AN70118

Understanding Asynchronous Dual-Port RAMs

Author: Sheetal Chennapnoor
Associated Project: No
Related Application Notes: None

This application note provides an overview of the architecture and functionality of Cypress's asynchronous dual-port RAMs. It also provides details regarding the features supported along with a few applications of asynchronous dual-port RAMs.

Contents

Introduction

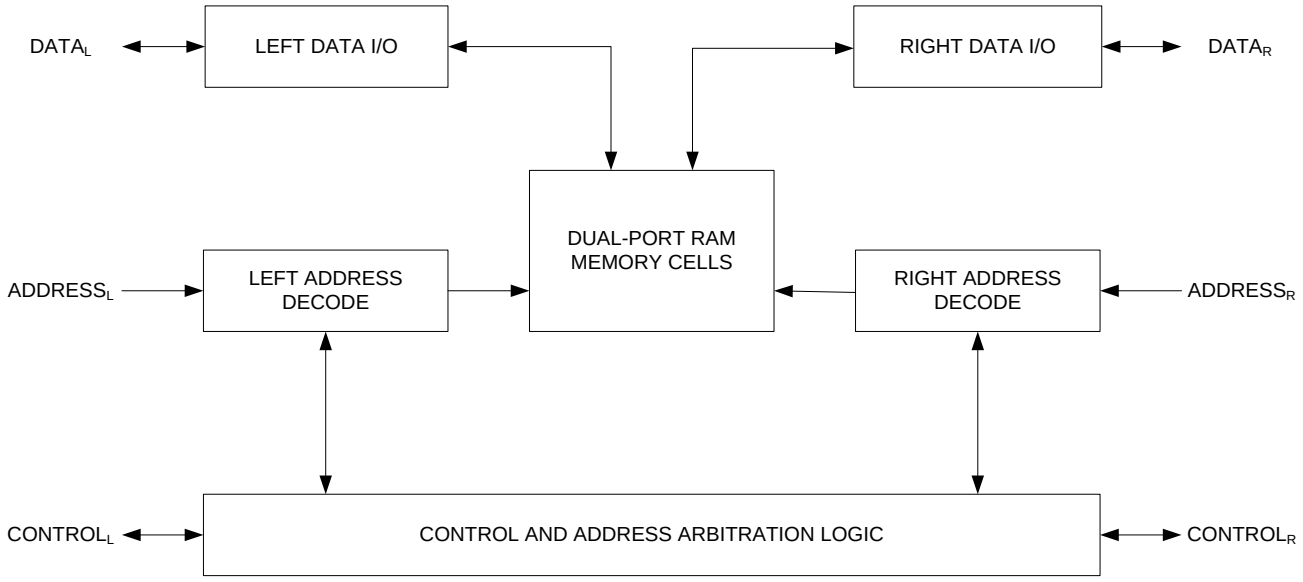
Several applications, such as wireless systems, audio and video processing, and control systems, typically use coprocessor architectures to maximize system performance. The computation stages of the application are split between the two processors based on the nature and complexity of the algorithm. The applications involve significant data transfer between the two processors. Dual-port memories are ideally suited to such applications because they provide flexible memory access from both processors. This application note describes the various design features of Cypress's asynchronous dual-port RAM device, which consists of a single memory array that supports access from two independent ports—each having a set of address, data, and control signals. The device allows simultaneous access to a single SRAM memory location from both ports. In addition, it includes the mechanism to resolve the bus contention that arises from simultaneous memory access and communication between devices that have different bus widths.

Cypress Dual-Port RAM Operation

Read/Write Operation

Asynchronous dual-port RAM responds to address and control pin inputs without the need for a clock. Figure 1 shows a simplified block diagram of a Cypress asynchronous dual-port RAM device. The device interface includes two ports, each of which has a set of signals: address, data, and control. Each signal has either the subscript L or R to designate left or right, respectively. The address pins are unidirectional inputs to the device. Their states specify the memory location to be read from or written to.

Figure 1. Dual-Port RAM



Data lines are bidirectional, and their states represent the data to be written to or read from the memory. Typical control pins are chip enable ($\overline{CE}$), read/write (R or $\overline{W}$), and output enable ($\overline{OE}$). A few of the Cypress dual-port devices have the byte select inputs ($\overline{B}_0 - \overline{B}_3$) in × 36 / × 32 bit devices and upper byte select and lower byte select ($\overline{UB}$, $\overline{LB}$) in × 16 / × 18 bit devices. Asserting these signals enables read and write operations to the corresponding bytes of the memory array. Table 1 shows the basic read/write functional operation.

