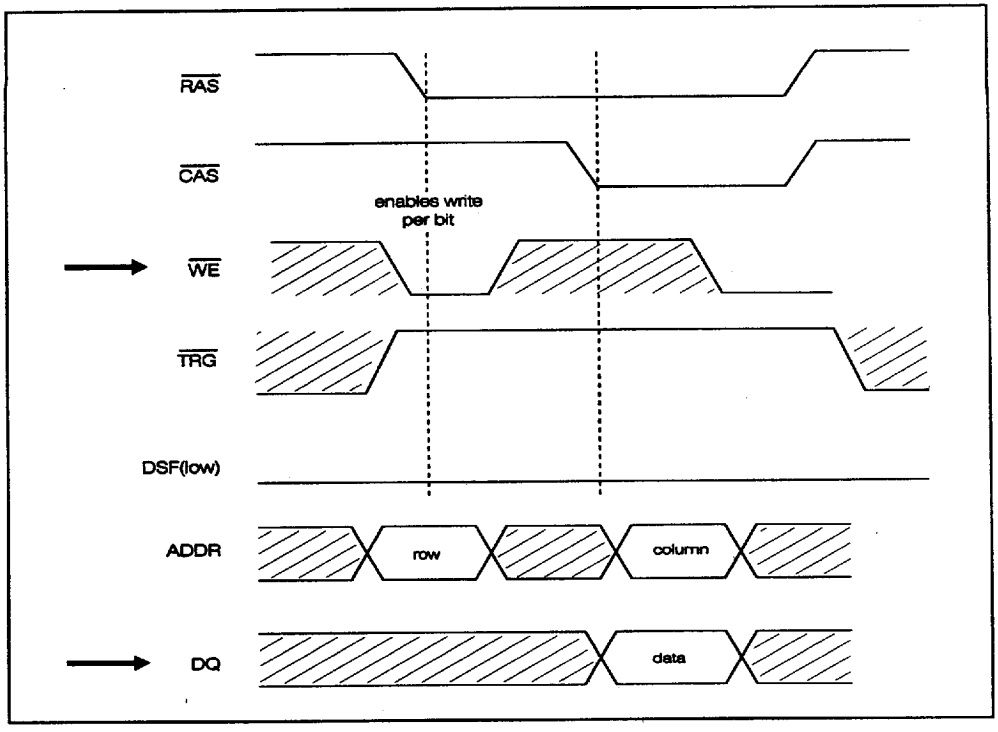


Figure 9. Persistent mode write per bit, write cycles

Byte Write

To facilitate byte operations, the 4Mb Video RAM is equipped with two write enable signals. $\overline{WEL}$ serves as the write enable for the lower byte. DQ0-DQ7, $\overline{WEU}$ serves as the write enable for the upper byte, DQ8-DQ15. Timing is similar to that of traditional single $\overline{WE}$ DRAM devices, with the following note worthy expections.

In the case where $\overline{WEL}$ and $\overline{WEU}$ are skewed, that is asserted or negated at different times, the write interval is defined by the assertion of the earlier of the two signals and the negation of the later of the two signals. In late write cycles, data setup time and data hold time are referenced from the earlier of $\overline{WEL}\downarrow$ or $\overline{WEU}\downarrow$. Also, if an early write cycle is invoked by asserting either $\overline{WEL}$ or $\overline{WEU}$ before $\overline{CAS}$ is asserted, then early write operation is invoked for both the high and low byte, with data setup time referenced to $\overline{CAS}$ for both bytes.

Byte write can be combined with write per bit. During write per bit operations, a DQ plane is written if and only if its corresponding write mask bit and byte write enables are asserted.

To invoke write per bit operation for both the high and low bytes, either $\overline{WEL}$ or $\overline{WEU}$ can be held low on the falling edge of $\overline{RAS}$ (See Table 2 and accompanying footnotes). If in non persistent mode, invoking write per bit will cause a new mask for both the high and the low bytes to be loaded from the DQ pins on the falling edge of $\overline{RAS}$. At write time, a write to each of the 16 bits will occur if and only if its corrsponding write mask register bit is set and its corresponding byte write signal ($\overline{WEL}$ or $\overline{WEU}$) is asserted. Figure 10 shows an example of byte write with and without write per bit invoked.

The dotted lines refer to write per bit operation. In this example, $\overline{WEL}$ alone controls the invokation of write per bit, although $\overline{WEU}$ could have been used either alone or with $\overline{WEL}$. Although write per bit is invoked according to the accompanying write per bit table, $\overline{WEU}$ is not asserted at write time and therefore no bits in the high byte will be written.

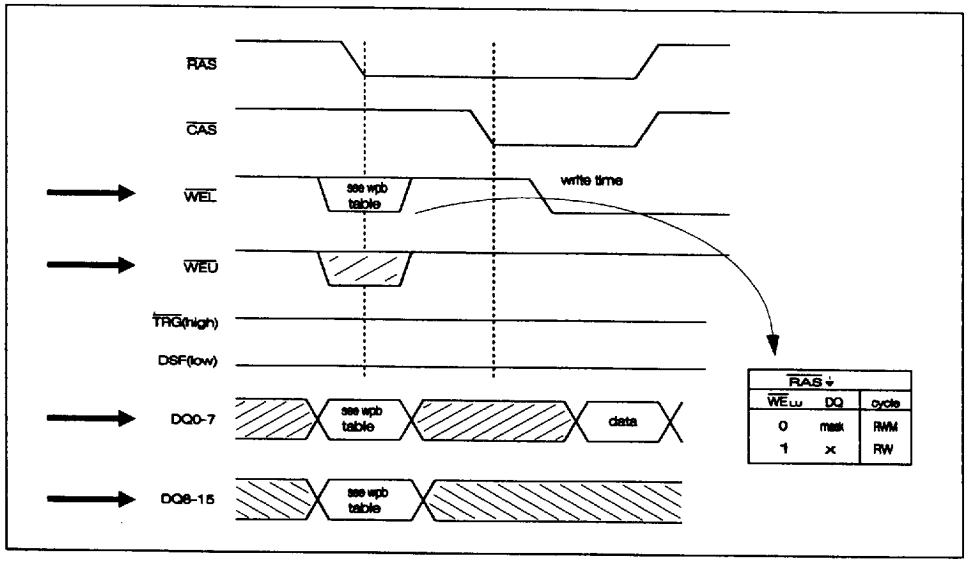


Figure 10. Example, byte write to low byte

Block Write

Block Write allows any subset of eight column location in RAM, chosen by A3-A8, to be written in a single write cycle using the contents of an on chip color register. The one-of-eight column select function normally performed by cloumn addresses A2-A0 is replaced by an "n of eight" (n=1 throulgh 8) column select using DQ0-7 for the low byte and DQ8-15 for the high byte as an address mask. The Block Write thus gains write access to 16x8=128 bits of information. The application example in Figure 11 shows the 16 DQs of the 4Mb Video RAM split into two(8 bit per pixel)sets. So each 16 bit "word" represents an even and odd pixel pair.

The eight column addresses *2 pixels per address = 16 pixels are mapped to 16 contiguous horizontal pixel locations on the display. By using an address mapping scheme known as "tiling," it is possible to combine such groups of pixels into multiple rows, creating a rectangular grid of pixels all mapped to a single page of the DRAM as shown. The graphics controller can mainpulate the pixels in the grid very quickly by operating in page mode. Each page mode cycle accesses on horizontal group of 16 pixels, that group consisting of 8 pixel pairs. Using Block Write, a write operation can be performed to any subset of the 8 pixel pairs using the contents of the color register. Block write is fully compatible with byte write and write per bit operation.

The individual pixels in each pixel pair can be write controlled (written or not written) by either byte write operation(using $\overline{WEL}$ and $\overline{WEU}$) or by using write per bit mode. Thus, by combining block write with either bype write or write per bit, any subset of the 16 pixels accessed on each page mode cycle can be written. With a little imagination, we can see how patterns(eg characters, line segments) within localized rectangular arrays of pixels can be quickly manipulated.

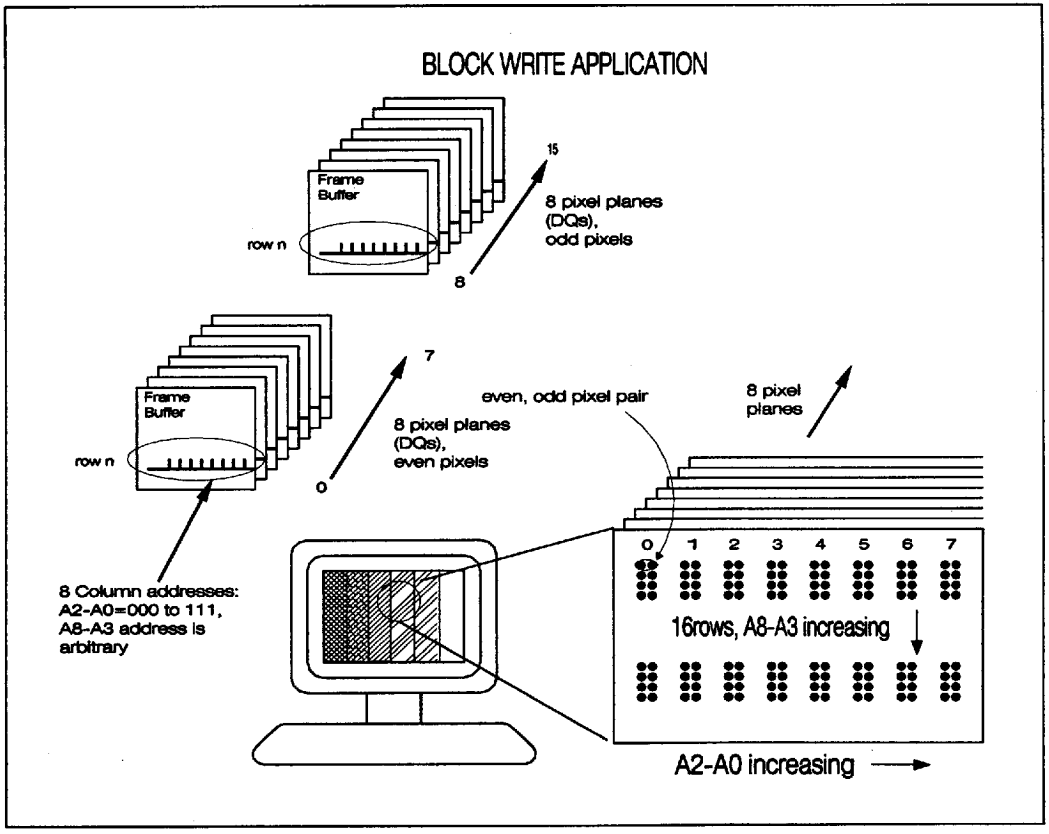


Figure 11. Application of the Block Write

Load Color Register

To perform block write cycles, the data to be written to the multiple column locations must first be loaded into the on chip color register using a Load Color Register(LCR) cycle. The timing and control is similar to a normal DRAM write cycle. The important differences are the values of DSF(RAS), DSF(CAS) and the fact that the address information supplied to the chip is ignored. Figure 12 illustrates the Load Color Register cycle using late write timing. The Color Register can also be loaded using early write timing. Also byte write is supported, allowing individual write control of each byte in the color register. The arrows in Figure 12 are intended to show that the writes of the high and low bytes are controlled by $\overline{WEL}$ and $\overline{WEU}$ respectively.

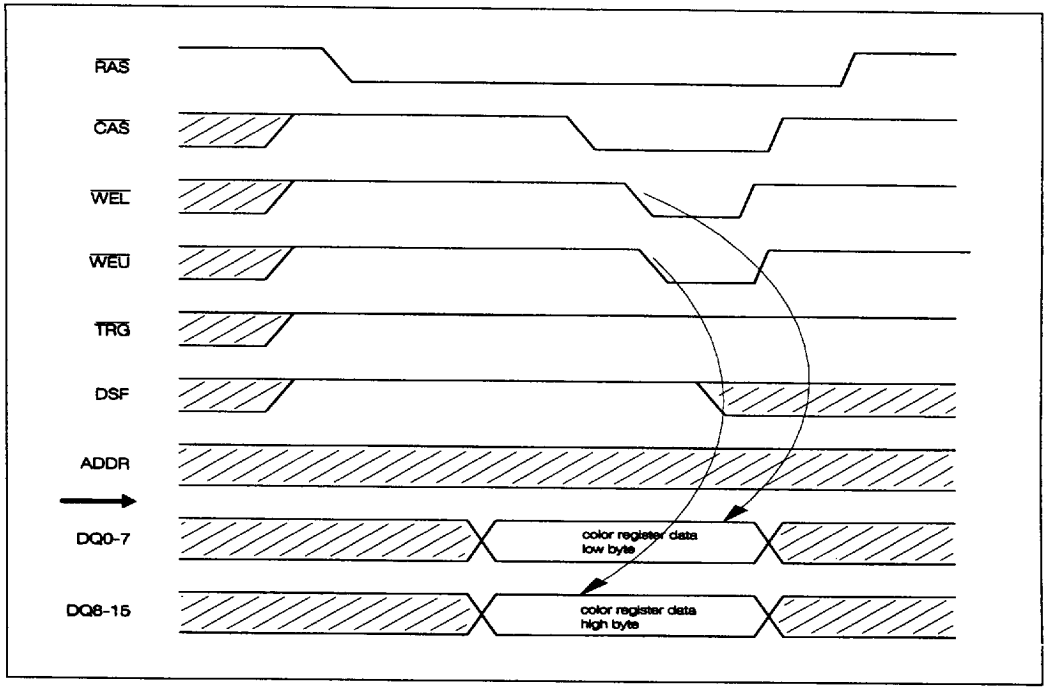


Figure 12. Timing of the Load Color Register Cycle, late write timing

Block Write Cycles

One the desired data is loaded into the color register, the block write cycles can be performed. Figure 13 shows a sequence of page mode block write cycles. Block write can be combined with write per bit, as denoted by the dotted lines. Either $\overline{WEL}$(RAS)=0 or $\overline{WEU}$(RAS)=0 is sufficient to enable write per bit mode to both the low and the high byte. In this case, nonpersistent mode write per bit is being used, however persistent mode write per bit operation can also be used with block write. Consequently, block write can selectively write any DQ and any combination of eight contiguous column locations(selected by column address A8-A3) in a single CAS cycle. AS many as 8 ×16=128 bits of data can be written per CAS cycle in full page mode or pipelined page mode.

In each CAS cycle, the state of DSF is latched on the falling edge of CAS. DSF(CAS)=1 enables block write operation while DSF(CAS)=0 disables block write operaton. DQ0-DQ7 is latched at write time, defined as the later of $\overline{CAS}$ falling and either $\overline{WEL}$ or $\overline{WEU}$ falling. DQ0-DQ7 are used as an eight bit address mask for the lower byte only, providing individual write control to the block of eight column addresses defined by A3(CAS)-A8(CAS). The eight least significant DQ pins replace the address inputs A0-A2. For example, if DQ0(write time)=1, then column A2-A0=000b, lower byte will be written with the data stored in the lower byte of the color register. If DQ0(write)=0, then column A2-A0=000b, lower byte will not be written with the data stored in the lower byte of the color register.
Thus:

for i=0-7
DQi(write)=1 enables write to address A2-A0=i, low byte. DQi(write)=0 disables the write to A2-A0=i, low byte.

