4 K × 8 CMOS Dual Port RAM with Semaphore

Introduction

The M671342 is a very low power CMOS dual port static RAM organised as 4096 × 8, with full hardware support of semaphore signaling between the two ports.

The M671342 device provide two independant ports with separate control, address and I/O pins that permit independant, asynchronous access for reads and writes to any location in the memory. Only the simultaneous write by the two ports to the same RAM location is not allowed. The semaphores are available on chip to assist in arbitrating between ports. An automatic power down feature controlled by $\overline{CS}$ permits the on-chip circuitry of each port in order to enter a very low stand by power mode.

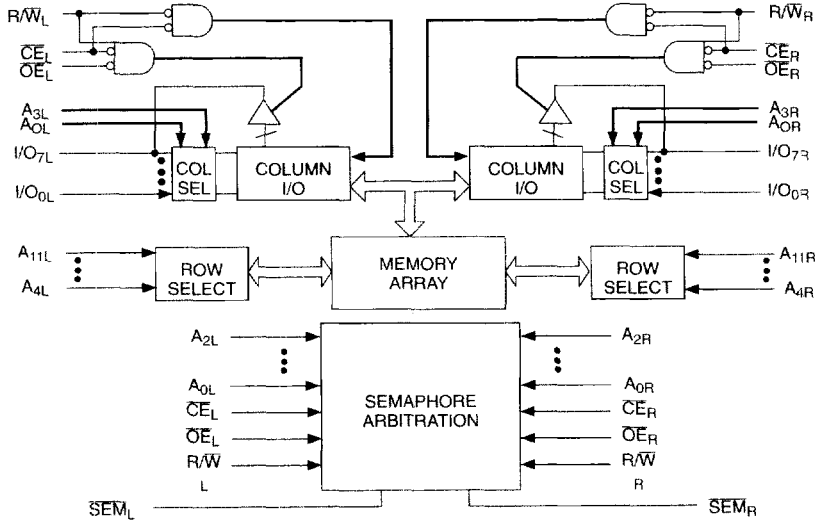
Using an array of eigh transistors (8T) memory cell and fabricated with the state of the art 0.6 µm lithography named SCMOS, the M 671342 combines an extremely low standby supply current (typ = 1.0 µA) with a fast access time at 18 ns over the full temperature range. All versions offer battery backup data retention capability with a typical power consumption at less than 1 µW.

For military/space applications that demand superior levels of performance and reliability the M671342 is processed according to the methods of the latest revision of the MIL STD 883 (class B or S) and/or ESA SCC 9000.

Features

- Fast Access Time : 18/20/25/30/35/45 ns
- Wide Temperature Range : –55°C to +125°C
- 671342 L Low Power
  671342 V Very Low Power
- Full On-Chip Hardware Support of Semaphore Signaling Between Ports
- Fully Asynchronous Operation From Either Port
- Battery Back-up Operation : 2 V Data Retention
- TTL Compatible
- Single 5 V ± 10 % Power Supply
- 3.3 V Version Also Available, Please Consult Sales
- Available in 52-pin Popular Hermetic and Plastic Packages

Figure 1. Block Diagram

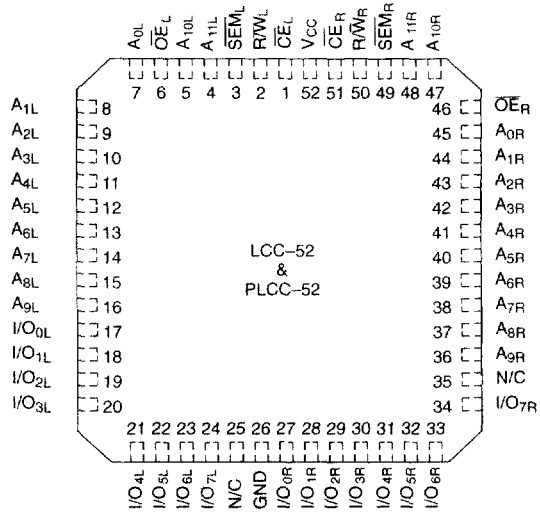


Pin Names

Left Port	Right Port	Names
$\overline{CS}_L$	$\overline{CS}_R$	Chip select
$R/\overline{W}_L$	$R/\overline{W}_R$	Read/Write Enable
$\overline{OE}_L$	$\overline{OE}_R$	Output Enable
A_{0L-11L}	A_{0R-11R}	Address
I/O_{0L-7L}	I/O_{0R-7R}	Data Input/Output
$\overline{SEM}_L$	$\overline{SEM}_R$	Semaphore Enable
Vcc		Power
GND		Ground

Functional Description

Pin Configuration



Truth Table

Table 1 : Non Contention Read/Write Control (1)

Inputs				Outputs	Mode
R/W	CE	SEM	OE	D_{0-7}	
X	H	H	X	Z	Port Disabled and in Power Down Mode
H	H	L	L	$DATA_{OUT}$	Data in Semaphore Flag Output on Port
X	X	X	H	Z	Output Disabled

Note :

1. $A_{0L} - A_{11L} \neq A_{0R} - A_{11R}$
   H = HIGH, L = LOW, X = Don't Care, Z = High Impedance
   ↗ = Low-to-High transition.