In addition, Cypress asynchronous dual-port RAM has some special-purpose control pins: semaphore ($\overline{SEM}$), interrupt ($\overline{INT}$), and busy ($\overline{BUSY}$). A semaphore control pin ($\overline{SEM}$) is addressed to solve the mutual exclusion problem. When two devices need to communicate through the shared dual-port RAM, they use the status flag $\overline{INT}$. The BUSY flag indicates the occurrence of a contention when both ports address the same memory location. The following sections provide additional information on these controls pins.

Table 1. Read/Write Functional Operation

$\overline{CE}$	R/ $\overline{W}$	UB/LB or B_{0-3}	$\overline{OE}$	Operation	I/O
H	X	X	X	Power Down	HIGHZ
X	X	X	H	I/O lines disabled	HIGHZ
L	L	H/L	X	Write to selected byte	DataIN
L	H	H/L	L	Read selected byte only	DataOUT

Arbitration

The arbitration scheme depends on the address and the chip enable ($\overline{CE}$) signal. Read/write operations use arbitration logic to maintain data integrity in situations in which both ports access the same memory location. If the port performing a write operation loses the arbitration and receives a BUSY signal, it will be internally prohibited from performing the respective write operation.

Unequal Port Addresses

When the addresses of the right and left ports are not equal, no arbitration is required. Hence, both ports can simultaneously access different memory locations for write/read operations.

Ports Camped on an Address

Consider the condition in which the right port addresses a memory location that is already being addressed by the left port. In that case, the right port's BUSY signal is asserted. The device operates the same, regardless of which port is camped on an address.

Ports Accessing the Same Memory Location Simultaneously

In general, when both ports access the same memory location simultaneously, the asynchronous dual-port RAM mandates a minimum time difference between the two events. If the events occur close together in time, the probability of each port either winning or losing the arbitration is approximately equal. This parameter is called port setup time for priority and is abbreviated as t_{PS} in the datasheets. If one port addresses memory location t_{PS} before the other port, the first port is guaranteed to win. If not, the result of the subsequent arbitration is unpredictable.

Key Timing Parameters for the BUSY Signal

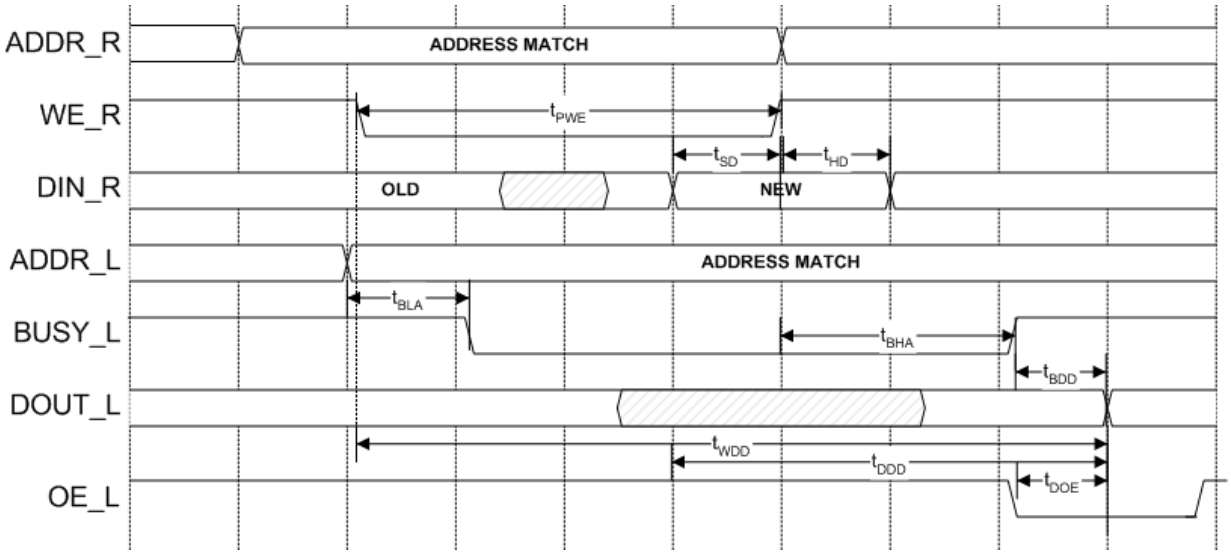
Typical key timing parameters are specified with respect to the BUSY signal. For example, BUSY LOW from address match t_{BLA} is the maximum time it takes BUSY to go LOW, as measured from the time the addresses on both ports are identical. Performance degrades if the value of t_{BLA} is greater than the memory cycle time. This period is less than the minimum cycle time for all speed grades of asynchronous dual-port RAMs.

