read from or written to. A data signal is bidirectional, its state indicating whether it the data is to be written or read.

Control signals include chip enable ($\overline{CE}$), read/write (R/$\overline{W}$), and output enable ($\overline{OE}$). Dual-port RAMs with semaphores also use a semaphore-enable control signal ($\overline{SEM}$). And there are two flags, called $\overline{INT}$ and $\overline{BUSY}$.

A *low* on the chip-enable input allows that port to become functional. Data is either read from the internal dual-port RAM array or written into it, depending upon the state of the read/write signal—e.g., a *low* initiates a write operation. The three-state data-output drivers are enabled by a *low* output enable.

When one port writes to a predetermined mailbox, an interrupt to the other port is generated. When the interrupted port reads that memory location, the interrupt is reset.

When both ports address the same memory location and both chip enables are active (*low*), the address is said to be in contention. A contention requires an arbitration (by a master) to determine ownership of the memory location, which is then is assigned to the winner. An active (*low*) $\overline{BUSY}$ signal notifies the loser of the arbitration.

Interrupt logic

It's necessary to assert a port's chip enable for the port to read from or write to any location, including the mailboxes. (Note that you can use the mailbox locations as conventional memory by not connecting the interrupt line to the appropriate processor.) The two uppermost memory locations can be used for message passing.

The highest memory location serves as the mailbox for the right processor. When the left processor writes to this mailbox, the interrupt (request) to the right processor, $\overline{INT_R}$, goes *low*. When the right processor reads its mailbox, the flip-flop is reset and $\overline{INT_R}$ goes *high*.

The second-highest memory location serves as the mailbox for the left processor. When the right processor writes to this mailbox, the interrupt (request) to the left processor, $\overline{INT_L}$, goes *low*. When the left processor reads its mailbox, the flip-flop is reset and $\overline{INT_L}$ goes *high*.

Each port can read the other port's mailbox without resetting the associated flip-flop. If your application does not require message passing, leave the appropriate pin open. Do not connect a pull-up resistor to the pin, and do not connect the pin to the processor's interrupt request pin.

Note that the active state of the $\overline{BUSY}$ signal prevents a port from setting the interrupt to the winning port. Additionally, an active $\overline{BUSY}$ signal to a port prevents that port from reading its own mailbox and thus resetting the interrupt. These operations are ramifications of the data-ownership concept.

Both ports reading. If both ports of a dual-port RAM read the same location at the same time, you can assume that both ports read the same data. When arbitration occurs as a result of contention in a Cypress dual-port RAM, the port that wins the arbitration gets temporary ownership of the memory location. The losing port can read the memory location, but the $\overline{BUSY}$ signal tells it that it lost the arbitration.

One port reading, one writing. Arbitration will allocate priority to either the reading or the writing port. In Cypress dual-port RAMs, if the losing port is attempting to write data, the write is inhibited so that the data in memory is not corrupted. The $\overline{BUSY}$ flag to the losing port signals that the write was not performed.

If the losing port is attempting to read data, the data is either old or new, or it may be some random combination of the two. The $\overline{BUSY}$ flag to the losing port signals that the old data is still being read on the losing port's data lines. The old data will remain undisturbed for an access time after either $\overline{BUSY}$ on the losing port goes *high*, or the losing port's address is toggled, or $\overline{CE}$ for the losing port is toggled, or R/$\overline{W}$ for the losing port is toggled during a valid read.

If the new data is needed, use the $\overline{BUSY}$ flag to generate a delay until the new data is present or can signal a processor to attempt the read again after $\overline{BUSY}$ is cleared.

Both ports writing. If both ports were trying to write at the same time, the losing port will be prevented from writing so that the data cannot be corrupted. Again, $\overline{BUSY}$ is asserted to the losing port, indicating that the write operation was unsuccessful.