Functional Description

The M 671342 has two ports with separate control, address and I/0 pins that permit independent read/write access to any memory location. Both ports are identical in function to standard CMOS static RAMs and can be read from or written to at the same time, with the only possible conflict arising from the simultaneous writing of, a non-semaphore location. The semaphores, available on the M 671342, are protected against such ambiguous situations and may be used by the system program to avoid any conflicts in the memory of the dual-port RAM. These devices have an automatic power-down feature controlled by $\overline{CS}$.$\overline{CS}$ controls on-chip power-down circuitry which causes the port concerned to go into stand-by mode when not selected ($\overline{CS}$ high). When a port is selected access to the full memory array is permitted. Each port has its own Output Enable control ($\overline{OE}$). In read mode, the port's $\overline{OE}$ turns the Output drivers on when set LOW. Non-conflicting READ/WRITE conditions are illustrated in table 1.

Semaphore Logic

Functional Description

The M 671342 is an extremely fast dual-port 4k × 8 CMOS static RAM with an additional locations dedicated to binary semaphore flags. These flags allow either of the processors on the left or right side of the dual-port RAM to claim priority over the other for functions defined by the system software. For example, the semaphore flag can be used by oner processor to inhibit the other from accessing a portion of the dual-port RAM or any other shared resource.

The dual-port RAM has a fast access time, and the two ports are completely independent of each another. This means that the activity on the left port cannot slow the access time of the right port. The ports are identical in function to standard CMOS static RAMs and can be read from, or written to, at the same time with the only possible conflict arising from simultaneous writing to, or a simultaneous READ/WRITE operation on, a non-semaphore location. Semaphores are protected against such ambiguous situations and may be used by the system program to prevent conflicts in the non-semaphore segment of the dual-port RAM. The devices have an automatic power-down feature controlled by $\overline{CS}$, the dual-port RAM select and $\overline{SEM}$, the semaphore enable. The $\overline{CS}$ and $\overline{SEM}$ pins control on-chip-power-down circuitry that permits the port concerned to go into stand-by mode when not selected. This conditions is shown in table 1 where $\overline{CS}$ and $\overline{SEM}$ are both high.

Systems best able to exploit the M 671342 are based around multiple processors or controllers and are typically very high-speed, software controlled or software-intensive systems. These systems can benefit from the performance enhancement offered by the M 671342 hardware semaphores, which provide a lock-out mechanism without the need for complex programming.

Software handshaking between processors offers the maximum level of system flexibility by permitting shared resources to be allocated in varying configurations. The M 671342 does not use its semaphore flags to control any resources through hardware, thus allowing the system designer total flexibility in system architecture.

An advantage of using semaphores rather than the more usual methods of hardware arbitration is that neither processor ever incurs wait states. This can prove to be a considerable advantage in very high speed systems.

How the Semaphore Flags Work

The semaphore logic is a set of eight latches independent of the dual-port RAM. These latches can be used to pass a flag or token, from one port to the other to indicate that a shared resource is in use. The semaphore provide the hardware context for the "Token Passing Allocation" method of use assignment. This method uses the state of a semaphore latch as a token indicating that a shared resource is in use. If the left processor needs to use a resource, it requests the token by setting the latch. The processor then verifies that the latch has been set by reading it. If the latch has been set the processor assumes control over the shared resource. If the latch has not been set, the left processor has established that the right processor had set the latch first, has the token and is using the shared resource. The left processor may then either repeatedly query the status of the semaphore, or abandon its request for the token and perform another operation whilst occasionally attempting to gain control of the token through a set and test operation. Once the right side has relinquished the token the left side will be able to take control of the shared resource.

The semaphore flags are active low. A token is requested by writing a zero to a semaphore latch, and is relinquished again when the same side writes a one to the latch.

The eight semaphore flags are located in a separate memory space from the dual-port RAM in the M 671342. The address space is accessed by placing a low input on the $\overline{SEM}$ pin (which acts as a chip select for the semaphore flags) and using the other control pins (address, $\overline{OE}$ and R/$\overline{W}$) as normally used in accessing a standard static RAM. Each of the flags has a unique address accessed by either side through address pins A0-A2. None of the other address pins has any effect when accessing the semaphores. Only data pin D_0 is used when writing to a semaphore. If a low level is written to an unused semaphore location, the flag will be set to zero on that side and to one on the other side (see table 2). The semaphore can now only be modified by the side showing the zero. Once a one is writen to this location from the same side, the flag will be set to one for both sides (unless a request is pending from the other side) and the semaphore can then be written to by either side.

The effect the side writing a zero to a semaphore location has of locking out the other side is the reason for the use of semaphore logic in interprocessor communication. (A thorough discussion of the use of this feature follows below). A zero written to the semaphore location from the locked-out side will be stored in the semaphore request latch for that side until the semaphore is relinquished by the side having control. When a semaphore flag is read its value is distributed to all data bits so that a flag set at one reads as one in all data bits and a flag set at zero reads as all zeros. The read value is latched into the output register of one side when its semaphore select ($\overline{SEM}$) and output enable ($\overline{OE}$) signals go active. This prevents the semaphore changing state in the middle of a read cycle as a result of a write issued by the other side. Because of this latch, a repeated read of a semaphore flag in a test loop must cause either signal ($\overline{SEM}$ or $\overline{OE}$) to go inactive, otherwise the output will never change.

The semaphore must use a WRITE/READ sequence in order to ensure that no system level conflict will occur. A processor requests access to shared resources by attempting to write a zero to a semaphore location. If the semaphore is already in use, the semaphore request latch will contain a zero, yet the semaphore flag will appear as a one, and the processor will detect this status in the subsequent read (see table 2). For example, assume a processor writes a zero to the left port at a free semaphore location. On a subsequent read, the processor will verify that it has written successfully to that location and will assume control over the resource concerned. If a processor on the right side then attempts to write a zero to the same semaphore flag it will fail, as will be verified by a subsequent read returning a one from the semaphore location on the right side has a READ/WRITE sequence been used instead, system conflict problems could have occurred during the interval between the read and write cycles.