Another parameter, t_{BHA}, is the maximum time it takes for BUSY to go HIGH after an address mismatch. This parameter indicates the time period for which the losing port cannot access the memory location owned by the opposite port and is generally lower than the minimum cycle time for all speed grades of dual-port RAMs.

The next two parameters are similar to the preceding two. The difference is that the chip enable pin controls the BUSY signal. The parameters are t_{BLC} and t_{BHC}; they are less than the minimum cycle time for all speed grades of asynchronous dual-port RAM. BUSY HIGH to valid data, t_{BDD}, is the maximum time it takes the data to become valid to the losing port after BUSY goes away. This parameter value is equivalent to address access time, t_{AA}, because a read cycle is initiated to the losing port when its BUSY signal transitions from LOW to HIGH.

Figure 2 shows the timing for the right port performing a write operation and the left port moving asynchronously to the same address and attempting to perform a read operation. The first parameter of interest is t_{DDD}, which is the maximum time between the stabilization of the data to be written by the winning port and that same data becoming valid at the outputs of the port that received the BUSY signal. The second parameter of interest is t_{WDD}, which is the maximum time between the HIGH-to-LOW transition of the winning port's write strobe and the data becoming valid at the output of the port that received the BUSY signal.

Figure 2. BUSY Timing



Key BUSY Timing Parameters:

t_{PWE}: Write pulse width

t_{SD}: Data setup to write end

t_{HD}: Data hold from write end

t_{BLA}: BUSY LOW from address match

t_{BHA}: BUSY HIGH from address mismatch

t_{BDD}: BUSY HIGH to data valid

t_{WDD}: Write pulse to data delay

t_{DDD}: Write data valid to read data valid

t_{DOE}: OE LOW to data valid

It is possible for the losing port to read old data, new data, or a combination of the two. If the read occurs early with respect to the write, old data is read else new data is read. If the read occurs at the same time that data is changing from old to new, the data read is not predictable.

To obtain a valid data read, dual-port RAM uses methods such as HIGH-to-LOW transition of the BUSY signal to the losing port; the operation can be repeated or the LOW level of the BUSY signal to the losing port to prompt one of three types of delays:

- Delay the reading of data until the data becomes valid, which occurs at access time after the LOW-to-HIGH transition of BUSY
- Insert wait states until BUSY goes HIGH
- Stretch the clock until BUSY goes HIGH

In general, the losing port has no control over the winning port. To successfully read the data just written, two operations are possible:

- Change an address line to a different address and then change it back to the original address. This toggles the BUSY signal to the losing port.
- Change the state of the chip enable. This also toggles the BUSY signal to the losing port.