Hardware semaphores

Semaphore signaling is a popular method of allocating mutually exclusive accesses to blocks of memory that are shared among several processors. Exclusive processor control guarantees data integrity in sensitive applications such as shared I/O buffers.

Semaphore signaling also improves the efficiency of block memory accesses by preventing delays and processor stalls due to a memory location being $\overline{BUSY}$ from another processor access.

Traditionally, semaphore signaling has been implemented in software using dedicated memory locations to hold the semaphore signals. In contrast, Cypress dual-port RAMs incorporate eight on-chip hardware semaphore latches that are independent of RAM memory locations. The latches are accessed through the data and address ports in the same way as a RAM cell access, and their independence from memory allows software to allocate block addresses and block sizes. Hardware semaphores eliminate the need to use a processor with an indivisible test-and-set instruction (as required by software semaphores). Instead, semaphore control requests are accommodated with a standard write to the semaphore latch followed by a read instruction. There is no requirement to lockout other processor accesses to the semaphore between the write and read.

Hardware semaphores provide flexible software configuration of shared memory. Cypress hardware semaphores implement a token-passing scheme that permits the port in possession of the token to have exclusive access to a block of shared memory. Possession of the token can only be relinquished by the port with possession; a port's request for possession of the token will be denied if the token is held by the other port.

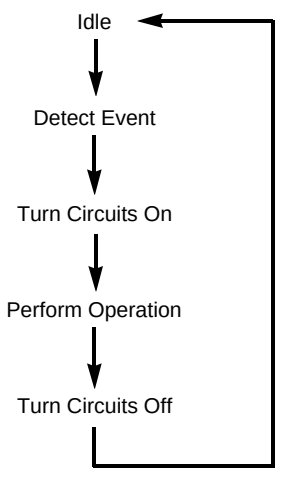
Address transition detection

All Cypress dual-port RAMs, whether master or slave, use a circuit design technique called address transition detection (ATD) to improve performance and reduce power dissipation.

ATD improves performance by equilibrating differential paths, pre-charging critical nodes, and forcing the outputs to a high-impedance state. Equilibration and pre-charging will bias critical nodes to voltage levels that are approximately in the mid-point of the small-signal operating range; thus, when data is sensed, it takes a shorter amount of time to transition to a *high* or *low*. Forcing the outputs to their high-impedance states improves speed slightly, but more importantly the technique reduces output switching noise by eliminating crowbar current and separating the output current into two pulses instead of one.

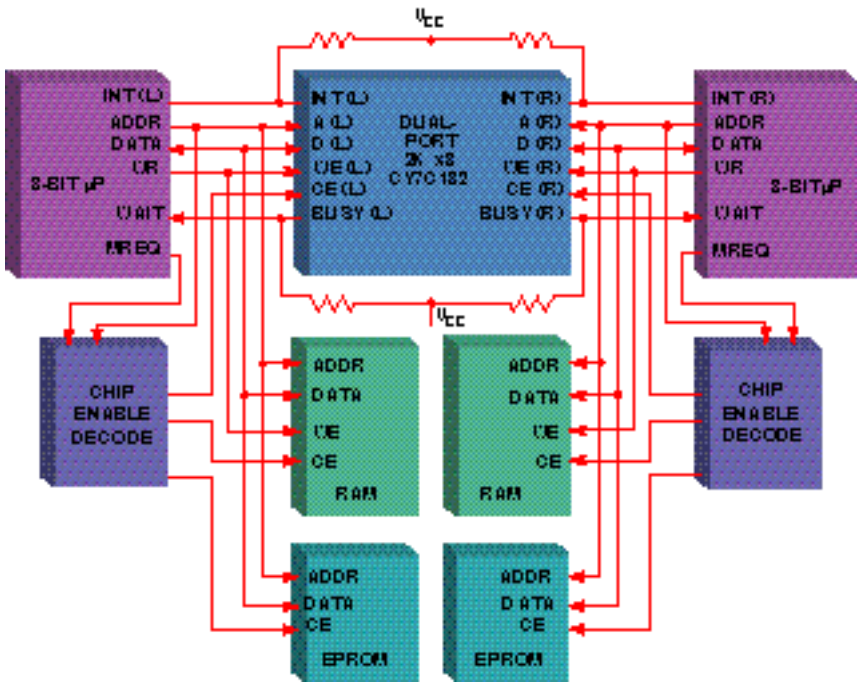
ATD minimizes power consumption because it turns on power-hungry circuits only when they are required. Slightly over half of a RAM's circuits are linear, and about 70% of the power is dissipated in the sense amplifiers during a read operation. When the RAM is operating at its maximum frequency, the ATD circuits are constantly triggered, so the power savings are minimal. At lower speeds or smaller duty cycles, however, the power savings are significant.