It must be noted that a failed semaphore request needs to be followed by either repeated reads or by writing a one to the same location. The simple logic diagram for the semaphore flag in figure 2 illusrates the reason for this quite clearly. Two semaphore request latches feed into a semaphore flag. The first latch to send a zero to the semaphore flag will force its side of the semaphore flag low and other side high. This status will be maintained until a one is written to the same semaphore request latch. Sould a zero be written to the other side's semaphore request latch in the meantime, the semaphore flag will flip over to this second side as soon as a one is written to the first side's request latch. The second side's flag will now stay low until its semaphore request latch is changed to a one. Thus, clearly, if a semaphore flag is requested and the processor requesting it no longer requires access to the resource, the entire system can hang up until a one is written to the semaphore request latch concerned.

Semaphore timing becomes critical when both sides request the same token by attempting to write a zero to it at the same time. Semaphore logic is specially conceived to resolve this problem. The logic ensures that only one side will receive the token if simultaneous requests are made. The first side to make a request will receive the token where request do not arrive at the same time. Where they do arrive at the same time, the logic will assign the token arbitrarily to one of the ports.

It should be noted, however, that semaphores alone do not guarantee that access to a resource is secure. As with any powerful programming technique, errors can be introduced if semaphores are misused or misinterpreted. Code integrity is of the utmost performance when semaphores are being used instead of slower, more restrictive hardware-intensive systems.

Semaphore initialization is not automatic and must therefore be incorporated in the power up initialization procedures. Since any semaphore flag containing a zero must be reset to one, initialization should write a one to all request flags from both sides to ensure that they will be available when required.

Using Semaphores - Some examples

Perhaps the simplest application of semaphores is their use as resource markers for the M 671342's dual-port RAM. If it is necessary to split the 4 k × 8 RAM into two 2 K × 8 blocks which are to be dedicated to serving either the left or right port at any one time. Semaphore 0 can be used to indicate which side is controlling the lower segment of memory and semaphore 1 can be defined as indicating the upper segment of memory.

To take control of a resource, in this case the lower 2 k of a dual-port RAM, the left port processor would then write a zero into semaphore flag 0 and then read it back. If successful in taking the token (reading back a zero rather than a one), the left processor could then take control of the lower 2 k of RAM. If the right processor attempts to perform the same function to take control of the resource after the left processor has already done so, it will read back a one in response to the attempted write of a zero into semaphore 0. At this point the software may choose to attempt to gain control of the second 2 k segment of RAM by writing and then reading a zero in semaphore 1. If successful, it will lock out the left processor.

Once the left side has completed its task it will write a one to semaphore 0 and may then attempt to access semaphore 1. If semaphore 1 is still occupied by the right side, the left side may abandon its semaphore request and perform other operations until it is able to write and then read a zero in semaphore 1. If the right processor performs the same operation with semaphore 0, this protocol would then allow the two processes to swap 2 k blocks of dual-port RAM between one another.

The blocks do not have to be any particular size, and may even be of variable size depending on the complexity of the software using the semaphore flags. All eight semaphores could be used to divide the dual-port RAM or other shared resources into eight parts. Semaphores can even be assigned different meanings on each side, rather than having a common meaning as is described in the above example.

Semaphores are a useful form of arbitration in systems such as disk interfaces where the CPU must be locked out of a segment of memory during a data transfer operation, and the I/O device cannot tolerate any wait states. If semaphores are used, both the CPU and the I/O device can access assigned memory segments, without the need for wait states, once the two devices have determined which memory area is barred to the CPU.

Semaphores are also useful in applications where no memory WAIT state is available on one or both sides. On a semaphore handshake has been performed, both processors can access their assigned RAM segments at full speed.

Another application is in complex data structures. Block arbitration is very important in this case, since one processor may be responsible for building and updating a data structure whilst the other processor reads and interprets it. A major error condition may be created if the interpreting processor reads an incomplete data structure. Some sort of arbitration between the two different processors is therefore necessary. The building processor requests access to the block, locks it and is then able to enter the block to update the data structure. Once the update is completed the data structure may be released.

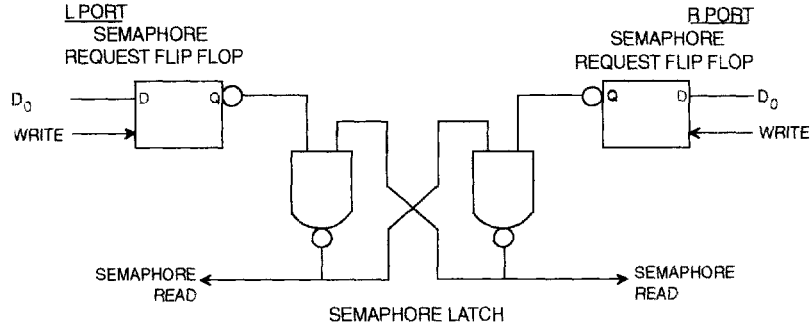
This allows the interpreting processor, to return to read the complete data structure, thus ensuring a consistent data structure.