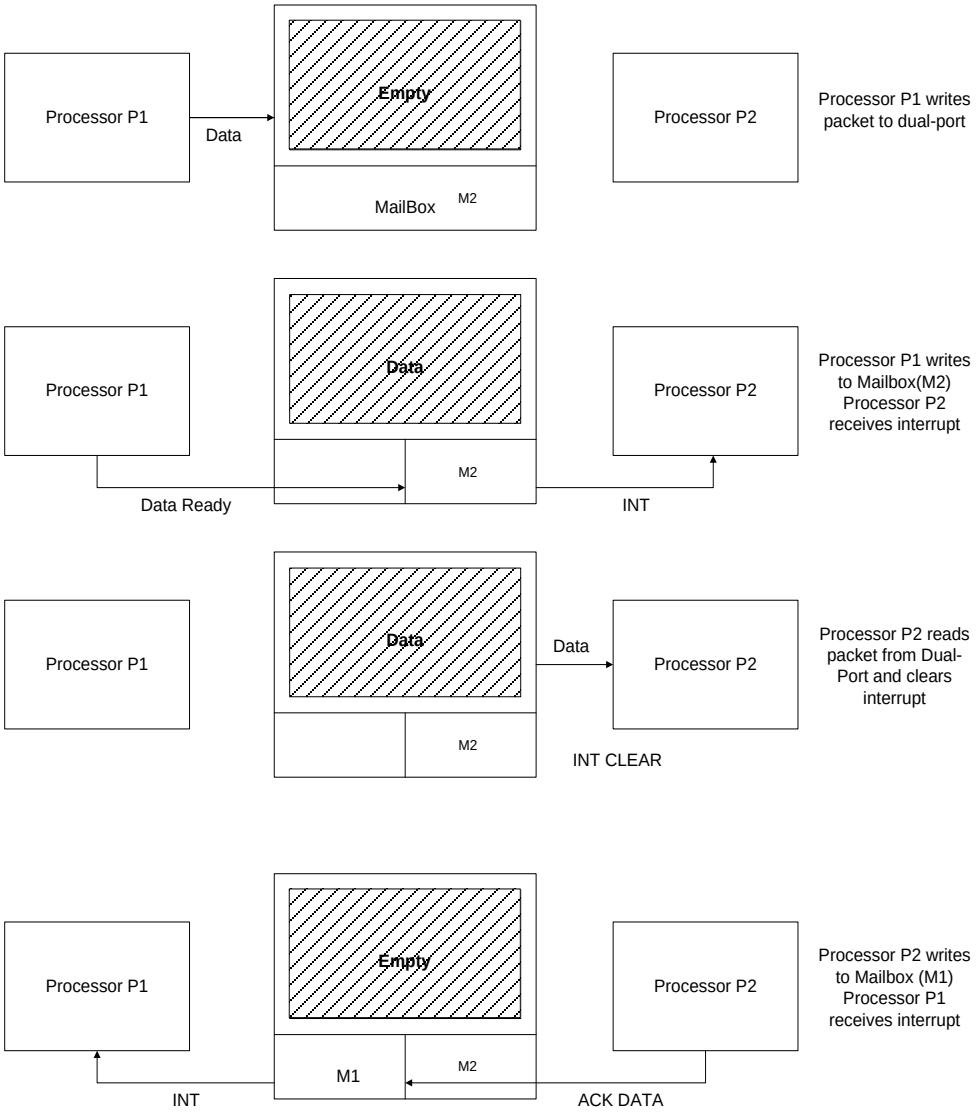
Mailbox

You can use the mailbox feature of the asynchronous dual-port RAM to pass messages in a multiprocessor environment. The upper two memory addresses are designed to act as mailboxes. For example, in a 2K, eight-xx dual-port RAM device, 7FF is the right port's mailbox and 7FE is the left port's mailbox. The paragraphs below describe the operation of this feature.

A port's chip enable must be asserted for the port to either read from or write to a specific location, including the mailboxes. Note that you can use a mailbox location as a conventional memory location by not connecting the interrupt line to the appropriate processor. When the left port writes to the highest dual-port location, an interrupt signal is set on the right port—i.e., INTR goes LOW. To send an interrupt from the right side to the left, a write operation must occur at the next to uppermost location—i.e., INTL goes LOW. In either case, the interrupt is cleared by a read of the opposite port's interrupt location. It is important to note that the read will clear the interrupt signal but the data in this location will be undisturbed. Also, each port can read the other port's mailbox without resetting the associated flip-flop.

Figure 3 shows two processors communicating using the mailbox feature of the dual-port RAM. If the application does not require this feature, the appropriate pin ($\overline{\text{INT}}$) can be left unconnected. Do not connect the pin to a pull-up resistor or to the processor interrupt request pin. Note that the active state of the BUSY signal prevents a port from setting the interrupt to the winning port. In addition, an active BUSY signal to a port prevents that port from reading its own mail and thus resetting the interrupt.

Figure 3. Mailbox Logic



Semaphores

Concept of Semaphores

A semaphore can be hardware, or it can be a software tag variable whose value indicates the status of a common resource. The purpose of a semaphore is to lock the shared resource being used. A process that needs a shared resource checks the related semaphore to determine the status of the resource based on a decision made. In multi-tasking/multi-threaded operating systems, activities are synchronized using semaphores. Semaphore signaling is a popular method of allocating mutually exclusive access to blocks of memory that are shared among several processors. This technique ensures data integrity and also improves efficiency of block memory access. It prevents delays and processor stalls caused by access to a memory location owned by another process.

Hardware Semaphores

Cypress offers dual-port RAMs with eight on-chip hardware semaphore latches that are independent from RAM memory locations. Hardware semaphores eliminate the need for atomic tests and set instructions in shared systems (see References). Their control requests are handled using a standard write to the semaphore latch followed by a read instruction. There is no requirement to lock out other processor access to the semaphore between the write and read. Hardware semaphores provide flexible software configuration of shared memory and operate independent of any memory in the RAM, allowing software to allocate block addresses and sizes.