Likewise, DQ8(write)-DA15(write) are used as an eight bit address mask for the upper byte only, providing individual write control to the block of eight column addresses defined by A3(CAS)-A8(CAS).

for i=8-15
DQi(write)=1 enables write to address A2-A0=i, high byte. DQi(write)=0 disables the write to A2-A0=i, low byte.

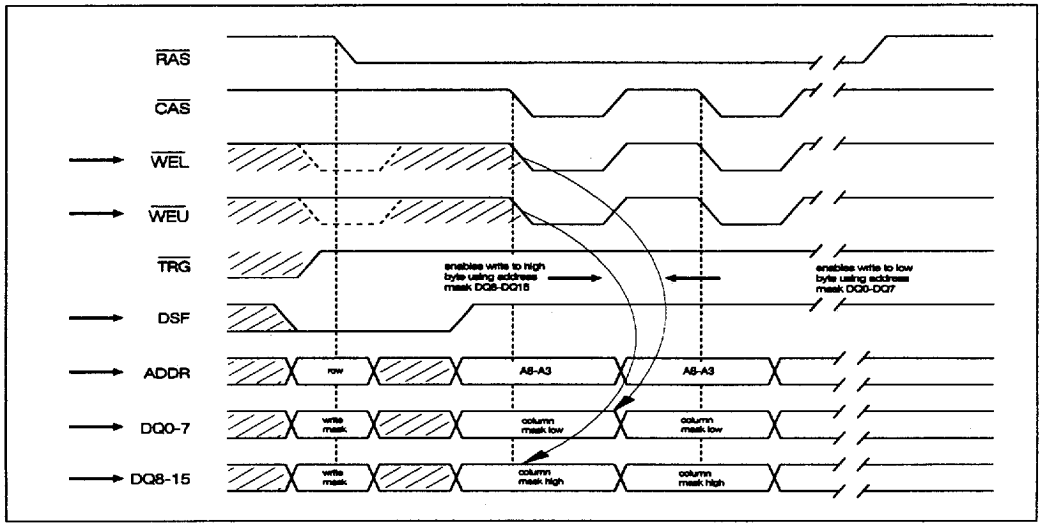


Figure 13. Timing of block write, with and without write per bit.

Flash Write(non standard feature, available at customer request)

Flash Write allows all 512 columns of an addressed row to be written in a single minimum RAS cycle time using the contents of the on chip color register. Flash Write is unconditionally masked with respect to each DQ in RAM using the contents of the Write Mask Register either in persistent or in non persistent mode. If the Write Mask bit for DQi is set to "1", then during the Flash Write cycle all columns in the addressed row for DQi will be written using the contents of bit "i" in the color register. A simplified timing diagram of the Flash Write cycle is shown in Figure 14. Flash Write is invoked by holding $\overline{CAS}$, $\overline{TRG}$, and DSF high and either $\overline{WEL}$ or $\overline{WEU}$ low during the falling edge of $\overline{RAS}$. If the device is operating in nonpersistent mode, the write mask must be supplied on the DQ pins during the falling edge of $\overline{RAS}$. If the device is operating in persistent mode, the write mask must be previously loaded using a Load Mask Register(LMR) cycle as described. The color register is loaded using a Load Color Register (LCR) cycle as described.

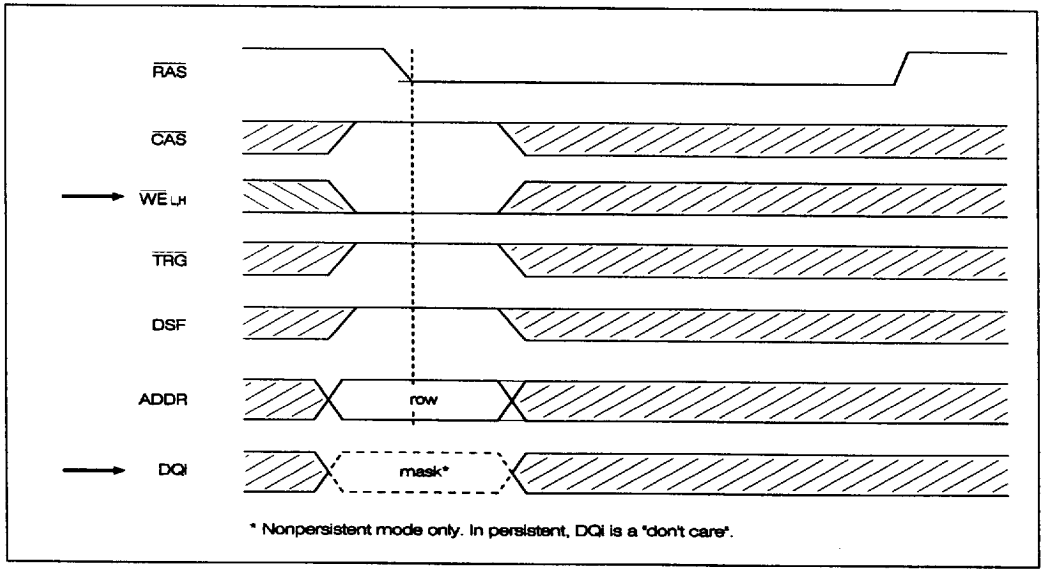


Figure 14. Masked Flash Write general timing

Transfer Operation, RAM to SAM

Periodically, data will need to be transferred from a specified row in RAM to the SAM in order to provide data to the video display system. This is referred to as a *transfer, a transfer read,* or *memory-to-register transfer* operation. These terms will be used interchangeably in this specification unless otherwise noted.

Discussion of Address Mapping

Half SAM architecture

To facilitate discussion of the SAM serial architecture, the reader should review the definition of terms and conventions.

During normal transfer cycles in which address A8=0 on the falling edge of $\overline{CAS}$, data from column addresses 0-255 are loaded from the selected row in RAM to the SAM. During normal transfer cycles in which address A8=1 on the falling edge of $\overline{CAS}$, data from column addresses 256-511 are loaded from the selected row in RAM to the SAM. The selection of the upper or lower 256 columns is shown in Figure 15.

Figure 15 also illustrates the internal mapping in RAM required for the half SAM architecture. The columns in RAM are interleaved with respect to A8, creating a sequence of:

n, n+256 for n=0 to 255

From an internal chip design standpoint, we will see that this internal mapping is modified when the Video RAM is operated in binary stop bit mode. To support binary stop bit mode, an internal(transparent to user) column address bit swap between column addresses A8 and A7 is invoked. The effect of this swap is to reverse the physical positions of quad rows 2 and 3 in both RAM and SAM. This address swap is required to maintain socket compatibility with full SAM devices.

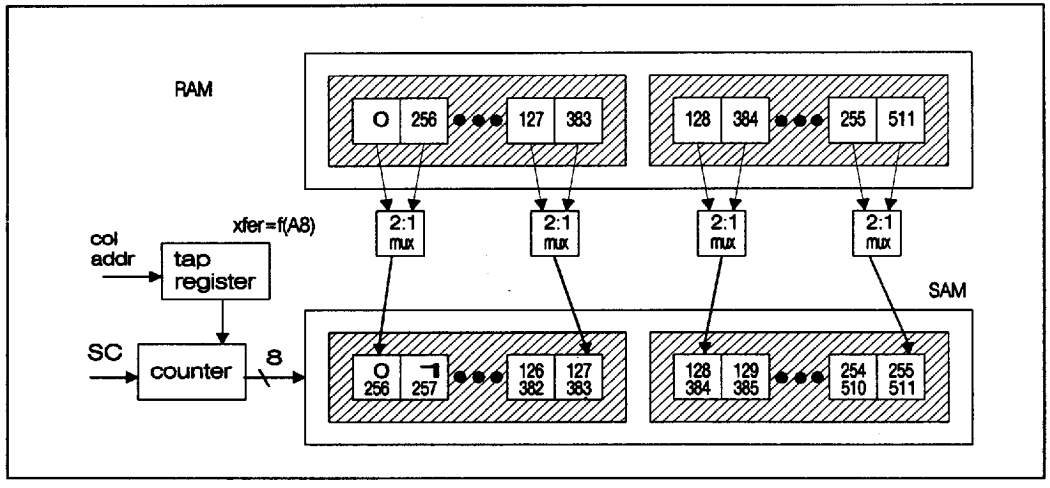


Figure 15. Half SAM architecture and mapping to RAM

Overview of transfer operations

There are five distinct types of transfer operations:

1. normal transfer
2. read time transfer and real time transfer with late load
3. early transfer(or early load)
4. split register transfer, for use in split register serial operation
5. split register transfer, for use in split register serial operation, with binary stop bit mode enabled

Normal Load

Figure 16 illustates the sequence of operations for a normal transfer read. To transfer data to the register from DRAM, the DRAM begins to access the required row of data similar to the way it would in a normal read cycle. On the falling edge of $\overline{CAS}$, A8 is latched and determines whether the lower 256 column will be loaded into the SAM(A8=0) or the upper 256 columns will be loaded into the SAM(A8=1).

Also on the falling edge of $\overline{CAS}$, the starting location, or tap, is loaded into an on chip tap register from column address pins A0-A7. Subsequently, when $\overline{TRG}$ goes high, the data are transferred to the serial register and the serial counter is reloaded from the tap address register. The data at the serial output does not change until the first positive edge of the serial clock after $\overline{TRG}$ goes high.

There are historically six key timing parameters that must be met during transfer cycles, all relating to the physical transfer of information to the SAM and the clocking of the SAM register directly before and after the physical transfer. These are shown in Figure 16 and are discussed below.

t_{TSL}, or last SC↑ of old register contents to $\overline{TRG}$ ↑, guarantees the last bit will appear out of the SAM from the old register contents before the transfer and will remain until the next SC↑.

t_{TSD}, or $\overline{TRG}$ ↑ to first SC↑ of new register contents after transfer, guarantees that the transfer is completed in sufficient time to guarantee the serial data access time for the first access after the transfer.

t_{RTH}, or $\overline{RAS}$↓ to $\overline{TRG}$↑, guarantees that the data from the half row to be transferred is safely latched on the bit lines before transfer begins.

t_{CTH}, or $\overline{CAS}$ ↓ to $\overline{TRG}$↑, guarantees that the tap address is safely latched so the access of the first address after transfer is not delayed.

t_{ATH}, or column address valid to $\overline{TRG}$↑, also guarantees that the tap address is safely latched so the access of the first address after transfer is not delayed.

t_{TRRH}, or $\overline{TRG}$ ↑ to $\overline{RAS}$↓, guarantees that the transfer will be completed before the end of the $\overline{RAS}$ low time.

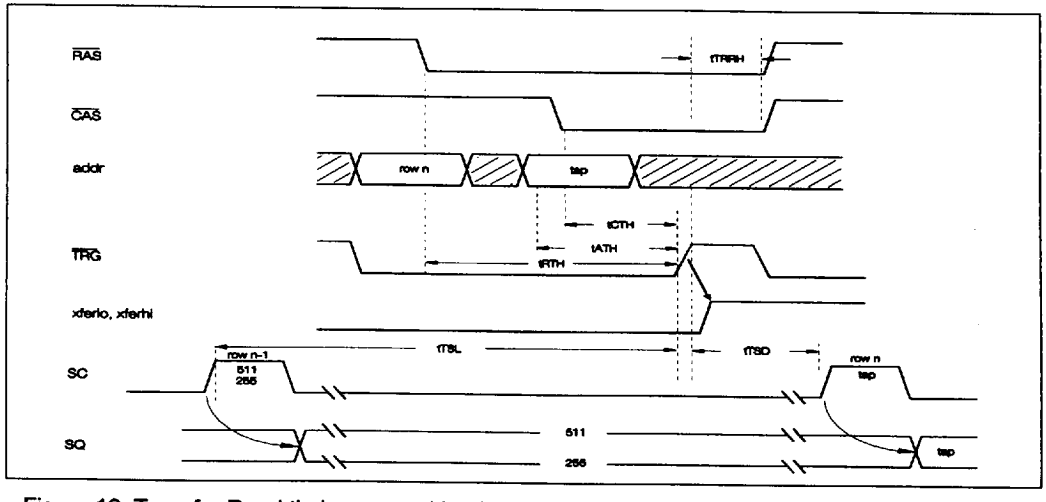


Figure 16. Transfer Read timing, normal load

Note that the physical transfer of data from RAM to SAM must be performed between two serial clock positive edges. Under normal load conditions, this time interval, defined mostly by t_{TSL} + t_{TSD} is generally large and poses no significant synchronization problems. For some applications, however, this time interval will become very narrow, and synchronization problems will occur. To circumvent these problems, this Video RAM supports both *real time load* and *split register load*.

In addition to the problem of synchronizing the serial clock to the transfer operation, the parameters t_{RTH}, t_{CTH}, $_{ATH}$, and t_{TRRH} have historically caused syncronization problems by placing significant timing constraints on $\overline{TRG}\uparrow$, particularly when the $\overline{RAS}$ low time is near the minimum spec value. To relax the timing constraints in this regard, this Video RAM supports *early load* and *late load* transfer cycles.

Real time transfer

Real time transfer, or real time load, is a simple extension of normal load. It is intended to allow continuous serial data streams while the transfer is occuring, but places most of the burden of synchronization on the user. Basically, the real time load is a normal load in which the serial clock, SC, continues to clock data during the transfer cycle both before and after $\overline{TRG}\uparrow$. In other words, the parameters t_{TSL} and t_{TSD} are taken to their specification limits, as shown in Figure 17.