Table 2 : Example Semaphore Procurement Sequence

Function	$D_0 - D_7$ Left	$D_0 - D_7$ Right	Status
No Action	1	1	Semaphore free
Left Port Writes "0" to Semaphore	0	1	Left Port has semaphore token
Right Port Writes "0" to Semaphore	0	1	No change. Right side has no write access to semaphore
Left Port Writes "1" to Semaphore	1	0	Right port obtains semaphore token
Left Port Writes "0" to Semaphore	1	0	No change. Left port has no write access to semaphore
Right Port Writes "1" to Semaphore	0	1	Left port obtains semaphore token
Left Port Writes "1" to Semaphore	1	1	Semaphore free
Right Port Writes "0" to Semaphore	1	0	Right port has semaphore token
Right Port Writes "1" to Semaphore	1	1	Semaphore free
Left Port Writes "0" to Semaphore	0	1	Left port has semaphore token
Left Port Writes "1" to Semaphore	1	1	Semaphore free

Note : 1. This table denotes a sequence of events for only one of the 8 semaphores on the M671342

Figure 2. M671342 – Semaphore Logic



Electrical Characteristics

Absolute Maximum Ratings

Supply voltage (VCC-GND) : ..... – 0.3 V to 7.0 V
Input or output voltage applied : ..... (GND – 0.3 V) to (VCC + 0.3 V)
Storage temperature : ..... – 65 °C to + 150 °C

Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS 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

1

	Operating Supply Voltage	Operating Temperature
Military	V_{CC} = 5 V ± 10 %	– 55 °C to + 125 °C
Industrial	V_{CC} = 5 V ± 10 %	– 40 °C to + 85 °C
Commercial	V_{CC} = 5 V ± 10 %	0 °C to + 70 °C
Automotive	V_{CC} = 5 V ± 10 %	– 40 °C to + 125 °C

DC Parameters

Parameter	Description	Version	671342-18	671342-20		671342-25		671342-30		Unit	Value
			COM only	COM	IND MIL	COM	IND MIL	COM	IND MIL		
I_{CCSB} (1)	Standby supply current (Both ports TTL level inputs)	V L	10 40	10 40	10 50	10 40	10 50	10 40	10 50	mA mA	Max Max
I_{CCSB1} (2)	Standby supply current (Both ports CMOS level inputs)	V L	25 500	25 500	50 1000	25 500	50 1000	25 500	50 1000	µA µA	Max Max
I_{CCOP} (3)	Operating supply current (Both ports active)	V L	200 220	200 220	200 220	180 200	190 210	180 200	190 210	mA mA	Max Max
I_{CCOP1} (4)	Operating supply current (One port active – One port standby)	V L	140 160	140 160	140 160	120 140	130 150	120 140	130 150	mA mA	Max Max

DC Parameters (continued)

Parameter	Description	Version	671342-35		671342-45		Unit	Value
			COM	IND MIL	COM	IND MIL		
I_{CCSB} (1)	Standby supply current (Both ports TTL level inputs)	V L	10 40	10 50	10 40	10 50	mA mA	Max Max
I_{CCSB1} (2)	Standby supply current (Both ports CMOS level inputs)	V L	25 500	50 1000	25 500	50 1000	µA µA	Max Max
I_{CCOP} (3)	Operating supply current (Both ports active)	V L	170 190	180 200	170 190	180 200	mA mA	Max Max
I_{CCOP1} (4)	Operating supply current (One port active – One port standby)	V L	110 130	120 140	110 130	120 140	mA mA	Max Max

Notes : 1. $\overline{CS}_L = \overline{CS}_R \geq 2.2$ V
2. $\overline{CS}_L = \overline{CS}_R \geq VCC - 0.2$ V
3. Both ports active – Maximum frequency – Outputs open – $\overline{OE}$ = VIH
4. One port active (f = fMAX) – Output open – One port stand-by TTL or CMOS Level Inputs – $\overline{CS}_L = \overline{CS}_R \geq 2.2$ V

Parameter	Description	671342 -18/-20/-25/-30/-35/-45	Unit	Value
IL I/O (5)	Input/Output leakage current	± 10	µA	Max
VIL (6)	Input low voltage	0.8	V	Max
VIH(6)	Input high voltage	2.2	V	Min
VOL (7)	Output low voltage (I/O_0-I/O_{15})	0.4	V	Max
VOH (7)	Output high voltage	2.4	V	Min
C IN	Input capacitance	5	pF	Max
C OUT	Output capacitance	7	pF	Max

Notes : 5. V_{CC} = 5.5 V, Vin = Gnd to V_{CC}, $\overline{CS}$ = VIH, Vout = 0 to V_{CC}
6. VIH *max* = V_{CC} + 0.3 V, VIL *min* = –0.3 V *or* –1 V *pulse width* 50 ns
7. V_{CC} min, IOL = 4 mA, IOH = –4 mA

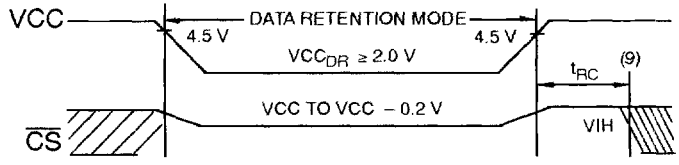
Data-retention Mode

MHS CMOS RAMs are designed with battery backup in mind. Data retention voltage and supply current are guaranteed over temperature. The following rules insure data retention :

1 – Chip select ($\overline{CS}$) must be held high during data retention ; within V_{CC} to V_{CC} - 0.2 V

2 – $\overline{CS}$ must be kept between V_{CC} + 0.3 V and 70 % of V_{CC} during the power up and power down transitions.

3 – The RAM can begin operation > t_{RC} after V_{CC} reaches the minimum operating voltage (4.5 volts).