A semaphore latch (formed by two cross-coupled NOR gates) indicates the allocated memory block for a semaphore, as Figure 4 shows. You can set the latch so only one port controls the semaphore at a time. Additional input latches on the semaphore ports are used to hold requests to set or clear the latch. An output latch on each port is used to prevent the output from changing during a read from the port.

Semaphore latches are accessed through the data and address ports in the same way as with RAM cell access. The semaphore control pin ($\overline{SEM}$) initiates a semaphore access cycle. Using address line select (A_{0-2}), the semaphore latch is accessed and the data line (D_0) is latched during a write. The other data lines are ignored. During a read, the semaphore drives all of the data lines (D_0 through D_7) with the semaphore signal.

Processor requests control of a semaphore by writing a 0 to the D_0 port of the semaphore addressed by A_{0-2}. The 0 is latched into the port's input register and is held until another write attempts to set it to 1. If the semaphore is free at the time of the request, the port will immediately be granted control of the semaphore. If the semaphore is controlled by the other port, the request for control will be denied. The controlling port can relinquish control of the semaphore only by writing a 1 to the semaphore.

Figure 4. Semaphore Latch

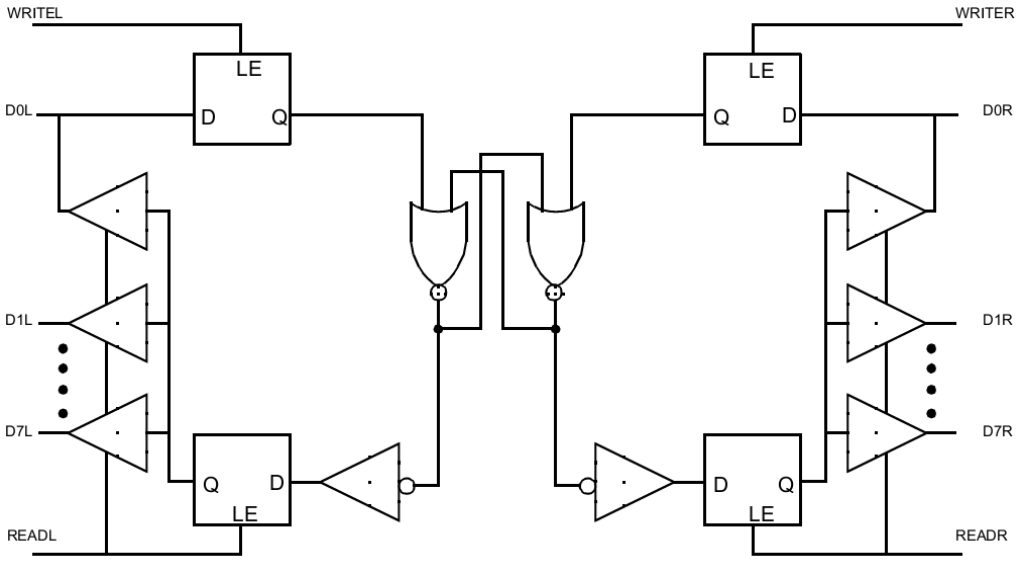
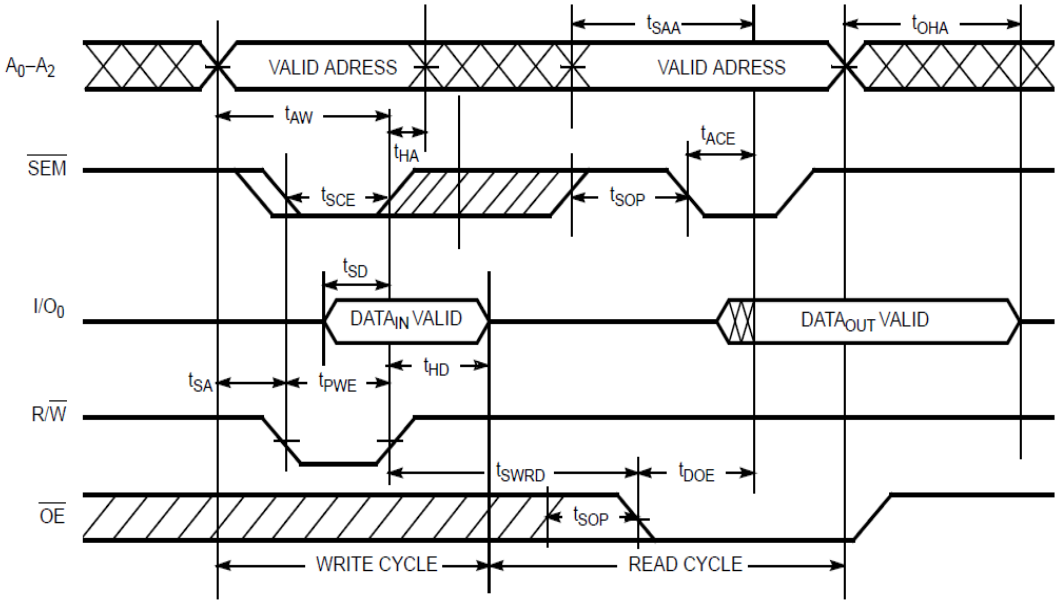