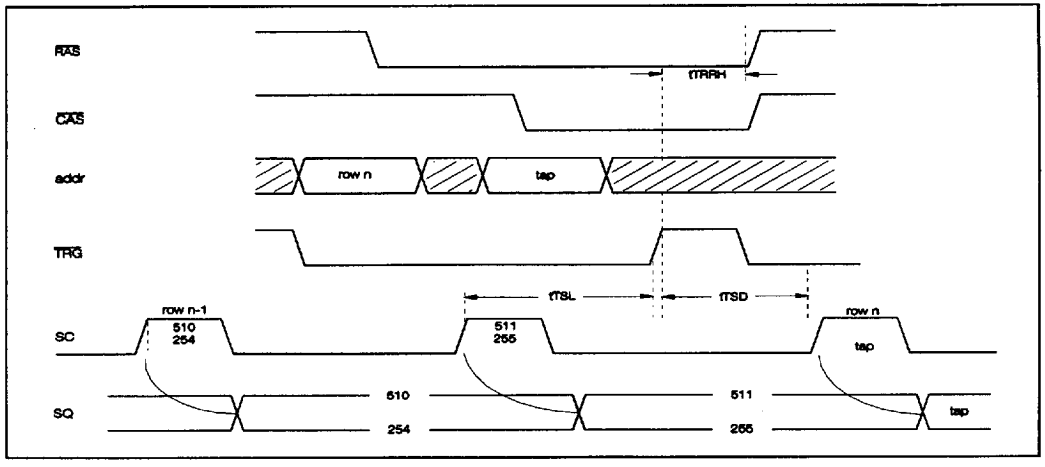


Figure 17. Transfer Read timing, real time load

Real Time Transfer with Late Load

In the Real time transfer with late load, the t_{TRRH} parameter is eliminated, allowing $\overline{TRG}$ to go high before, coincident with, or after $\overline{RAS}$ goes high. Internally, $\overline{RAS}$ is prevented from going high, since the data from the RAM must remain on the bit lines until after $\overline{TRG}\uparrow$ and the transfer has completed. Since the precharge cycle cannot begin until the internal RAS signal has been negated, parameter t_{TRP} or $\overline{TRG}\uparrow$ to $\overline{RAS}\downarrow$ is imposed on the user and is set equal to the RAS precharge time plus any additional time necessary to complete the transfer before the row and sense clocks can be reset. Figure 18 shows the timing and a *simplified* logic implementation of the late load, including important internal signals. See the Real Time and Late Load timing diagram, Figure 38, for more datails.

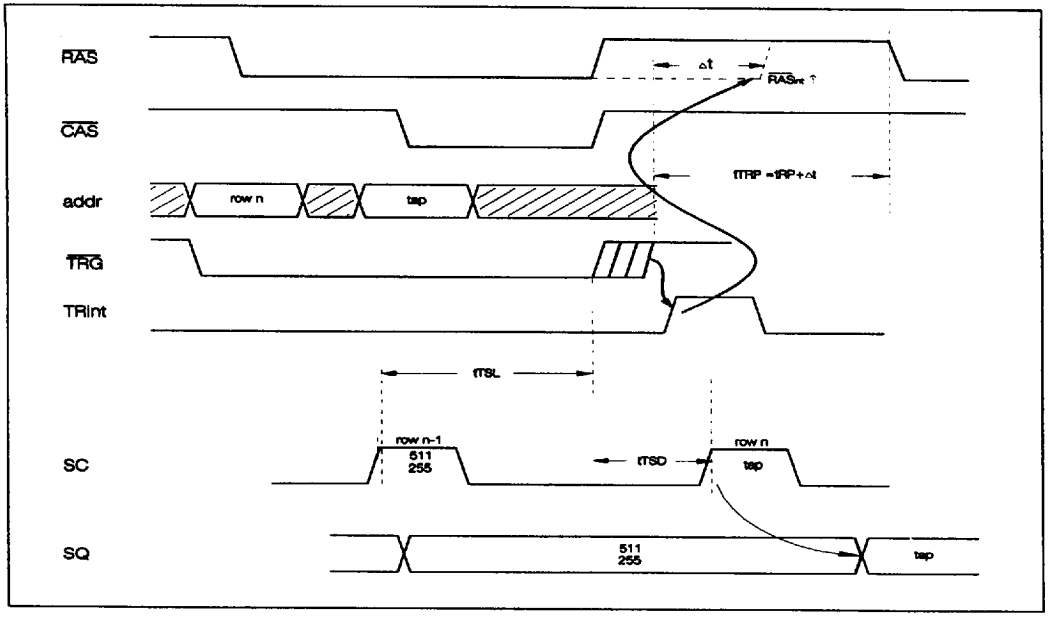


Figure 18. Real time transfer, late load timing and control.

Early Register Load

The early load cycle is an extension of the normal load cycle and is intended for transfers where the time interval of the two clock pulses which surround $\overline{TRG}\uparrow$ is large. To simplify the timing interface with respect to $\overline{TRG}\uparrow$, $\overline{TRG}$ is allowed to go high after a minimum hold time from $\overline{RAS}\downarrow$. This hold time is identical to the row address hold time. so the graphics processor can treat $\overline{TRG}$ like an address pin. Internally $\overline{TRG}\uparrow$ is "held off," or prevented from occuring, until data from the row in DRAM is safely sensed and latched onto the bit lines. Figure 19 illustrates the timing and control of the early load cycle. The timing parameter t_{TSL} is retained. Also, t_{RSD}, t_{CSD}, and t_{ASD} are introduced. t_{RSD} guarantees that the first serial clock pulse accessing the new register contents does not occur before the internally synchronized transfer operation is completed. Likewise, t_{CSD}, or $\overline{CAS}\downarrow$ to first SC↑ after $\overline{TRG}\uparrow$, and t_{ASD}, or colunm address to first SC↑ after $\overline{TRG}\uparrow$, guarantee that the tap address is loaded sufficiently before the transfer occurs so that the correct first address after the load can be read out with no access time penalty. See the Early load timing diagram, Figure 37, for more details.

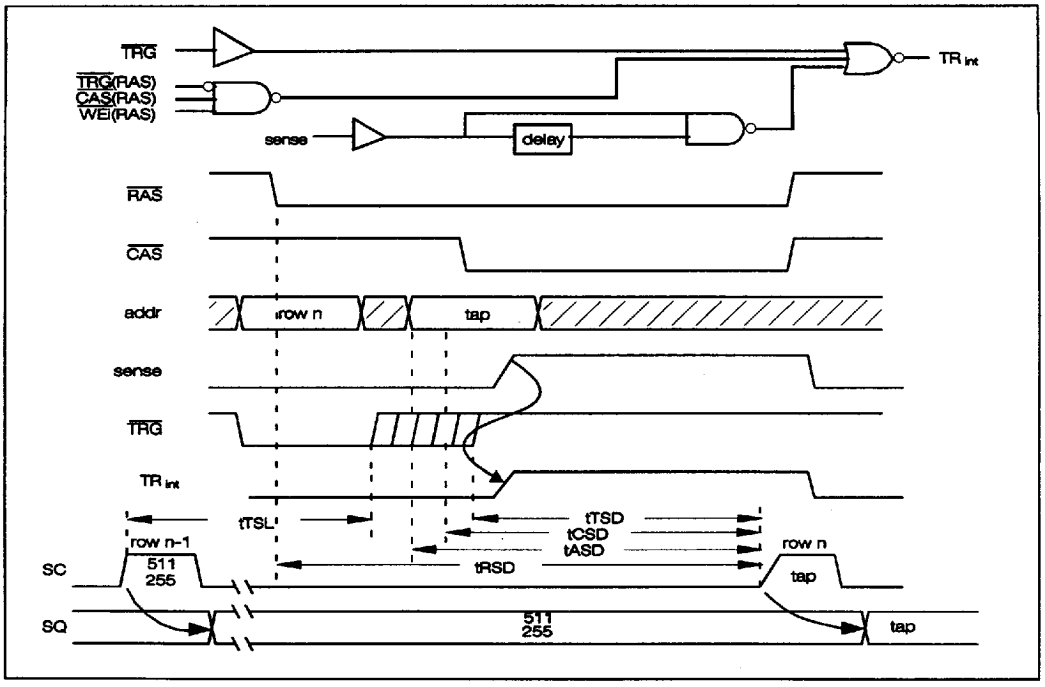


Figure 19. Transfer read, early load

Split Register Mode Transfer

Sequence of Operations

Whenever the length of the SAM (number of column addresses) mapped to the display is not a sub multiple of the number of pixels per scan line on the CRT, it is possible that SAM will run out of data before the CRT scans the last pixel on a line. Serial data rates are generally too fast for the SAM to reload from the RAM and still get the next pixel to the display in time, even using real time transfer cycles. To eliminate this problem, the 4Mb Video RAM is designed with the SAM split into two banks, with each bank corresponding to one quad row of RAM. The architecture allows one bank of the SAM to be loaded from a quad row in RAM while the other bank reads data from a different quad row in RAM to the SAM output port. After the active quad row delivers its most significant of data to the SAM output port, control is automatically transferred to the other SAM bank, which now becomes the active bank, and serial access is continued from the tap location selected during the load of that quad row into that SAM bank. Figure 20 illustrates the sequence of transfer loads and serial access in the split register mode

of operation for the case when binary stop bit mode is disabled.

As shown in the figure, the typical sequence of operations is as follows:

SAM port access	Transfer RAM to SAM
....	
read row n, quad row 0 from low SAM bank	load row n, quad row 1 into high SAM bank
read bit 127	
switch to high bank	
read bit 128 (or tap)	
read row n, quad row 1 from high SAM bank	load row n, quad row 2 into low SAM bank
read bit 255	
switch to low bank	
read bit 256 (or tap)	
read row n, quad row 2 from low SAM bank	load row n, quad row 3 into high SAM bank
read bit 383	
switch to high bank	
read bit 384 (or tap)	
read row n, quad row 3 from high SAM bank	load row n+1, quad row 0 into low SAM bank
repeat for all rows...	

This example assumes that the tap point is always set at the least significant column address of each quad row, however any tap location can be chosen. During bank switches, control transfers from the most significant address of the one bank to the tap location of the other bank.

Timing and Control

Figure 21 shows a simplified timing diagram and related circuitry for a split register transfer cycle. The transfer into the low (high) bank of the SAM is performed well in advance of the serial access of the most significant address in the high (low) bank.

In this example, the high bank is reading quad row 1 to the serial output(SQ0-SQ15) while the low bank has already read out and still holds data from quad row 0 of the same row. The left hand timing diagram in Figure 21 shows the transfer cycle which loads quad row 2 of the same row into the low bank. The transfer cycle is performed with column address A8 set to 0 and occurs serveral clock cycles before the most significant address of the high bank, in this case address 255, is read out. During the transfer cycle, the tap register is loaded with the column address information supplied from A0-A7 when $\overline{CAS}$ falls. The high bank continues to be read out serially until address 255 is accessed. On the rising edge of the clock pulse which sends address 255 to the serial output, the counter is not incremented but instead is loaded from the tap register. Only bits A0-A6 are loaded from the tap register during split register mode. A7 is loaded from a toggle flip flop, as shown in Figure 21. During split register mode only, this toggle flip flop serves as a bank pointer, both to select which bank is active for output to the serial port and to determine which bank will be loaded on the next split register transfer cycle. On the next serial clock cycle, the tap location of the low bank is read to the serial output.

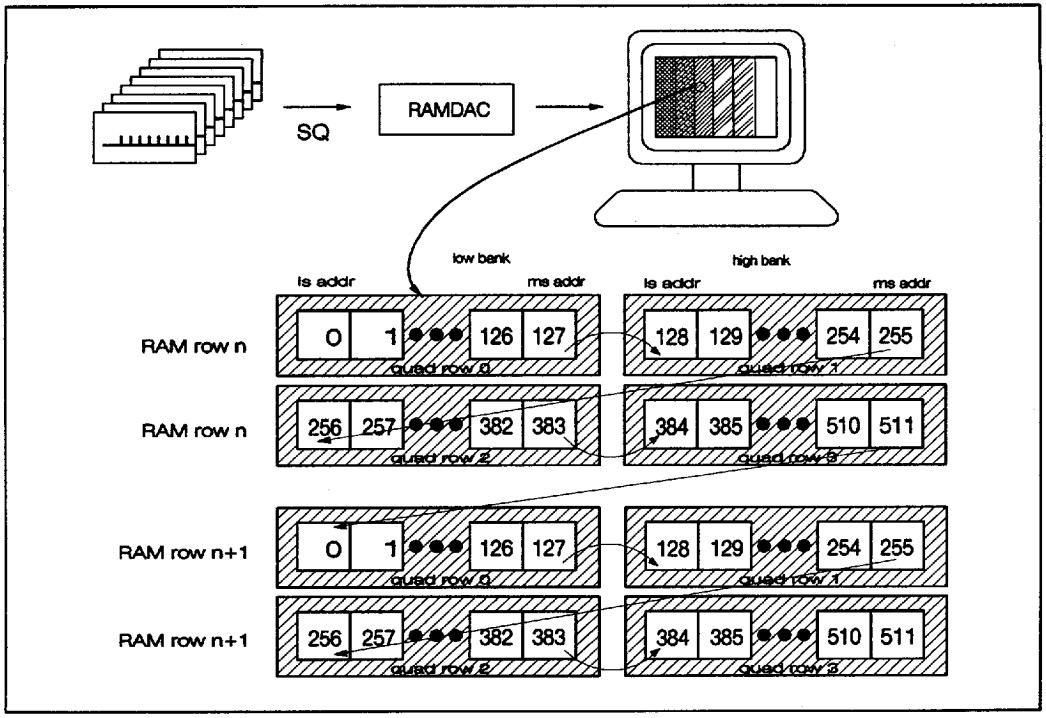


Figure 20. Split register transfer sequence and address mapping in the SAM

The toggle flip flop can be preset only during non-split register transfer cycles. This means that before any split register transfer cycles are executed, at least one non-split register transfer must be done. This is required to tell the SAM which bank to begin reading data from, since in split register transfer cycles the A7 pin is ignored and substituted for with the internal toggle flip flop.

QSF Output

QSF is an output which is intended primarily for use in split register mode to communicate to the user which SAM bank is actively delivering data to the SAM output port and which bank will be reloaded next. QSF is simply a buffered output of the most significant bit of the serial counter (bit C7), Whenever this counter bit changes state, either by incrementing the counter or by loading a new tap address into the counter, QSF will toggle. These two cases are considered below.

CASE 1. Split register mode disabled

During serial access (ie, no transfers in progress) QSF will change state *after* the SC↑ corresponding to the last address read in the active bank *before* the bank switch. In the case where split register mode is disabled, QSF changes after the SC↑ which accesses the most significant address of the active bank before the bank switch. The *next* SC↑ will access the least significant address in the opposite bank.

QSF will also change state *during* memory to register transfer cycles in which the tap address causes a jump to the opposite bank. The transition follows $\overline{TRG}\uparrow$ as shown in the Early and Real Time register transfer timing diagrams.

CASE 2a. Split register mode enabled, binary boundary stop bit mode disabled

During split register mode with binary stop bit disabled. QSF changes state during serial operation similar to normal register mode: after the SC↑ which accesses the most significant address in the active bank before the bank switch. However, the *next* SC↑ will access the tap address in the opposite bank. This tap address was loaded during the memory to register transfer cycle of the inactive bank. An example of this sequence is shown in Figure 21. The rising edge of the serial clock pulse which sends the most significant address (address 255 in the example) to the serial output also causes QSF to switch from high to low, indicating that the *next* SC↑ will access data from the tap address in the *lower* bank.

QSF does not change state after $\overline{TRG}\uparrow$ during memory to register transfer cycles when split register mode is enabled.

CASE 2b. Split register mode enabled, binary boundary stop bit mode enabled

If binary stop bit mode is enabled the QSF output will change state on the SC↑ which accesses the specified boundary address in the active bank. The *next* SC↑ will access the tap address in the opposite bank. This tap address was loaded during the memory to register transfer cycle of the inactive bank.