Timing

Parameter	Test Conditions (9)	Max	Unit
ICC_{DR1}	@VCC_{DR} = 2 V	20	µA
ICC_{DR2}	@VCC_{DR} = 3 V	30	µA

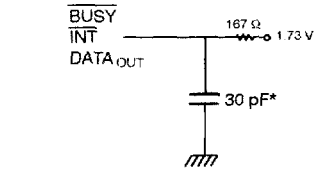
Notes : 8. $\overline{CS}$ = V_{CC}, Vin = Gnd to V_{CC}
9. t_{RC} = Read cycle time

Test Conditions (8)

AC Test Conditions

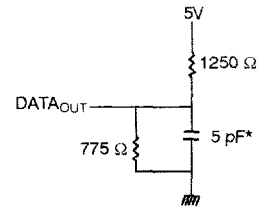
Input Pulse Levels : GND to 3.0 V
Input Rise/Fall Times : 5 ns
Input Timing Reference Levels : 1.5 V
Output Reference Levels : 1.5 V
Output Load : see figures 1, 2

Figure 3. Output Load.



* Including scope and jig

Figure 4. Output Load (for t_{HZ}, t_{WZ} and t_{OW}).



1

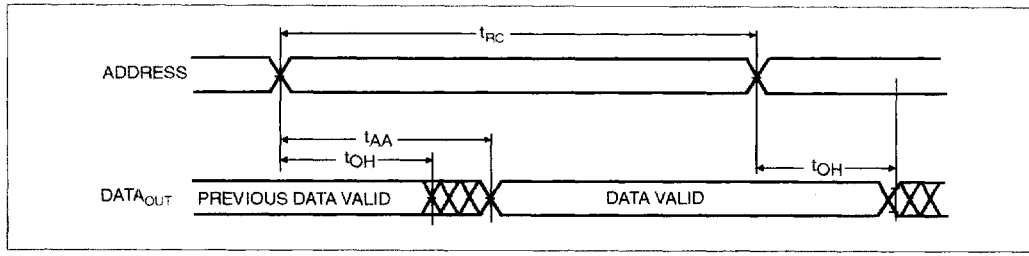
AC Electrical Characteristics

Over the Full Operating Temperature and Supply Voltage Range

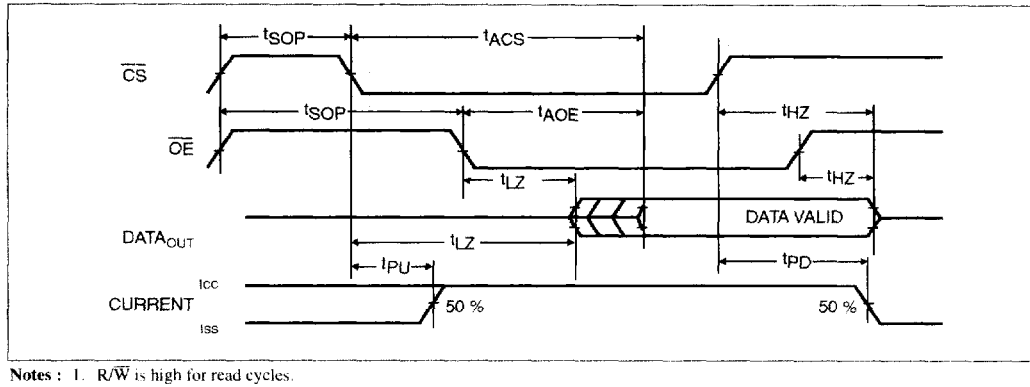
Read Cycle		Parameter	M671342-18 COM only		M671342-20		M671342-25		M671342-30		M671342-35		M671342-45		Unit
Symbol (4)	Symbol (5)		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
TAVAVR	t_{RC}	Read cycle time	18	–	20	–	25	–	30	–	35	–	45	–	ns
TAVQV	t_{AA}	Address access time	–	18	–	20	–	25	–	30	–	35		45	ns
TELQV	t_{ACS}	Chip Select access time (3)	–	18	–	20	–	25	–	30	–	35	–	45	ns
TGLQV	t_{AOE}	Output enable access time	–	10	–	11	–	13	–	15	–	20	–	25	ns
TAVQX	t_{OH}	Output hold from address change	3	–	3	–	3	–	3	–	3	–	3	–	ns
TELQZ	t_{LZ}	Output low Z time (1, 2)	3	–	3	–	3	–	3	–	3	–	3	–	ns
TEHQZ	t_{HZ}	Output high Z time (1, 2)	–	13	–	15	–	15	–	15	–	15	–	20	ns
TPU	t_{PU}	Chip Select to power up time (2)	0	–	0	–	0	–	0	–	0	–	0	–	ns
TPD	t_{PD}	Chip disable to power down time (2)	–	50	–	50	–	50	–	50	–	50	–	50	ns
TSOP	t_{SOP}	SEM flag update pulse ($\overline{OE}$ or $\overline{SEM}$)	10	–	12	–	12	–	15	–	15	–	15	–	ns
TWDD	t_{WDD}	Write pulse to data delay (6)	–	40	–	45	–	50	–	55	–	60	–	70	ns
TDDD	t_{DDD}	Write data valid to read data delay (6)	–	28	–	30	–	35	–	40	–	45	–	55	ns

Notes : 1. Transition is measured ± 500 mV from low or high impedance voltage with load (figures 1 and 2)
2. This parameters is guaranteed but not tested
3. To access RAM $\overline{CS}$ = V_{IL}, $\overline{SEM}$ = V_{IH}. To access semaphore $\overline{CS}$ = V_{IH}, $\overline{SEM}$ = V_{IL}. Refer to table 1
4. STD symbol
5. ALT symbol
6. Port-to-port delay through RAM cells from writing port to reading port, refer to Timing Waveforms of Read with Port-to-port delay