Figure 5. Timing of a Semaphore Read After a Write



To see if a request for control of the semaphore was successful, perform a read of the semaphore; if the processor reads 0 on D_0, the port controls the semaphore. If it read 1, the port does not control the semaphore. The semaphore output drives all of the data lines with the state of the semaphore, so $D_{0–7}$ will be “00000000" when control is granted and will be “11111111" when control is denied.

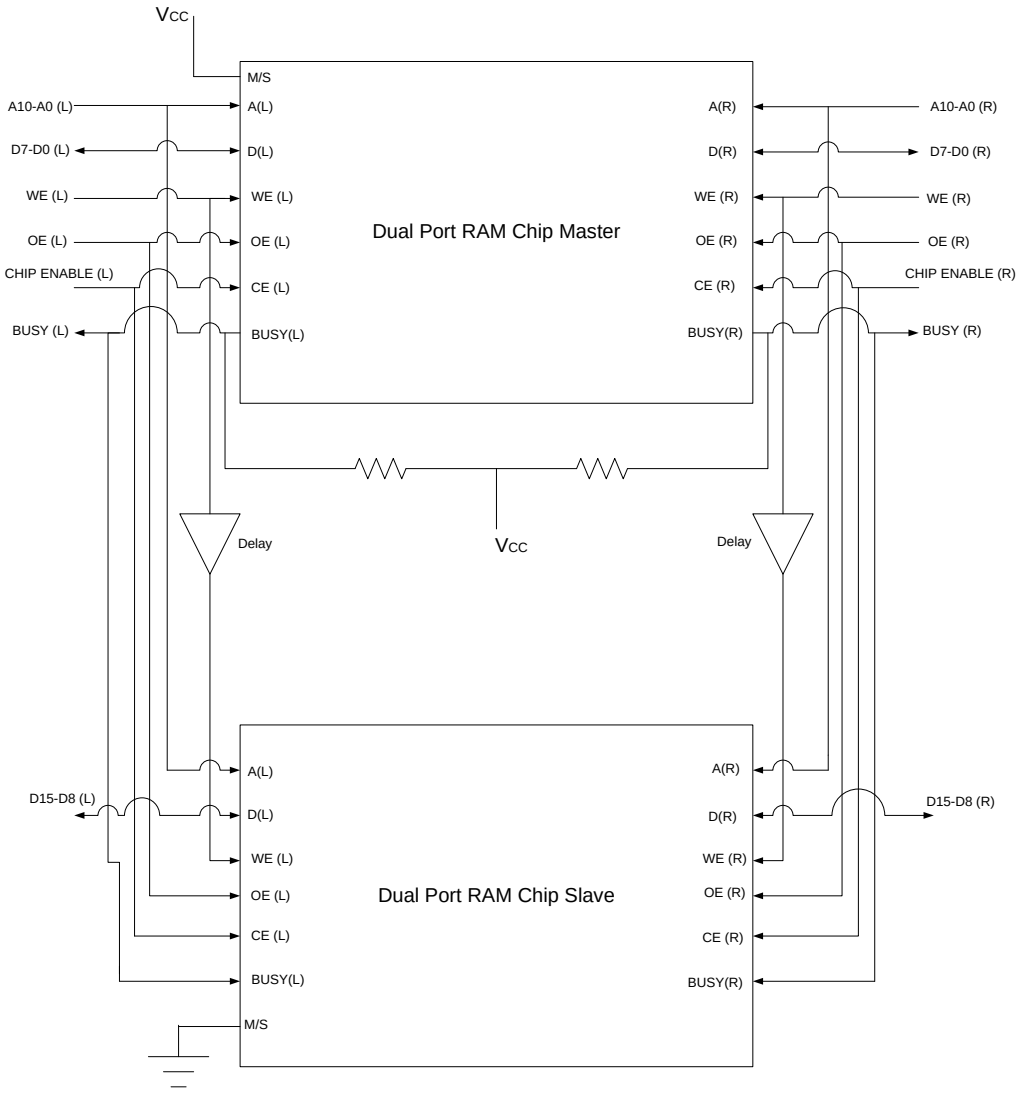
A new read cycle must be performed to update the port’s output lines. If both ports attempt to write a 0 within t_{SPS} (the semaphore flag contention window) of each other while the semaphore is free, arbitration logic will guarantee that only one side gains control of the semaphore. Figure 5 depicts the timing of a semaphore read after a write.