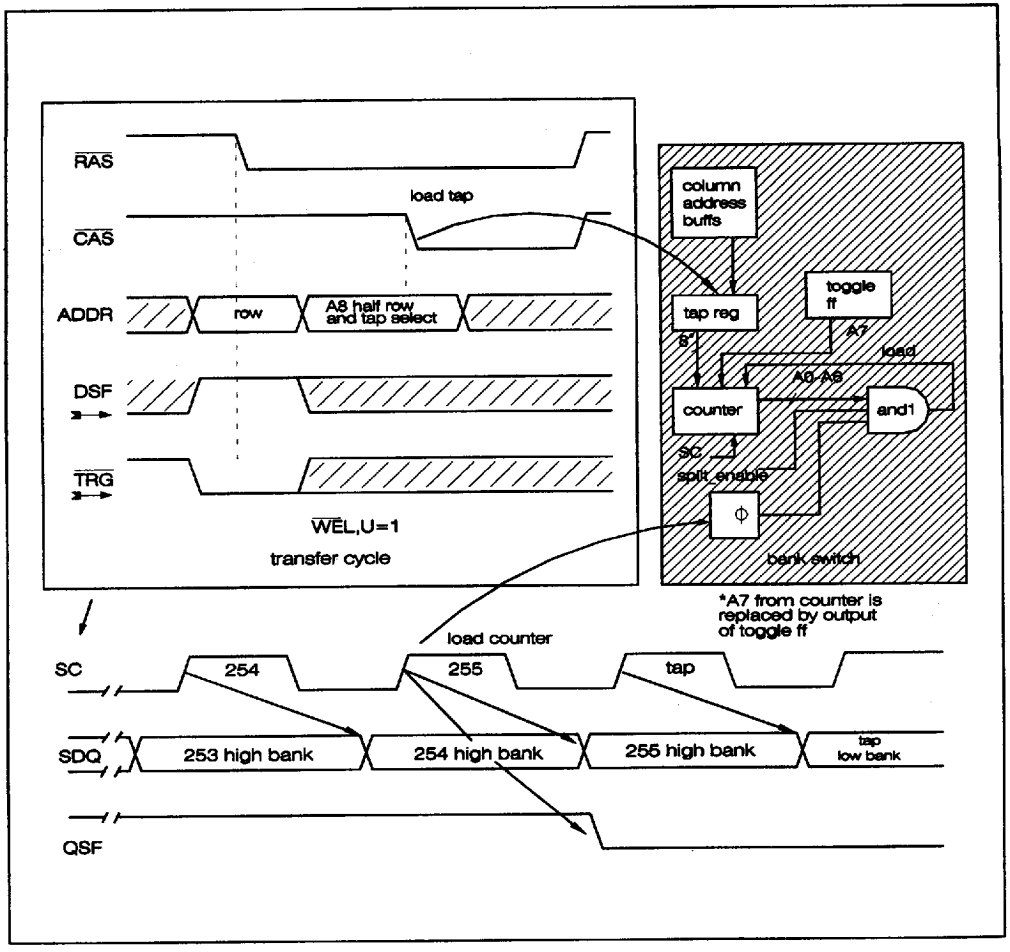


Figure 21. Split register transfer cycle timing and bank switch control

Split register mode with binary stop bit

System application

Common graphics drawing algorithms such as line draw and BitbIt operations tend to be clustered in small, 2-dimensional regions. For example, a line segment can be drawn at any angle, as shown in Figure 22. By mapping the rows of DRAM to large rectangles on the display, algorithms can take maximum advantage of page mode and thus draw images to the frame buffer very quickly. Figure 22 shows the 512 bits of a page of DRAM mapped to two adjacent 16 × 16 rectangular regions of the display. But there is a problem using tile maps which involves address mapping to the serial port of the Video RAM. The diagram shows that the CRT must scan from memory column address 15 to memory column address 256. What is needed is a way to jump from bit 15 to bit 256 in the SAM in "real time". Likewise, we will need to jump from 31 to 272 on the next line, and so on to jumping from 255 to 496 on the last line in the tile. In this example, we switch banks and jump to a new starting location in the SAM every 16 clock cycles. In general, the jump point can be any reasonable number 2^n-1, where n=4, 5, 6, 7. That is, the rectangle grids can be 16, 32, 64, or 128 bits wide, although the smaller the grid, the more transfer cycles will be required to support the display.

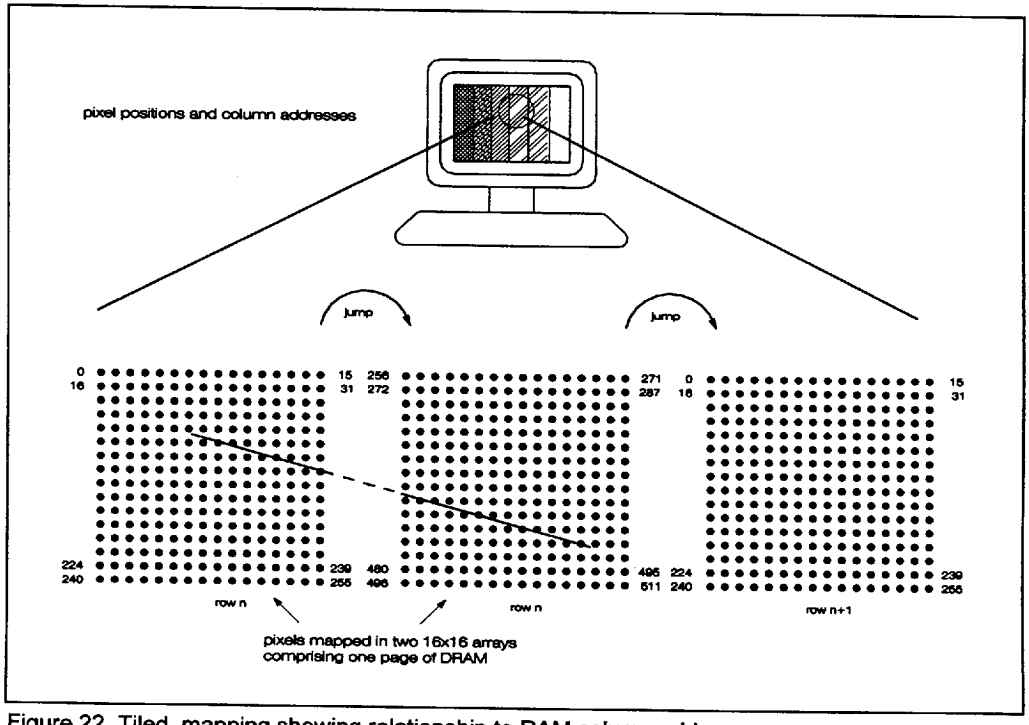


Figure 22. Tiled, mapping showing relationship to RAM column addresses

Illustration of stop bit mode with full SAM architecture

There are differences in how stop bit mode is implemented in the full SAM (SAM holds a full row of DRAM data) and half SAM architectures. Some minor changes to the half SAM architecture are required to make it compatible with the full SAM during stop bit mode. To understand this, it is instructive to introduce the stop bit implementation using the full SAM architecture.

Figure 23 illustrates how stop bit mode would be implemented in a 256k × 16 Video RAM using a full SAM. Like the half SAM, the full SAM is divided into two banks, but each bank holds twice as many columns. The lower bank consists of column addresses 0-255 and the upper bank consists of column addresses 256-511. In split register mode, the full SAM loads the lower (upper) 256 columns of the selected row into the SAM low (high) bank while the SAM high (low) bank delivers data to the serial output port. With reference to Figure 22, the SAM must, for example, be able to "jump" from address 15 to address 256 with no penalty in serial access time. Subsequently, the SAM must be able to jump from address 271 to address 0 of the next row, again with no penalty in serial access time. The sequence is as follows:

read out from low bank, row n	load high bank, row n, tap=256
read out bit 15, low bank, row n	
jump to bit 256, high bank, row n	
read out from high bank, row n	load low bank, row n+1, tap=0
read out bit 271, high bank, row n	
jump to bit 0, low bank, row n+1	
etc...	

In general, a jump must be made possible from address 2^n-1 in one bank to the tap location (selected during the reload) in the opposite bank. But this is just an extension of split register mode, except that the most significant address "compare for l's" is done only on a contiguous subset (the lower four bits in our example) of the counter instead of the eight bits required to support full SAM split register mode. The more significant bits of the counter are masked off (not used in the compare to activate the bank switch) according to what stop bit address is set. The stop bit registr is loaded when a $\overline{CAS}$ before $\overline{RAS}$ cycle with Stop mode select (CBRS) cycle is executed.

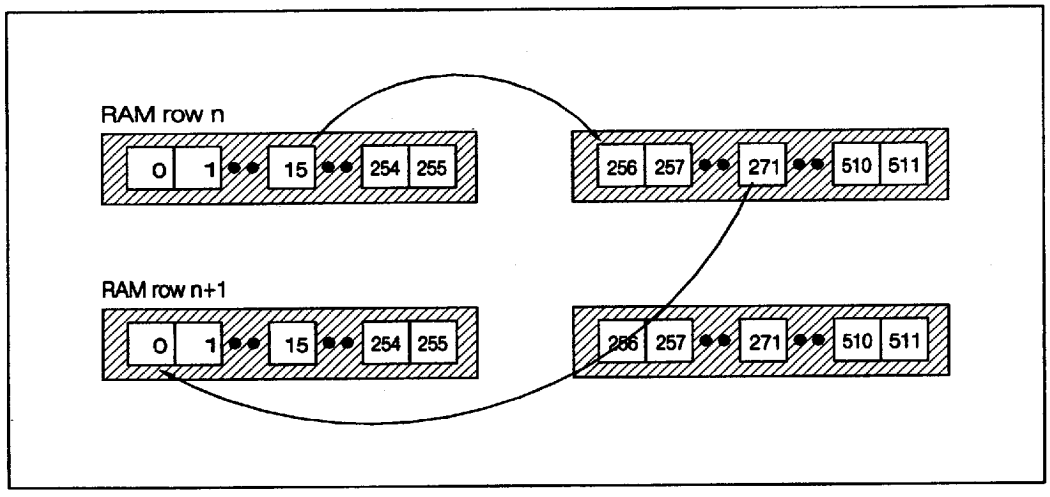


Figure 23. Internal mapping of binary stop bit operation, Full SAM.

Extension to the Half SAM Architecture

Figure 24a illustrates again the column address mapping of the DRAM to the SAM using a half SAM architecture. In contrast to the full SAM address mapping shown in Figure 23, it is impossible to access column address 15 and then immediately thereafter access column address 256 without reloading the lower bank of the SAM. This is because these addresses belong to different quad rows(0 and 2) which are mapped to the same (low) bank in the SAM. Likewise, all of the "jumps" implied in Figure 22 are impossible to implement with the half SAM architecture without changing the column address scheme.

What is needed in order to be compatible with full SAM Video RAMs operating in stop bit mode is to be able to jump between quad rows 0 and 2 and also to be able to jump between quad rows 1 and 3 with no penalty in serial access time. This means that when using stop bit mode the even numbered quad rows must be resident in separate SAM banks. Likewise, the odd numbered quad rows must be resident in separate SAM banks. To accomplish this, the 4Mb Video RAM performs an internal remapping of column address bits A8 and A7 in the address mapping of both the RAM and the SAM when a CBRS cycle is executed, as shown in Figure 24b.

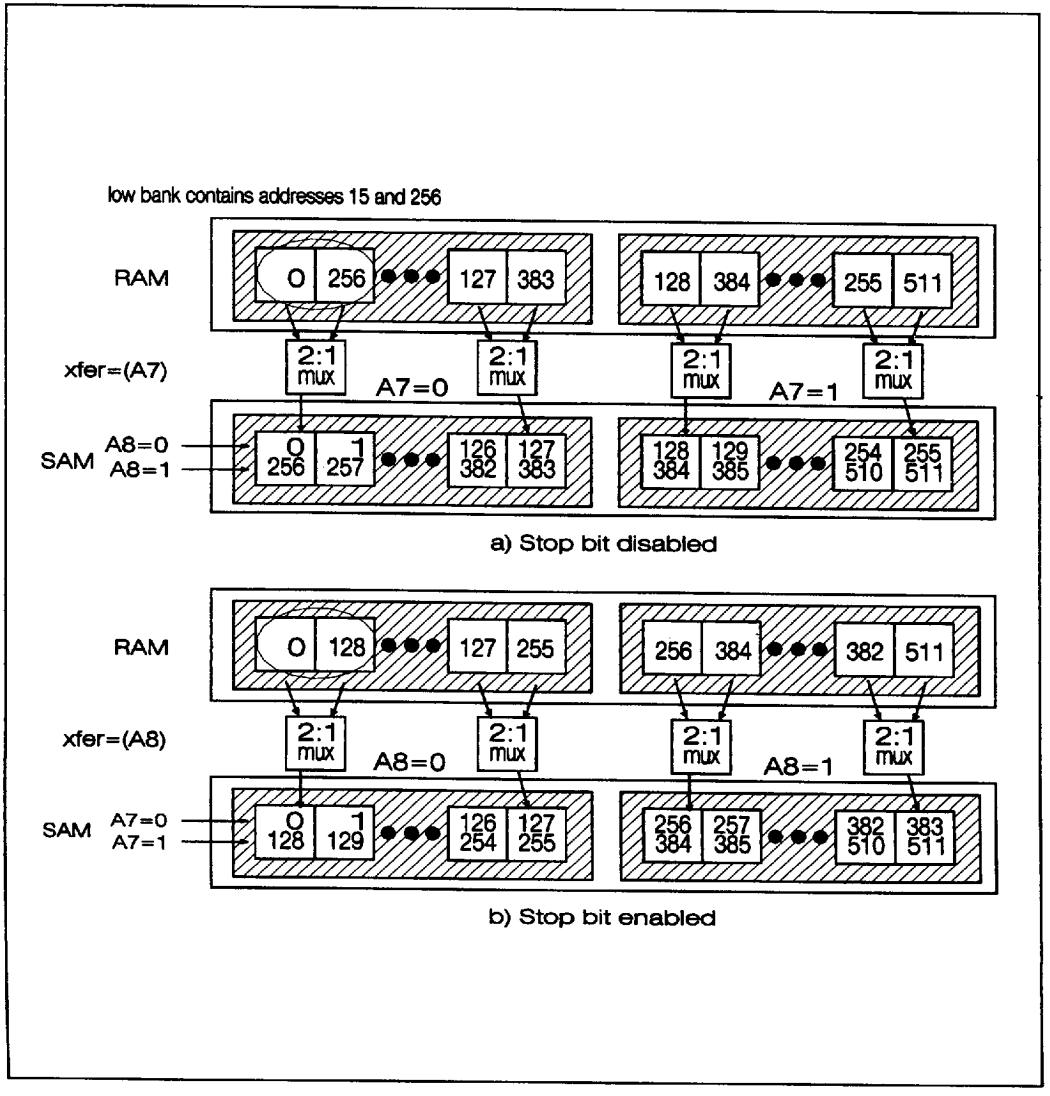


Figure 24. Half SAM column for stop mode enabled and disabled

Stop bit mode enable and stop bit select cycle

To select the stop bit boundary (the address at which the banks switch will occur), a $\overline{CAS}$ before $\overline{RAS}$ cycle is executed according to the CBRS cycle as shown in Table 2. The timing is shown in Figure 6. Table 3 describes how the setting of the stop bits during the CBRS cycle control the bank switching during SAM access. To reset stop bit mode, a similar $\overline{CAS}$ before $\overline{RAS}$ cycle, designated CBRR in Table 2 and also shown in Figure 6 is executed. According to the JEDEC standard for Video RAM, the CBRR cycle also resets persistent write per bit mode.

row address A7-A4 set during last CBRS cycle [6]	stop bit register S6-S0	counter bits compared for "1"	stop boundaries(bank switch enabled) column address.
0111	111 1111	C6 - C0	127,255,383,511(default)
0011	011 1111	C5 - C0	63,127,191,255,319,383,447,511
0001	001 1111	C4 - C0	31,63,95,127,159,191,223,255,287, 319,351,383,415,447,479,511
0000	000 1111	C3 - C0	15,31,47,63,95,111,127,143,159, 175,191,207,223,239,255,271, 287,303,319,335,351,367,383, 399,415,431,447,463,479,495,511

Table 3. Control of bank switching using stop bits in split register mode.