Timing Waveform of Read Cycle n° 1, Either Side (1, 2, 4)

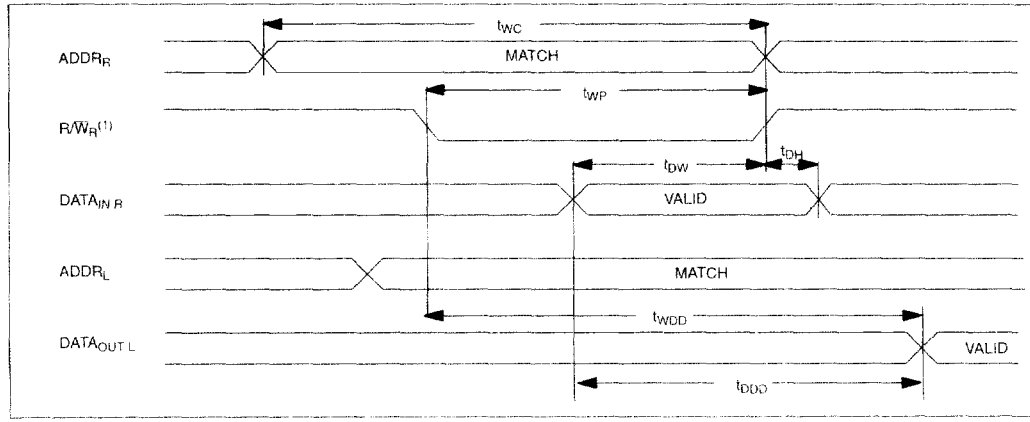


Timing Waveform of Read Cycle n° 2, Either Side (1, 3, 5)



Notes : 1. R/$\overline{W}$ is high for read cycles.
2. Device is continuously enabled, $\overline{CS}$ = V_{IL}. This waveform cannot be used for semaphore reads.
3. Addresses valid prior to or coincident with $\overline{CS}$ transition low.
4. $\overline{OE}$ = V_{IL}.
5. To access RAM, $\overline{CS}$ = V_{IL}, $\overline{SEM}$ = V_{IH}. To access semaphore, $\overline{CS}$ = V_{IH}, $\overline{SEM}$ = V_{IL}. Refer to table 1.

Timing Waveform of Read with Port-to-Port Delay (1, 2)



Notes : 1. Write cycle parameters should be adhered to, in order to ensure proper writing.
2. Device is continuously enabled for both ports.