Word-Width Expansion

One feature of Cypress’s asynchronous dual-port RAM is data width expansion. The slave device is designed to meet the requirement of expanding these devices in width. It is important to understand the problem that can occur if two or more master devices (devices that have arbitration) are used in width expansion. There is the potential for the two devices to arbitrate differently and for a BUSY signal to be sent to both sides of the system (one from each of the two devices), which could lock the system. It is necessary to address this problem with a device that does not drive a BUSY signal but can accept input from the master device that performs the arbitration.

Figure 6 shows the interconnection of a master and slave to form word-width expanded data. The diagram does not show the interfaces to the processors or the connections for the interrupt signals.

Figure 6. Word-Width Expansion



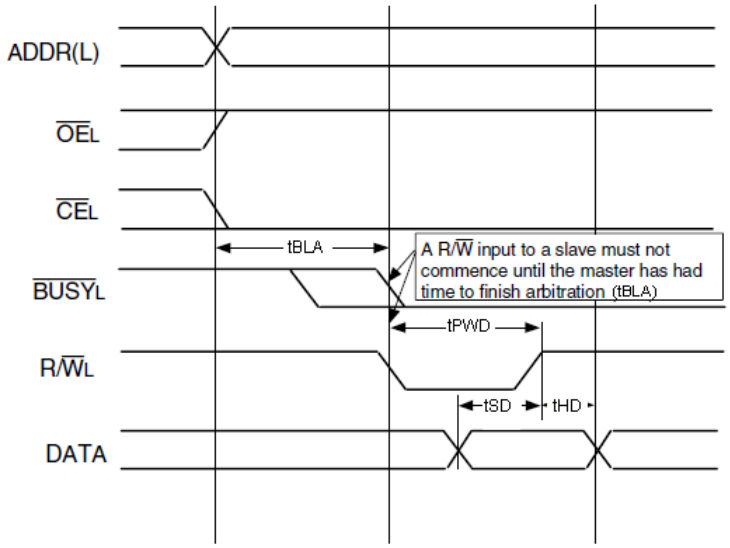
Delaying the Write Strobe

In a width-expanded mode, to compensate for the timing of the master/slave interface of the asynchronous dual-port RAM, the timing of a write cycle must be modified—i.e., the read-write (R/W) signal of the master interface must be delayed (delay element) to allow for the dual-port RAM to issue the BUSY signal. This is because the slave device used in width expansion does not have an internal hold-off mechanism, which the master device does. It is totally dependent on the master device to drive the BUSY input, which in turn disables the write to the RAM for the port that receives the BUSY signal. Figure 7 depicts the timing required for width expansion. In equation form:

$$t_{WC} = t_{BLA} + t_{PWD}$$

where the delay must be at least equal to t_{BLA}. Note that if you add more slaves to increase the word width (e.g., 24 or 32 bits), the delay elements' output can connect directly to the write strobe inputs. Additional delay elements are not required.

Figure 7. Timing Required for Width Expansion



Bus Matching

Cypress offers a bus-matching feature that allows a 36-bit bus to interface to an 18-bit or 9-bit bus or an 18-bit bus to an 9-bit bus. When designing the 36-bit dual-port SRAM in an application that operates on 8-bit bytes, 16-bit words, or 32-bit longs, take care to match the data widths of the interfaces to the dual port.