NOTES:

[6] A7 is not used. However to maintain compatibility with 512 word (full) SAM Video RAM devices, it is recommended that A7 be supplied and set to a logic low level during all CBRS cycles.

Figure 25 shows a simplified logic diagram of the binary stop bit mode. Signals CBRR(RAS) and CBRS(RAS) are signals generated during CBRR(normal CBR) and CBRS(CBR with stop bit mode enable and register load) cycles, respectively and are functions of the input signals as specified in Table 2. The stop bit register is loaded from the row address buffers during CBRS cycles.

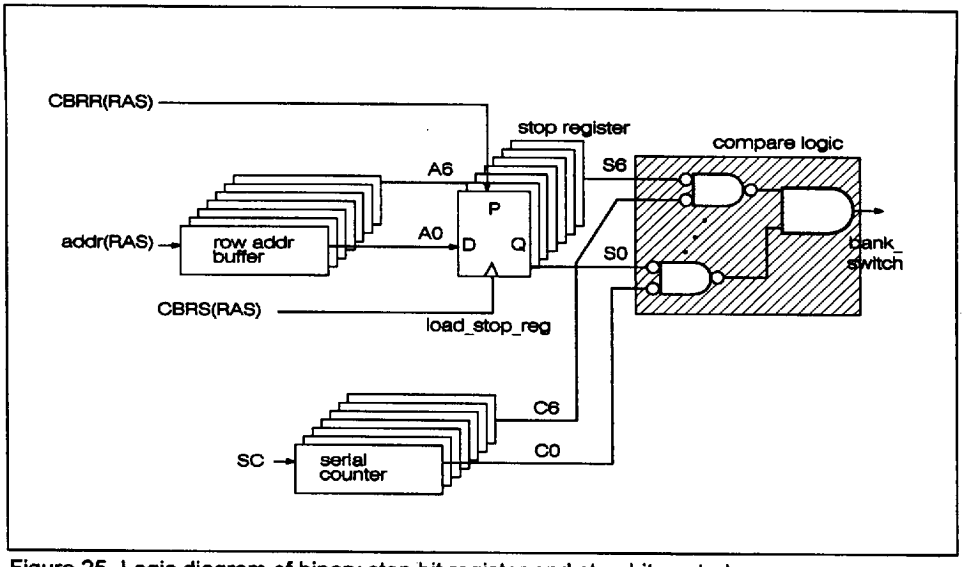


Figure 25. Logic diagram of binary stop bit register and stop bit control

Serial access

Normal Mode (split register mode disabled)

While split register mode is disabled, the SAM begins reading after a memory to register transfer cycle from the tap location and reads out the serial data in ascending order of serial address. When the most significant SAM address has been accessed, the SAM will "wrap around" to the least signficant SAM address.

Split Register Mode Enabled, boundary stop bit mode disabled

In split register mode, but with boundary stop bit mode disabled, one bank of the SAM is loaded during every memory to register transfer cycle. After the active bank accesses its most significant address, a bank switch will occur and the SAM will next access the tap location in the newly active bank.

IMPORTANT: This assumes that a transfer to the inactive bank always occurs before the active bank reaches its most significant address. In the case where a transfer to the inactive bank does not occur before the active bank reaches its most significant address, split register operation will be terminated. Subsequently, the next bit read out after the most significant bank address (msba) will be msba+1, module 256.

Split Register Mode Enabled, boundary stop bit mode enabled

In split register mode, with boundary stop bit mode enabled, one bank of the SAM is loaded during every memory to register transfer cycle. After the active bank accesses the next specified boundary according to the boundary stops set in the most recent CBRS cycle, a bank switch will occur and the SAM will next access the tap location in the newly active bank.

IMPORTANT: This sequence of events assumes that a transfer to the inactive bank always occurs before the active bank reaches the next boundary address. In the case where a transfer to the inactive bank does not occur before the active bank reaches the next boundary address, split register operation will be terminated. Subsequently, the next bit read out after the boundary will be boundary + 1, module 256.

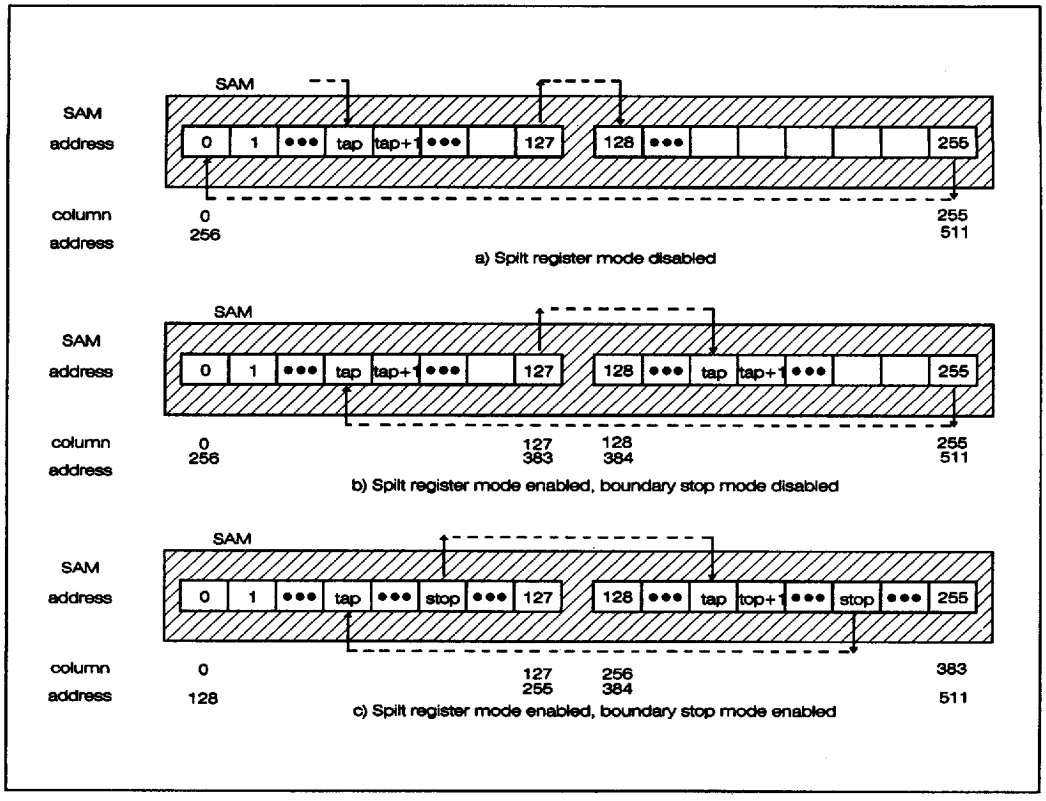


Figure 26. Serial address sequence by operating mode, half SAM

Figure 26 summarizes the address sequence of serial operation for the three basic serial/transfer operating modes. Note the internal column address swap of A7 and A8 in the case of binary stop bit mode enabled.

Power Up Operation

Sequence

1. $\overline{RAS}$ and $\overline{TRG}$ should be pulled up to V_{dd} as V_{dd} is turned on and remain so for a minimum of 200 μs after V_{dd} has stabilized. Hyundai recommends that in applications where the SQ pins are tied to other device drivers which may be active during power up the $\overline{SE}$ also be pulled up to V_{dd} as V_{dd} is turned on.

2. A minimum of any eight RAS cycles(eg. CBR, RAS only refresh, reads, writes, transfers) be performed.

Initial state

state, variable, etc.	state
persistent write per bit mode	disabled
binary stop bit mode	disabled
split register mode	unknown
SQi	hiZ[7]
QSF	hiZ[7]

Table 4. power up initial state assignments

Notes:

[7]. During power up, SQi and QSF will remain in the hi Z state regardless of the state of SE. After power up, SQi and QSF will remain in the hi Z state until a normal read transfer (RT) cycle is performed, after which the SE control input will resume hi Z control of SQi and QSF.

Timing diagrams

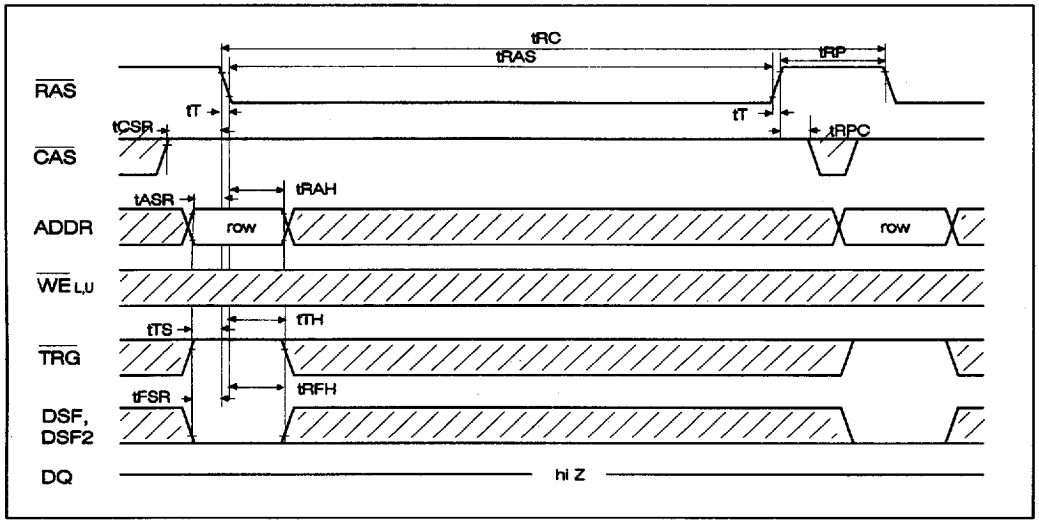


Figure 27. RAS only Refresh Cycle

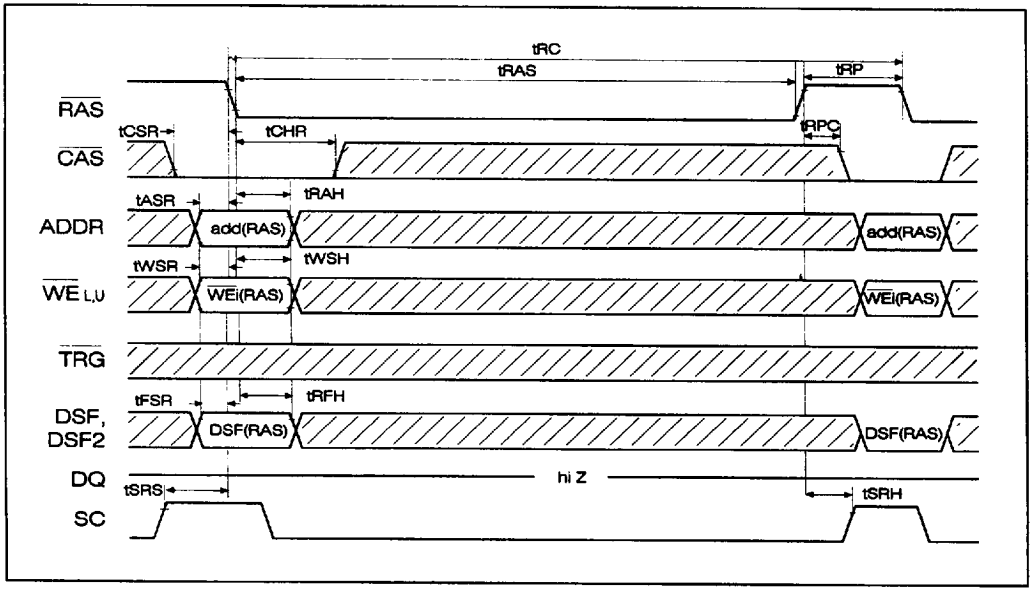


Figure 28. CBR auto refresh and option set/clear(CBRR, CBRN. CBRS)

Funciton ,or Cycle Type	RAS↓ signal(RAS)					mne code
	CAS	TRG	WEL and WEU[15]	DSF	Address(add)	
CBR, reset to nonpersistent & disable stop bit mode	0	X	1	0	X	CBRR
CBR, Enable,set Stop bit	0	X	0	1	stop(A4-A7)[16]	CBRS
CBR, No resetnonpersist ent or disable of stop bit	0	X	1	1	X	CBRN

[15] WEL and WEU must both be high on the falling edge of RAS to invoke a logic 1.
[16] A7 is ignored but should be supplied and set to a logic low level to guarantee compatibility with 512 SAM VRAMs.