1

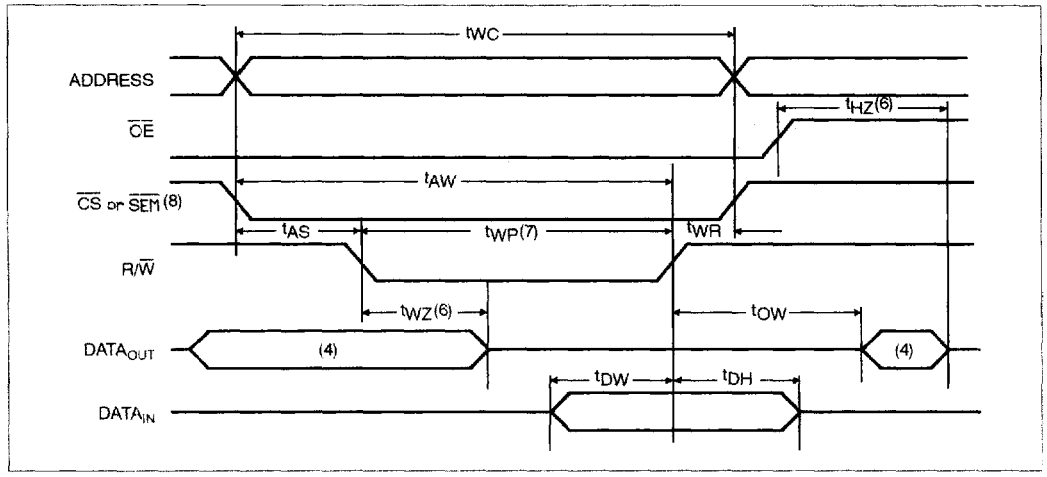
AC Electrical Characteristics

Over the Full Operating Temperature and Supply Voltage Range

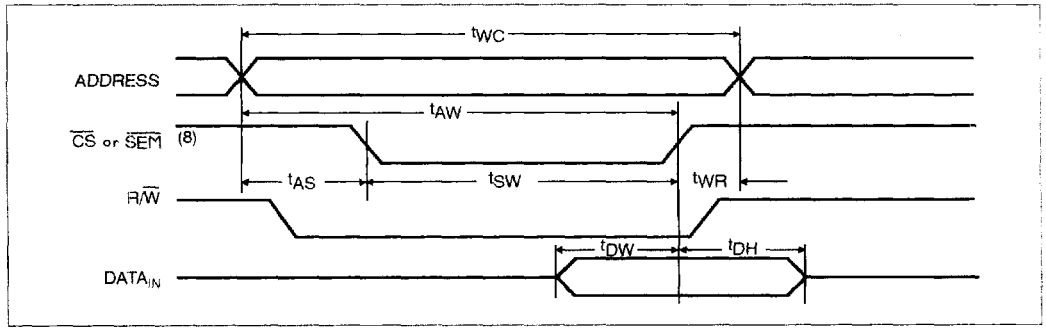
Write Cycle		Parameter	M671342-18 COM only		M671342-20		M671342-25		M671342-30		M671342-35		M671342-45		Unit
Symbol (5)	Symbol (6)		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
TAVAVW	t_{WC}	Write cycle time	18	–	20	–	25	–	30	–	35	–	45	–	ns
TELWH	t_{SW}	Chip select to end of write (3)	13	–	15	–	20	–	25	–	30	–	40	–	ns
TAVWH	t_{AW}	Address valid to end of write	13	–	15	–	20	–	25	–	30	–	40	–	ns
TAVWL	t_{AS}	Address Set-up Time	0	–	0	–	0	–	0	–	0	–	0	–	ns
TWLWH	t_{WP}	Write Pulse Width	13	–	15	–	20	–	25	–	30	–	35	–	ns
TWHAX	t_{WR}	Write Recovery Time	0	–	0	–	0	–	0	–	0	–	0	–	ns
TDVWH	t_{DW}	Data Valid to end of write	9	–	10	–	15	–	20	–	25	–	25	–	ns
TGHQZ	t_{HZ}	Output high Z time (1, 2)	–	13	–	15	–	15	–	15	–	15	–	20	ns
TWHDX	t_{DH}	Data hold time (4)	0	–	0	–	0	–	0	–	0	–	0	–	ns
TWLQZ	t_{WZ}	Write enable to output in high Z (1, 2)	–	13	–	15	–	15	–	15	–	15	–	20	ns
TWHQX	t_{OW}	Output active from end of write (1, 2, 4)	0	–	0	–	0	–	0	–	0	–	0	–	ns
TSWRD	t_{SWRD}	SEM flag write to read time	10	–	10	–	10	–	10	–	10	–	10	–	ns
TSPS	t_{SPS}	SEM flag contention window	10	–	10	–	10	–	10	–	10	–	10	–	ns

Notes : 1. Transition is measured ± 500 mV from low or high impedance voltage with load (figures 1 and 2)
2. This parameters is guaranteed but not tested
3. To access RAM $\overline{CS} = V_{IL}$, $\overline{SEM} = V_{IH}$. To access semaphore $\overline{CS} = V_{IH}$, $\overline{SEM} = V_{IL}$. This condition must be valid for entire t_{SW} time.
4. The specification for t_{DH} must be met by the device supplying write data to the RAM under all operating conditions. Although t_{DH} and t_{OW} values will vary over voltage and remperature, the actual t_{DH} will always be smaller than the actual t_{OW}.
5. STD Symbol.
6. ALT Symbol.

Timing Waveform of Write Cycle n° 1, R/$\overline{W}$ Controlled Timing (1, 2, 3, 7)

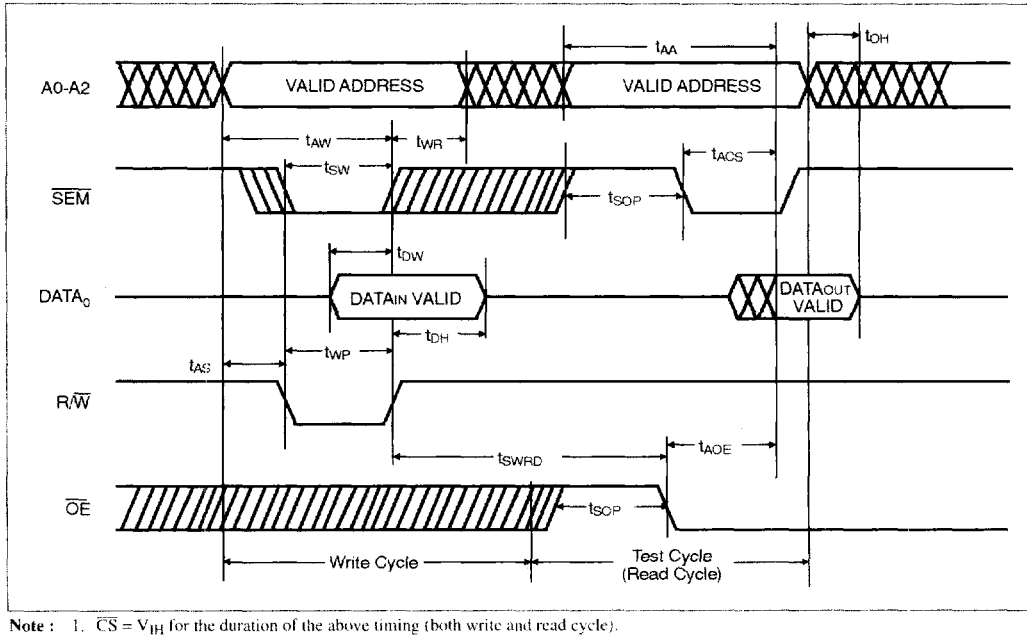


Timing Waveform of Write Cycle n° 2, $\overline{CS}$ Controlled Timing (1, 2, 3, 5)