Here is a typical ATD sequence:

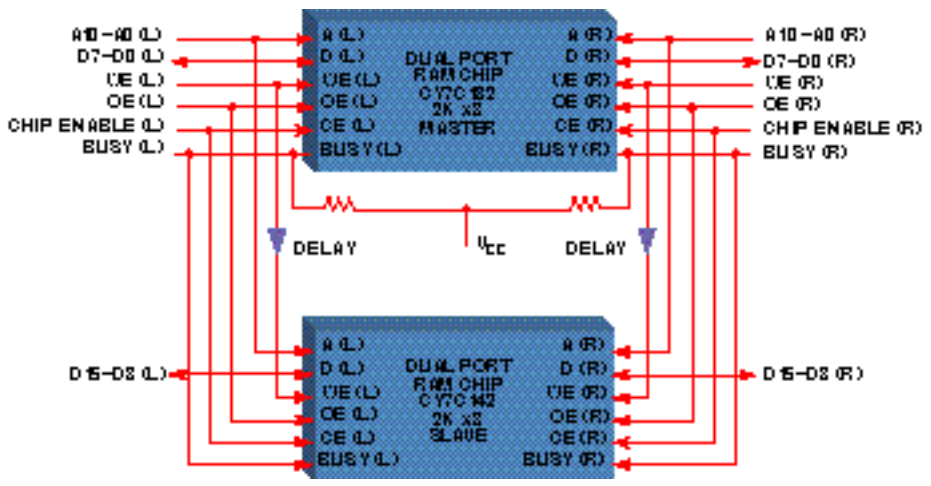


The event that triggers the ATD sequence for either port is a transition in any address, chip-enable, or read/write signal. Equilibration and pre-charging follow, then (for a read operation) turning on the sense amplifiers and latching the data or (for a write) pulling the BIT and $\overline{BIT}$ lines to the required levels at the addressed location. The master clock pulse lasts from 7 ns to 11 ns, depending upon temperature, supply voltage, and the distribution of IC processing parameters; at the end of the pulse, the data is latched and the appropriate circuits are turned off.. ❖

For literature, visit the Cypress web site. See the appropriate site address (URL) for article 203 in the listing on the back cover.



Two-processor example. This is a typical dual 8-bit microprocessor system, here implemented using a Cypress CY7C132 dual-port RAM, SRAM, and EPROM. The address lines of each processor are decoded to generate the chip-enables for the RAMs and EPROMs. Note the four pull-up resistors; two are required on the interrupt requests to the processors, and two are needed by the BUSY lines, which go to the processors'wait inputs.



Width expansion. Here's how to interconnect a Cypress CY7C132 (2K × 8) dual-port Master and a CY7C142 (2K × 8) slave to form a 16-bit-wide word. Due to package limitations, the interrupt outputs are not available at the 2K × 8 level in the 48-pin DIP. They are available in the LCC and PLCC packages, however, on the master and the slave devices alike. You can use either one, and you do not have to tie together the corresponding interrupt pins of the master and slave. (Processor interfaces and interrupt connections not shown.)