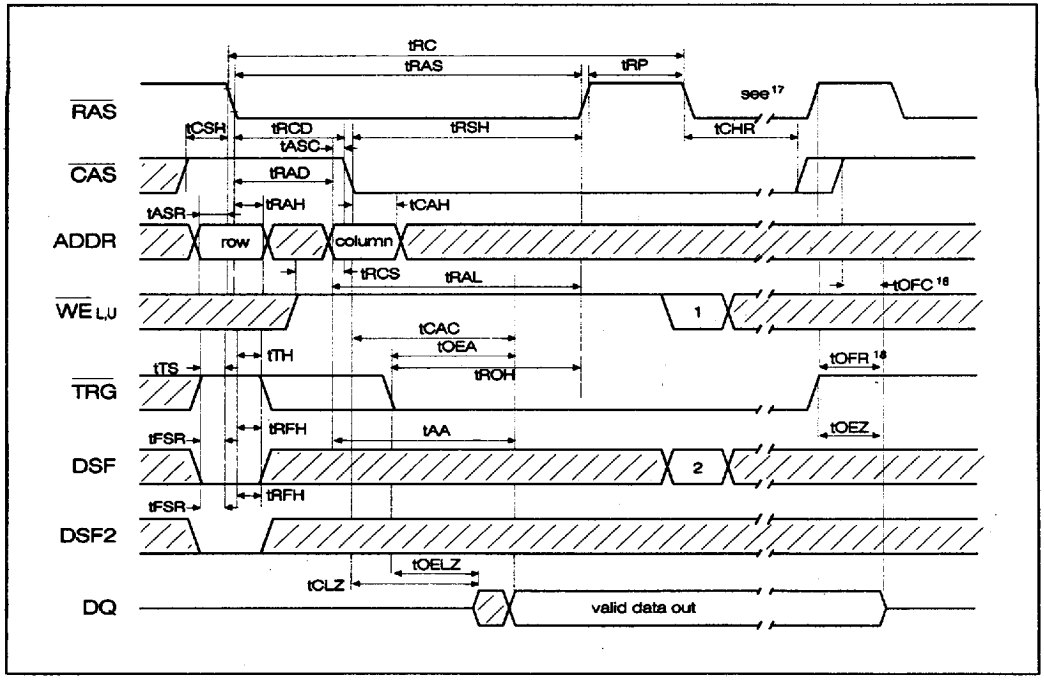


Figure 29. Read cycle with hidden refresh

Notes :

17. Multiple hidden refresh cycles can occur.
18. The later of $\overline{RAS}\uparrow$ and $\overline{CAS}\uparrow$ will cause the output to tri state.

STATE		Refresh Cycle Performed
1	2	
L	H	CBRS (stop point set)
H	L	CBRR (reset stop mode and persistent mode)
H	H	CBRN (no option reset)

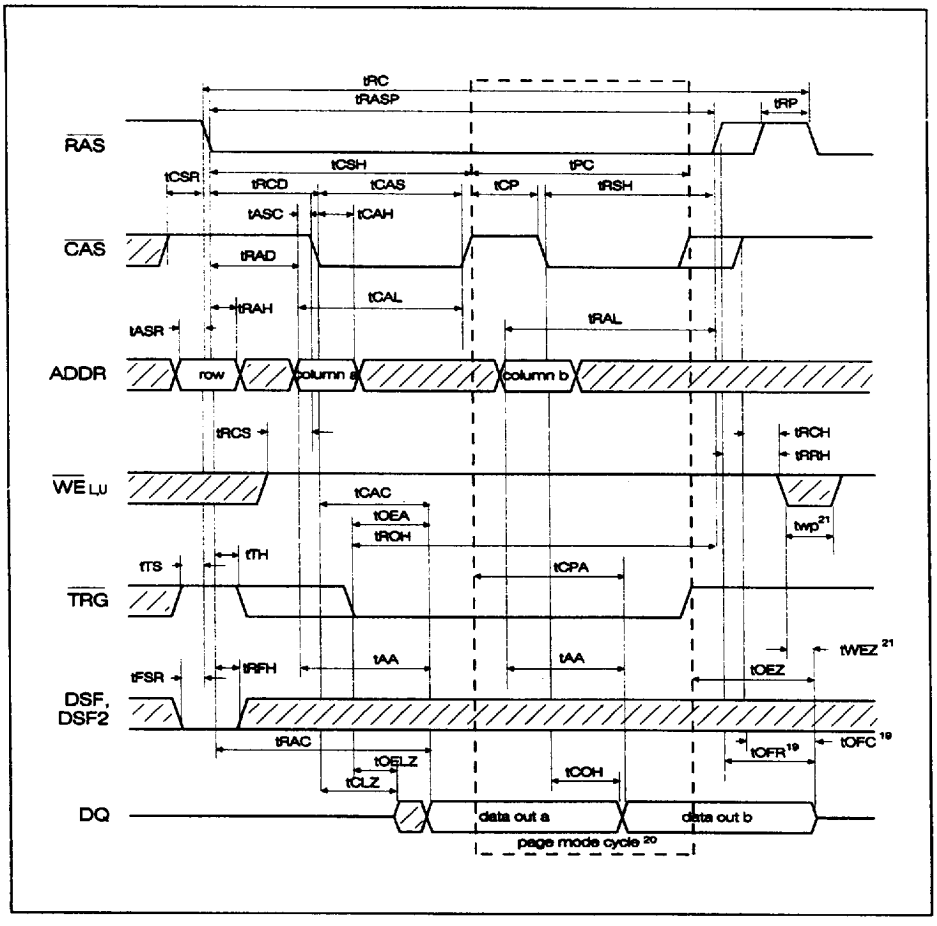


Figure 30. Read operation with extended output page mode

Notes :

19. The later of $\overline{RAS}\uparrow$ and $\overline{CAS}\uparrow$ will cause the output to tri state.

20. Page mode cycles can be repeated

21. $\overline{WEL}\downarrow$ or $\overline{WEU}\downarrow$, will cause outputs to tri-state until $\overline{WEL}$, $\overline{WEU}$ are returned high and $\overline{RAS}$, $\overline{CAS}$, and $\overline{TRG}$ are again brought low. twp must be met to assure that outputs tri state after $\overline{WEL}\downarrow$ or $\overline{WEU}\downarrow$

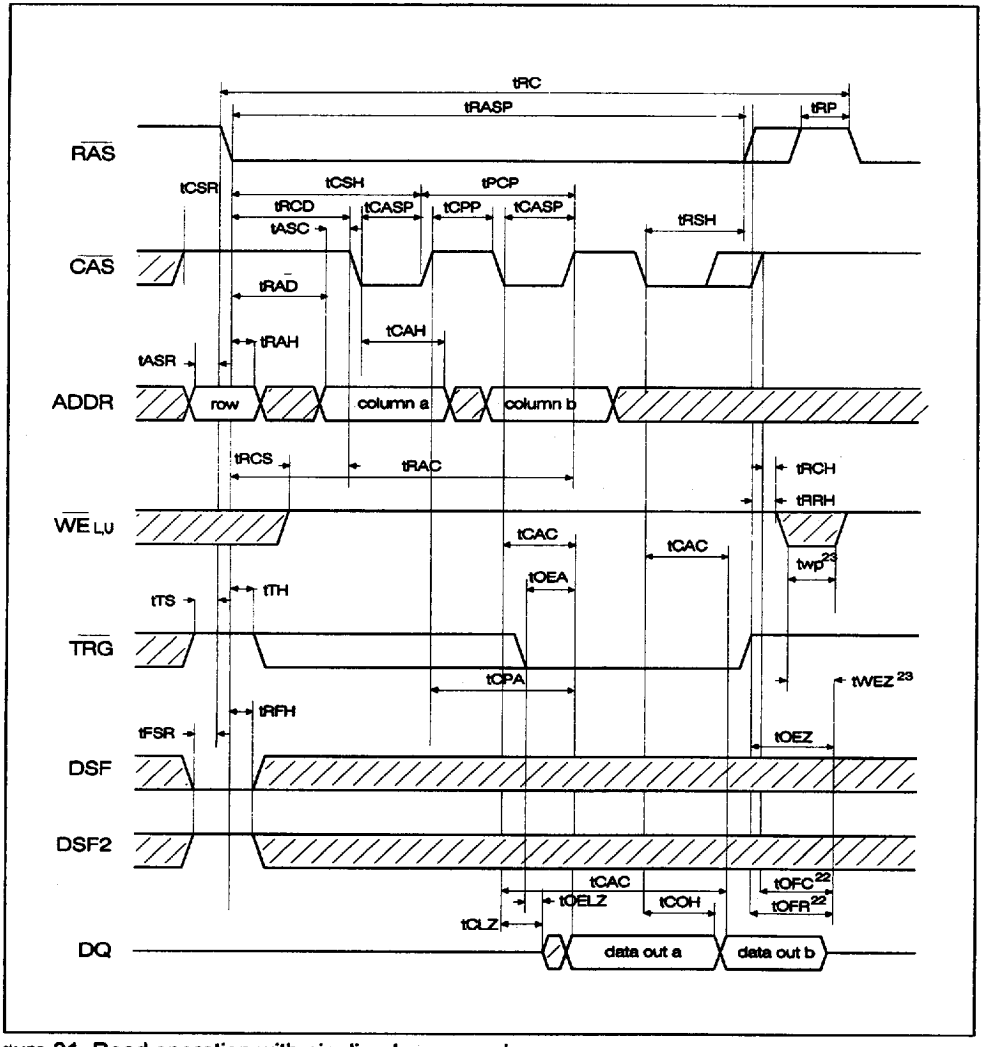


Figure 31. Read operation with pipelined page mode

Notes :

22. The later of $\overline{RAS}\uparrow$ and $\overline{CAS}\uparrow$ will cause the output to tri state.

23. $\overline{WEL}$ or $\overline{WEU}$, will cause the outputs to tri-state until $\overline{WEL}$, $\overline{WEU}$ are returned high and $\overline{RAS}$, $\overline{CAS}$, and $\overline{TRG}$ are again brought low. tWP must be met to assure that outputs tristate after $\overline{WEL}\downarrow$ of $\overline{WEU}\downarrow$

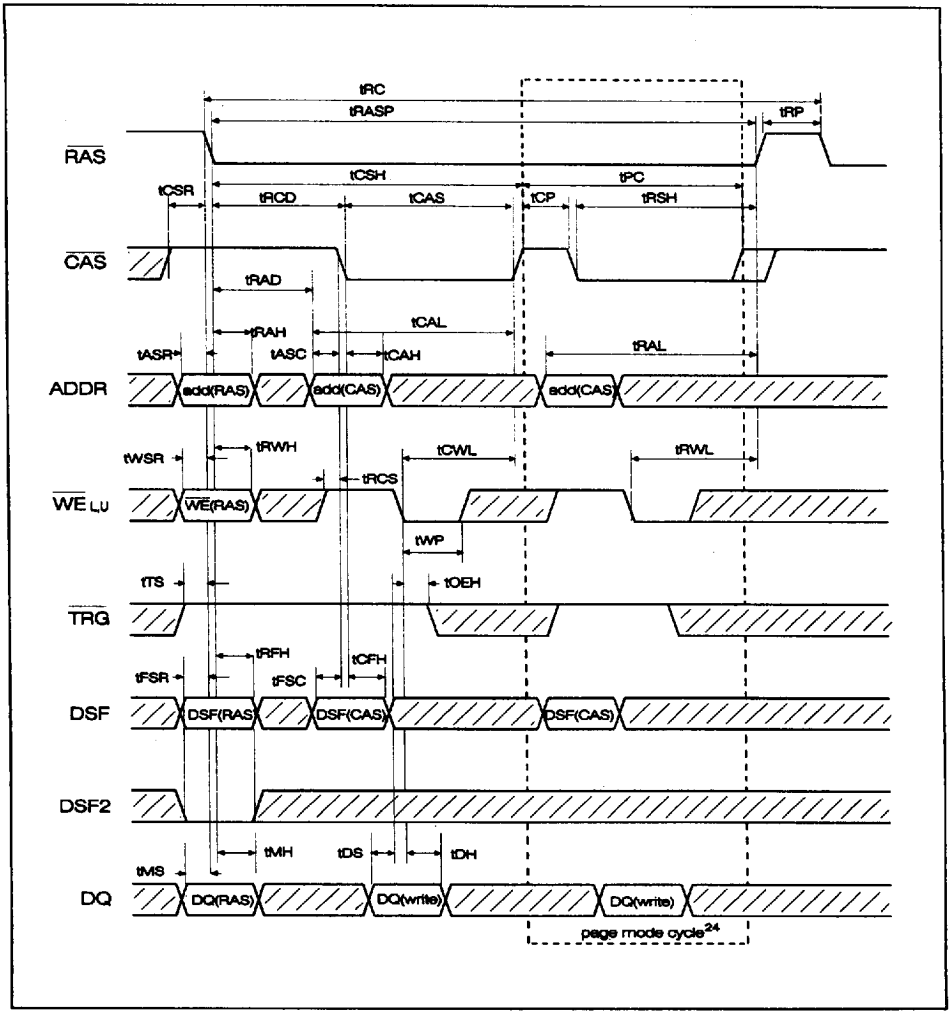


Figure 32. Late write, write mask load, color register load with page mode.

(See Table 5 for logic assignments.)

Function, or Cycle Type	$\overline{RAS}\downarrow$ signal(RAS)				$\overline{CAS}\downarrow$ signal(CAS)		$\overline{CAS}\downarrow$ & ($\overline{WEL}\downarrow$ + $\overline{WEU}\downarrow$)[25]	mne code
	$\overline{WEL}$& $\overline{WEU}$[26]	DSF	Address	DQi[27]	DSF	Address	DQi	
DRAM write per bit, nonpersistent, use new mask	0	0	row	write mask	0	column	data	RWM
DRAM block write with write per bit, nonpersistent, use new mask	0	0	row	write mask	1	block address A3-A8	column. address mask	BWM
DRAM write per bit, persistent mode, use old mask	0	0	row	X	0	column	data	RWM
DRAM block write with write per bit, persistent mode, use old mask	0	0	row	X	1	block address A3-A8	column address mask	BWM
DRAM write, no write per bit, no mask	1	0	row	X	0	column	data	RW
DRAM block write, no write per bit, no mask	1	0	row	X	1	block address A3-A8	column address mask	BW
Load Write Mask Register	1	1	X	X	0	X	write mask	LMR
Load Color Register	1	1	X	X	1	X	color data	LCR

Table 5. Truth table for write and register load operations.

Notes:

24. Page mode cycles can be repeated. Page mode is not available for Load Write Mask Register(LMR) or Load Load Color Register(LCR) operations.

25. The signals on all DQ pins are strobed in on the later of the falling edge of $\overline{CAS}$ and the falling edge of either $\overline{WEL}$ and $\overline{WEU}$.
However, the low(high) byte is not enabled for write operation unless $\overline{WEL}$ ($\overline{WEU}$) is asserted.

26. $\overline{WEL}$ and $\overline{WEU}$ must both be high on the falling edge of RAS to invoke a logic 1.