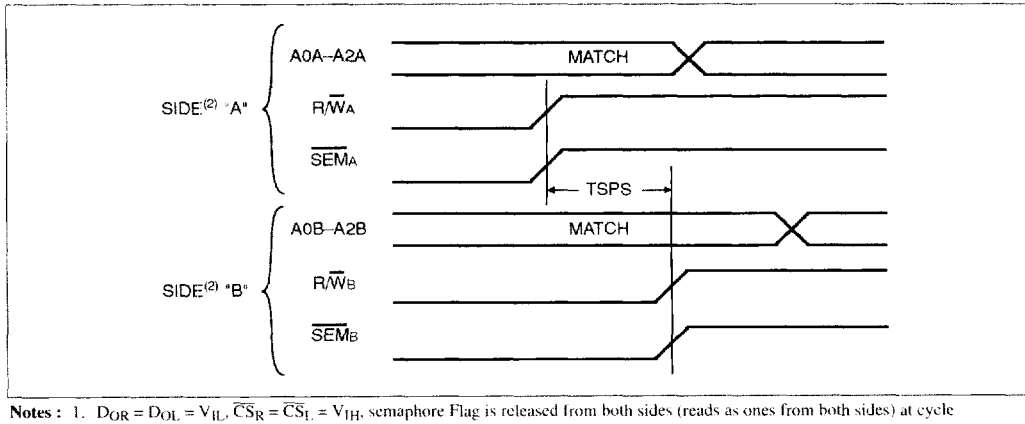
Notes :
1. R/$\overline{W}$ must be high during all adress transitions.
2. A write occurs during the overlap (t_{SW} or t_{WP}) of a low $\overline{CS}$ or $\overline{SEM}$ and a low R/$\overline{W}$.
3. T_{WR} is measured from the earlier of $\overline{CS}$ or R/$\overline{W}$ (or $\overline{SEM}$ or R/$\overline{W}$) going high to the end of write cycle.
4. During this period, the I/O pins are in the output state, and input signals must not be applied.
5. If the $\overline{CS}$ or $\overline{SEM}$ low transition occurs simultaneously with or after the R/$\overline{W}$ low transition, the outputs remain in the high impedance state.
6. Transition is measured ± 500 mV from steady state with a 5 pF load (including scope and jig). This parameter is sampled and not 100 % tested.
7. If $\overline{OE}$ is low during a R/$\overline{W}$ controlled write cycle, the write pulse width must be the larger of t_{WP} or (t_{WZ} + t_{DW}) to allow the I/O drivers to turn off and data to be placed on the bus for the required t_{DW}. If $\overline{OE}$ is high during an R/$\overline{W}$ controlled write cycle, this requirement does not apply and the write pulse can be as short as the specified t_{WP}.
8. To access RAM, $\overline{CS}$ = V_{IL}, $\overline{SEM}$ = V_{IH}.

Timing Waveform of Semaphore Read after Write Timing, Either Side[1]



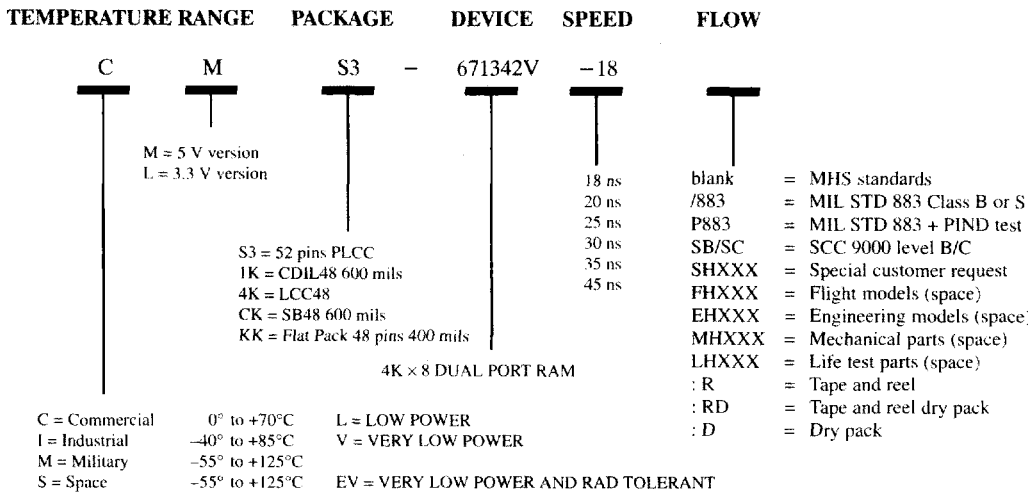
Note : 1. $\overline{CS} = V_{IH}$ for the duration of the above timing (both write and read cycle).

Timing Waveform of Semaphore Contention[1, 3, 4]



Notes : 1. $D_{OR} = D_{OL} = V_{IL}$, $\overline{CS}_R = \overline{CS}_L = V_{IH}$, semaphore Flag is released from both sides (reads as ones from both sides) at cycle start.
2. Either side "A" = left and side "B" = right, or side "A" = right and side "B" = left.
3. This parameter is measured from the point where $R/\overline{W}_A$ or $\overline{SEM}_A$ goes high until $R/\overline{W}_B$ or $\overline{SEM}_B$ goes high.
4. If t_{SPS} is violated, the semaphore will fall positively to one side or the other, but there is no guaranteed which side will obtain the flag.

Ordering Information



Military and Space Versions

The following tables give package/consumption/access time/process flow available combinations

Temp. range	Packages	Consumption		Access Time (ns)						Std process 67134		RT process 67134E	
		V	L	20	25	30	35	45	55	std	/883	/883	ESA-SCC
M	1K	●	●	●	●	●	●	●	●	●	●		
	4K	X	X	X	X	X	X	X	X	X	X		
	CK	X	X	X	X	X	X	X	X	X	X		
	KK	X	X	X	X	X	X	X	X	X	X		
	0	X	X	X	X	X	X	X	X	X	X		
S	4K	X			X	X	X	X	X			X	X
	CK	X			X	X	X	X	X			X	X
	KK	X			X	X	X	X	X			X	X
	0	X	X		X	X	X	X	X			X	X

● = product in production

X = call sales office for availibility