27. DQi refers to DQ0 - DQ15 collectively.

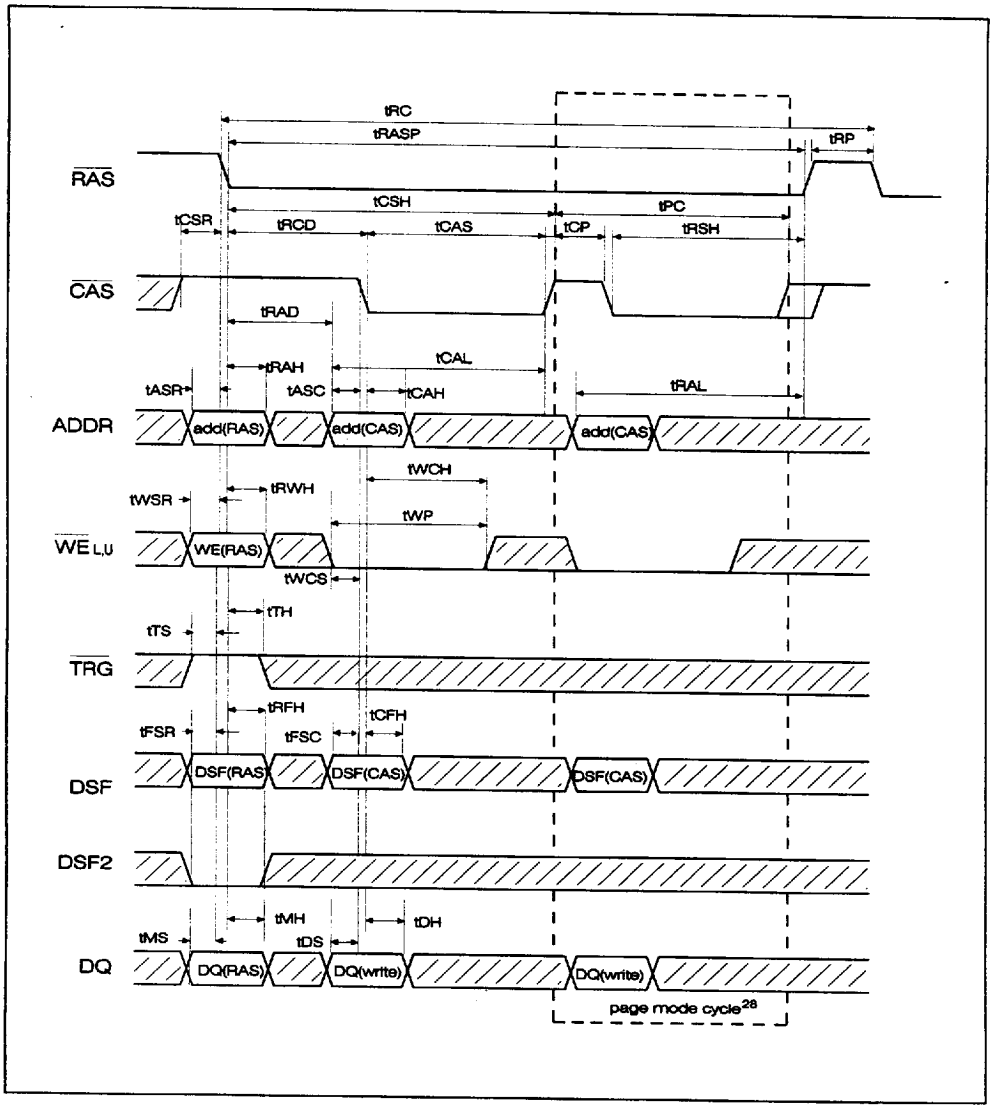


Figure 33. Early Write and Write Mask, color register load with page mode.
(See Table 5 for logic assignments.)

Notes :

[28] Page mode cycles can be repeated. Page mode is not available for Load Write Mask Register(LMR) or Load Color Register(LCR) operations.

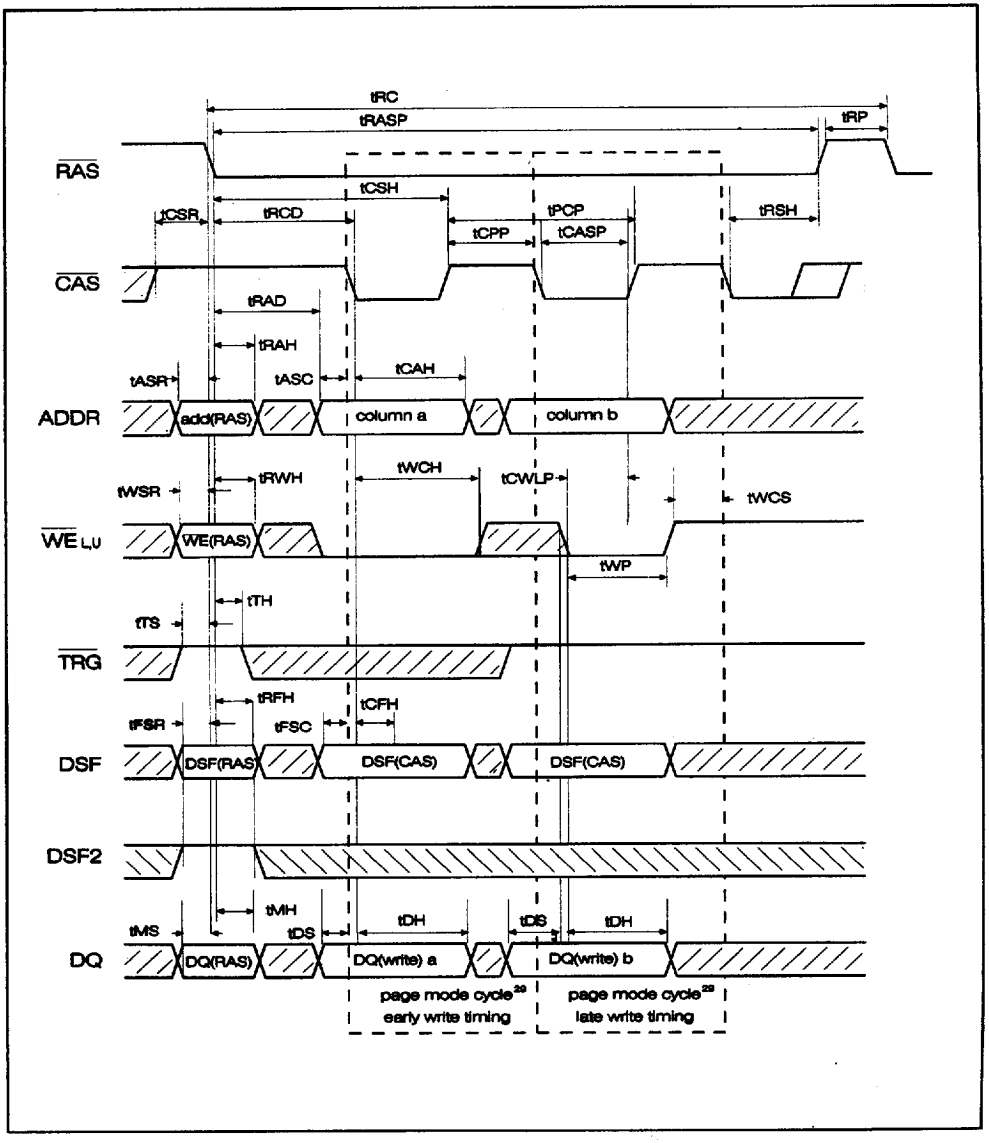


Figure 34. Pipelined page mode write cycles.
(See Table 5 for logic assignments.)

[29] Pipelined page mode is not available for Load Write Mask Register (LMR) or Load Color Register (LCR) operations.

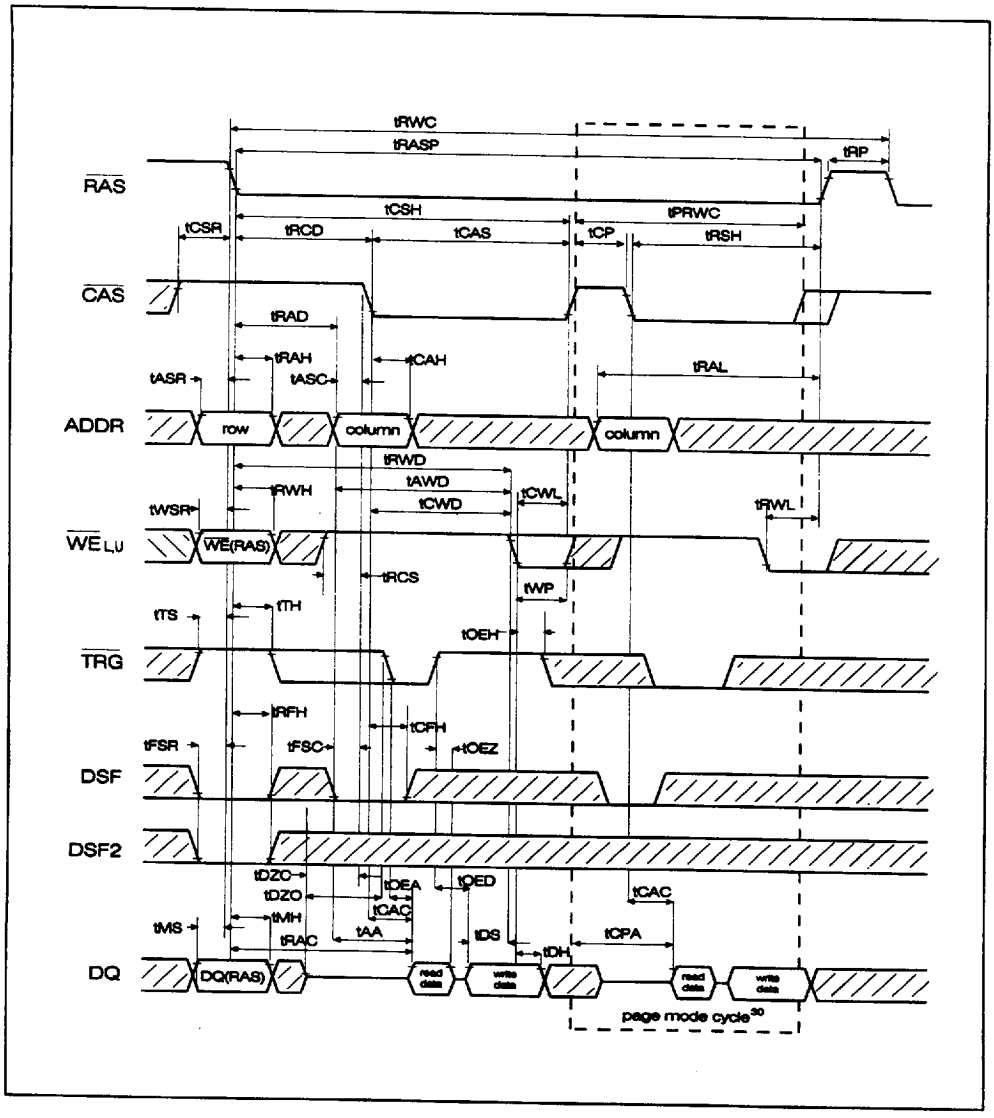


Figure 35. Read - Modify- Write with page mode.
(See Table 5 for logic assignments.)

30. Page mode cycles can be repeated. Page mode is not available for Load Write Mask Register(LMR) or Load Color Register(LCR) operations.

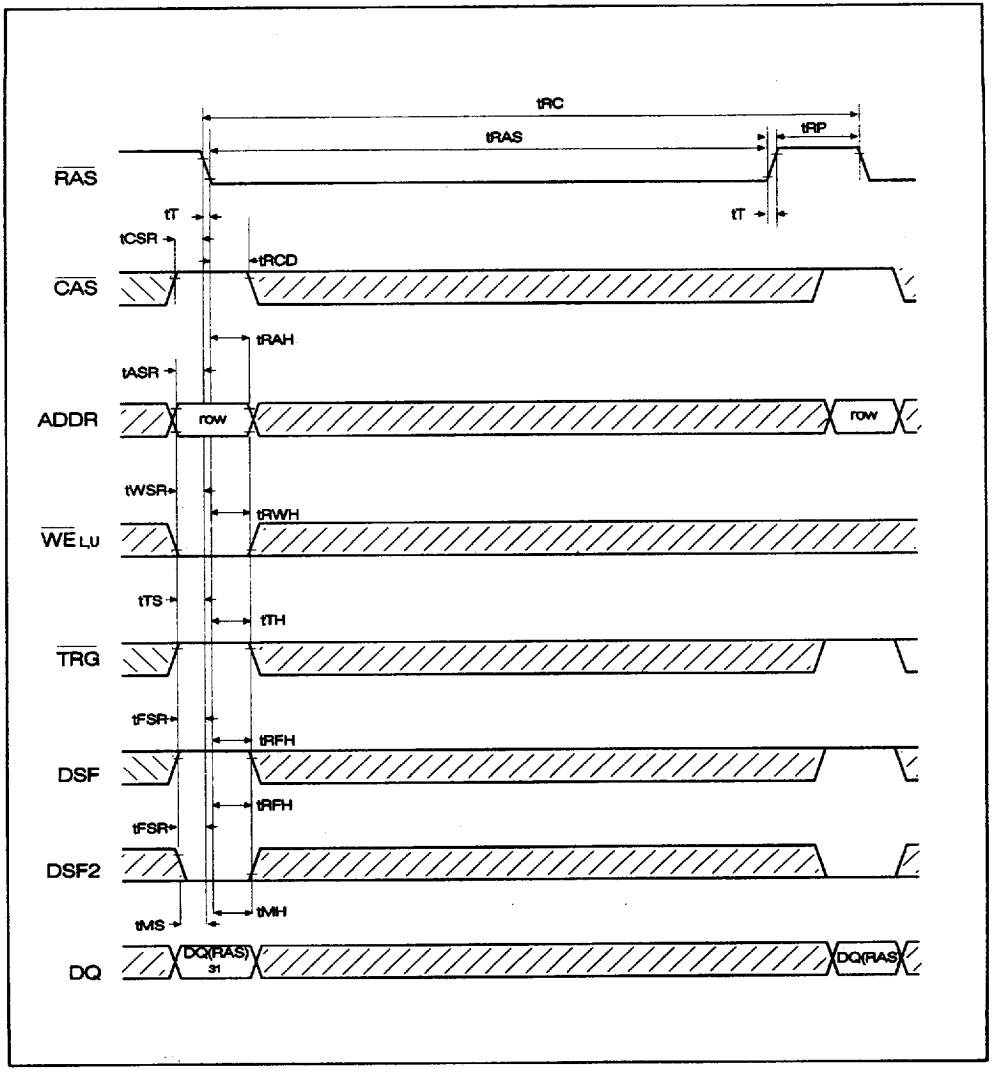


Figure 36. Masked Flash Write Cycles

Notes :

31. DQ(RAS)is required when operating in nonpersistent mode and is a "don't care" when operating in persistent mode.

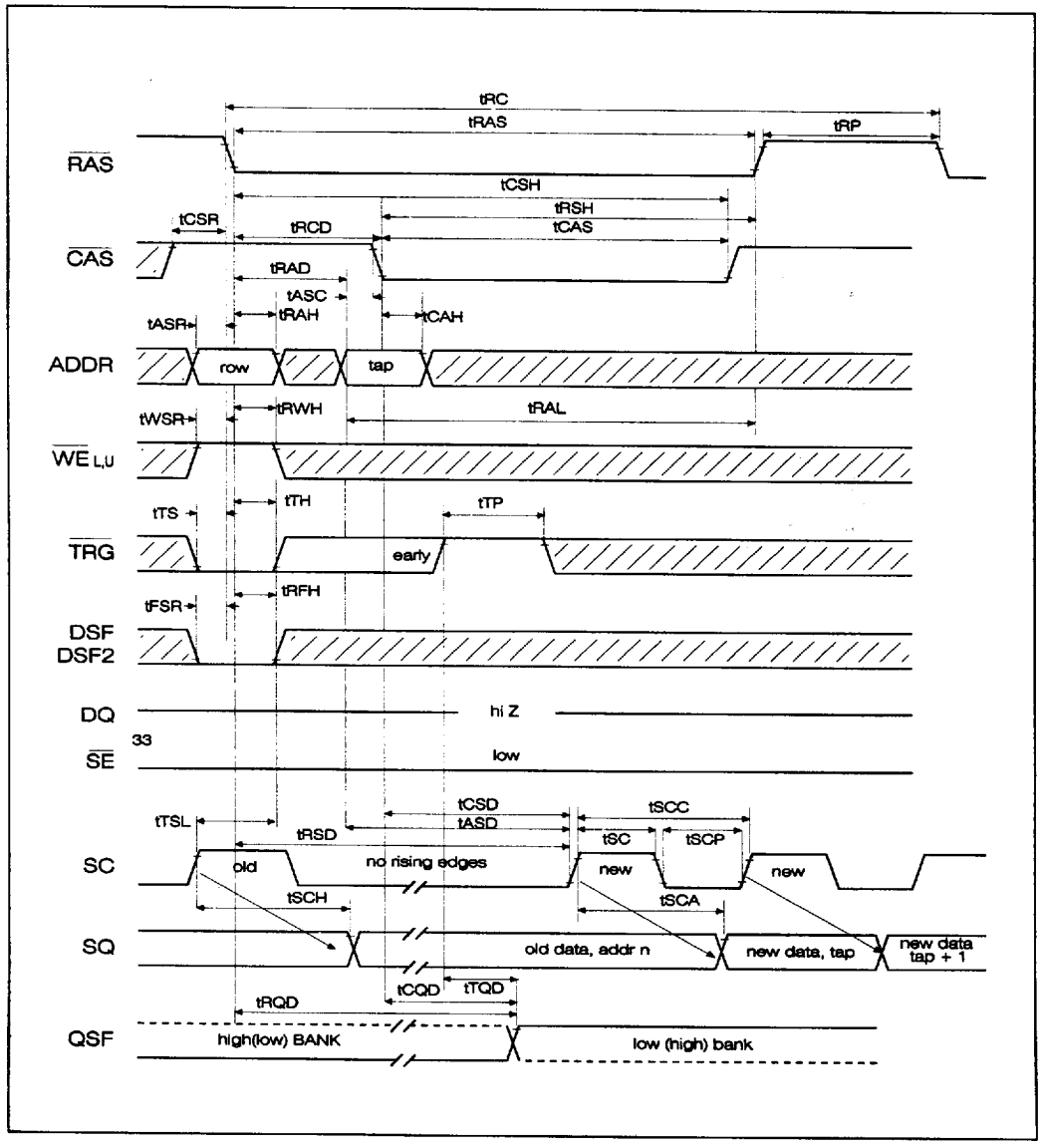


Figure 37. Memory to Register Transfer, Early Load[32].

[32] Early Load is defined where $\overline{RAS}\downarrow$ to $\overline{TRG}\uparrow$ is less than t_{RTH}, the minimum delay time required to perform a real time transfer. The Early Load cycle is designed to allow $\overline{TRG}$ to go high any time after t_{TH} from $\overline{RAS}\downarrow$. The syncronization of the memory to register transfer is handled internally. Since this internal synchronization is done independently of the serial clock. SC, Early Load is not intended for applications requiring real time memory to register data transfer. An example of an application not requiring real time memory to register transfer is a display system in which all memory to register transfer cycles are done during the blanking intervals of the display.

[33] $\overline{SE}\_$ acts only as a serial output enable control and does NOT inhibit transfer operations from occuring when negated.

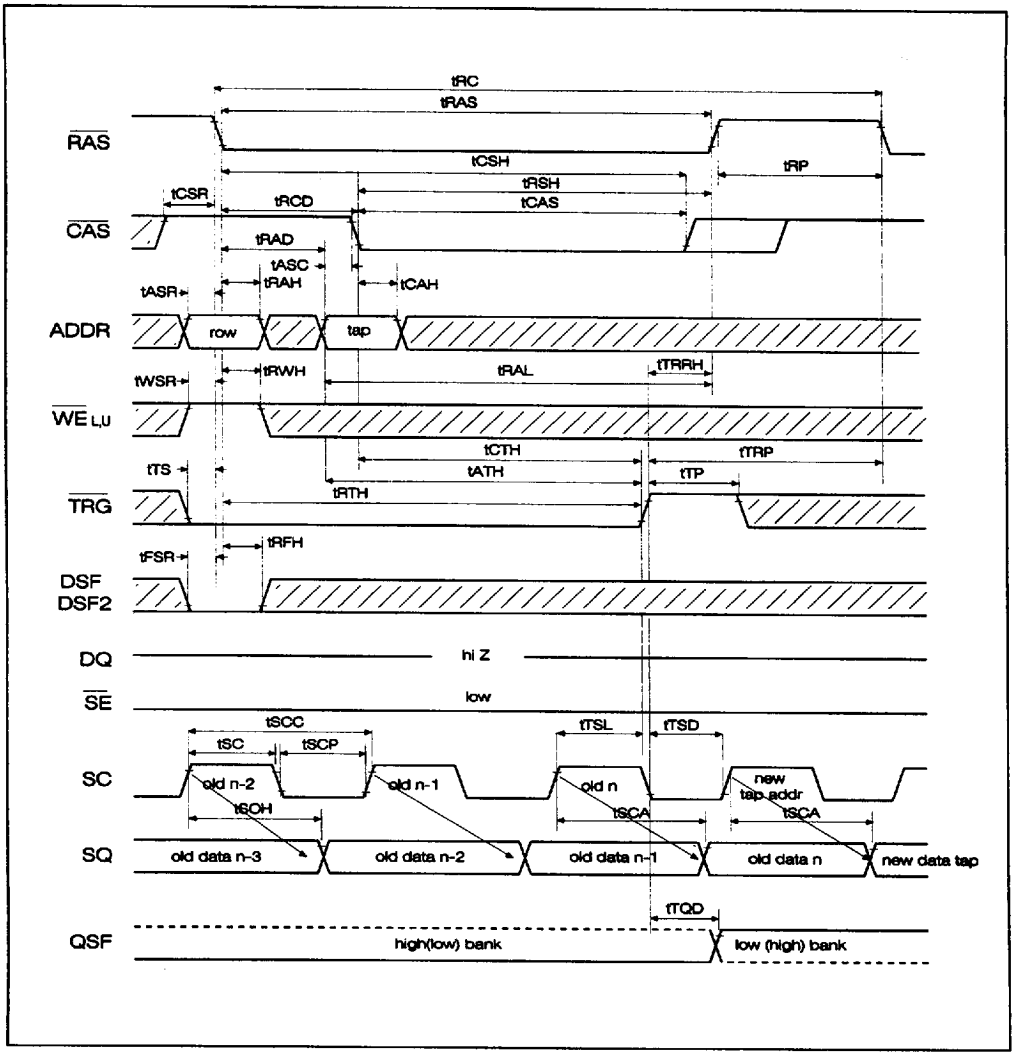


Figure 38. Real Time and Real Time late load Memory to Register Transfer

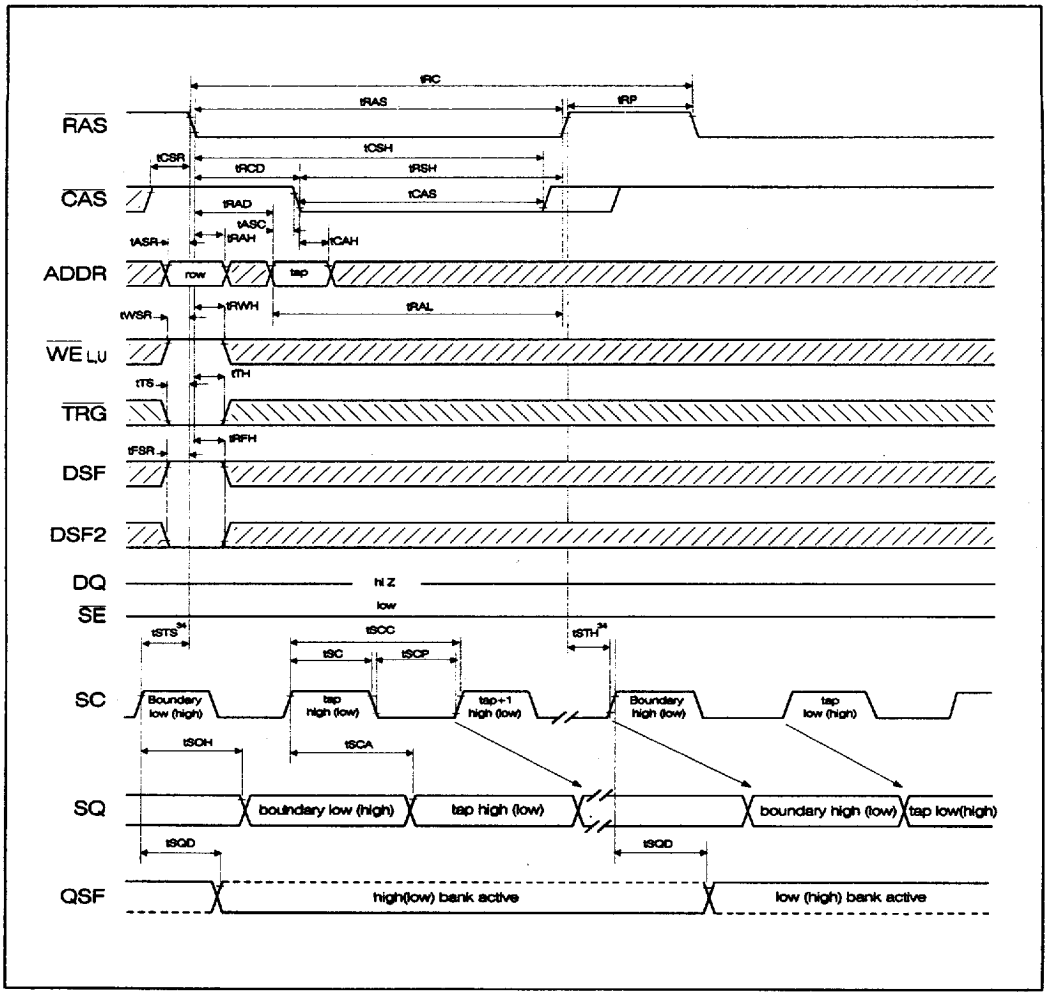


Figure 39. Memory to Register Transfer, Split Register

[34] tSTS and tSTH parameters pertain only to SC cycles which correspond to boundary addresses (ie, QSF changes).

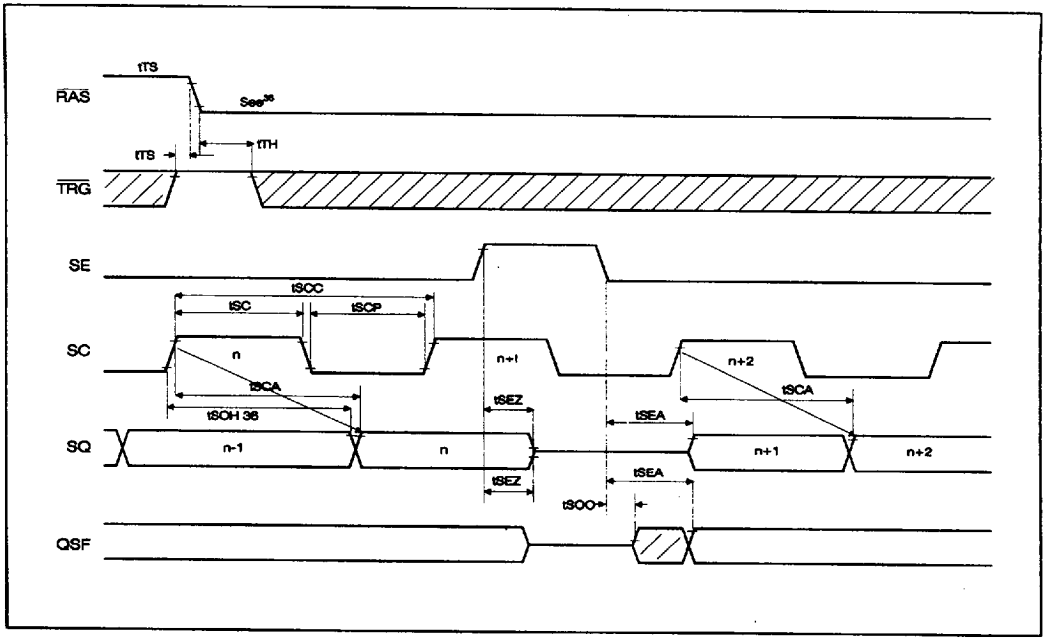


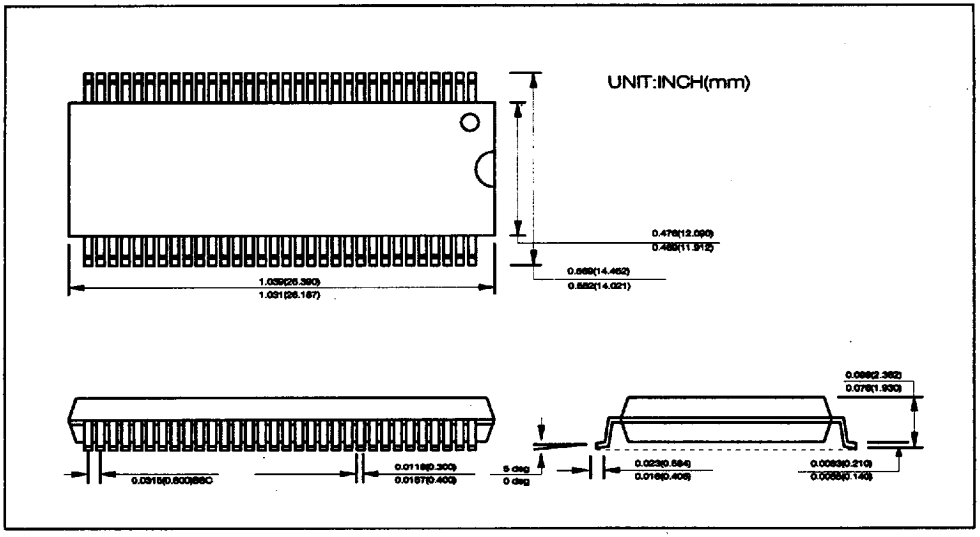
Figure 40. Serial Read

35. The serial read cycle timings assume there are no CBRR or CBRS refresh cycles executed during the serial access. Executing CBRR or CBRS refresh cycles during serial read access could give unexpected results.

36. tSOH is referenced from the point where SC leaves a valid logic low level.

PACKAGE INFORMATION

525mil 64pin Super Small Outline Gullwing leaded Package(GE)



ORDERING INFORMATION

PART NO	SPEED	PACKAGE
HY5216256GE	60/70	2/CAS, EDO 525 mil SSOP
HY5216257GE	60/70	2/WE, EDO 525 mil SSOP