The bus-matching scheme must maintain a specific order of information; byte ordering can be either "big-endian" or "little-endian." If data is configured in a big-endian format, byte 0 is always the leftmost byte; if data is configured in a little-endian format, byte 0 is always the least significant, rightmost byte. Few of Cypress's synchronous dual-port products have big-endian byte ordering capability. Refer to AN1043 for details.

In Cypress's asynchronous dual-port products, byte ordering is little-endian. The × 36-bit data line can be configured as a 36-bit long-word, 18-bit word, or 9-bit byte format for data I/O. Figures 8, 9, and 10 represent little-endian format. In Figure 8, B0, B1, B2, and B3 are bytes within the 36-bit and 18-bit buses. In Figure 9, B0, B1, B2, and B3 are bytes within the 36-bit and 9-bit buses. In Figure 10, B0, B1, B2, and B3 are bytes within the 18-bit and 9-bit buses.

Figure 8. Little-Endian Byte Mapping Between 36-Bit and 18-Bit Buses

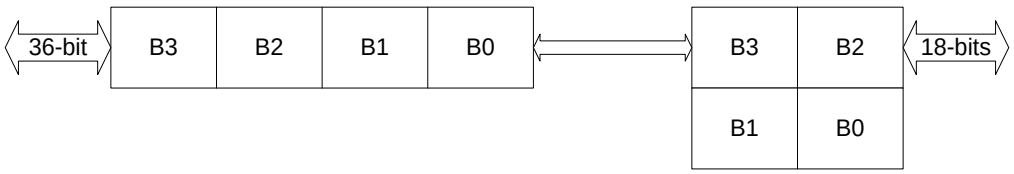


Figure 9. Little-Endian Byte Mapping Between 36-Bit and 9-Bit Buses

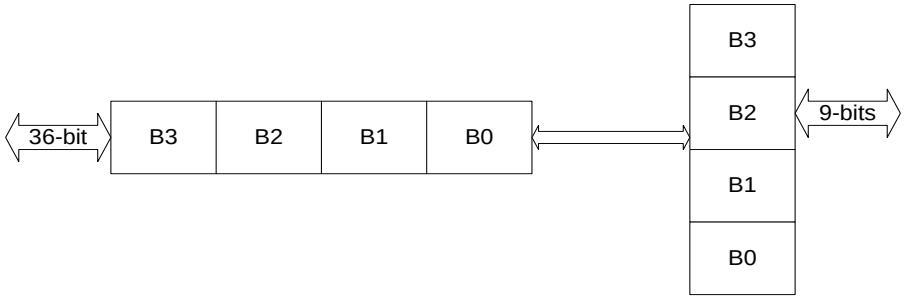
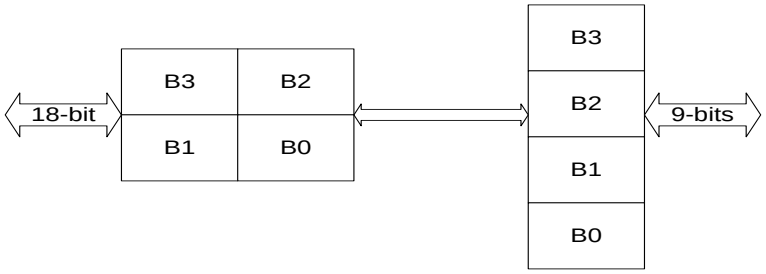


Figure 10. Little-Endian Byte Mapping Between 18-Bit and 9-Bit Buses



Bus matching is available in only one of the ports in Cypress part CY7C057V/056V (1-Mbit dual-port SRAM) and is controlled by the bus match select, bus size select, word address, and byte address control pins. The status of these pins selects the mapping: 36-bit, 18-bit, or 9-bit bus width. The bus match select pin works with the bus size select pin to select the bus width (long word, word, or byte) for the right port of the dual-port device. The data sequencing arrangement is selected using the word address and byte address input pins. Both writing to and reading from the right port is done using these pins. The level of bus match select must be static throughout the device operation.

Table 2 shows the combination of bus match select, bus size select (SIZE), byte address, and word address to select either a byte or word data arrangement on the right port. Logic HIGH on the SIZE pin when the bus match select pin is HIGH selects a byte bus (8-bit) data arrangement. Logic LOW on the SIZE pin when the bus match select pin is HIGH selects a word bus (16-bit) data arrangement. The level of the bus size select must also be static throughout normal device operation.

When writing from the right port into the mailbox 9, 18, or 36 bits, the generation of interrupt is done toward the left port.

Table 2. Byte Mapping Toward Right Port Access

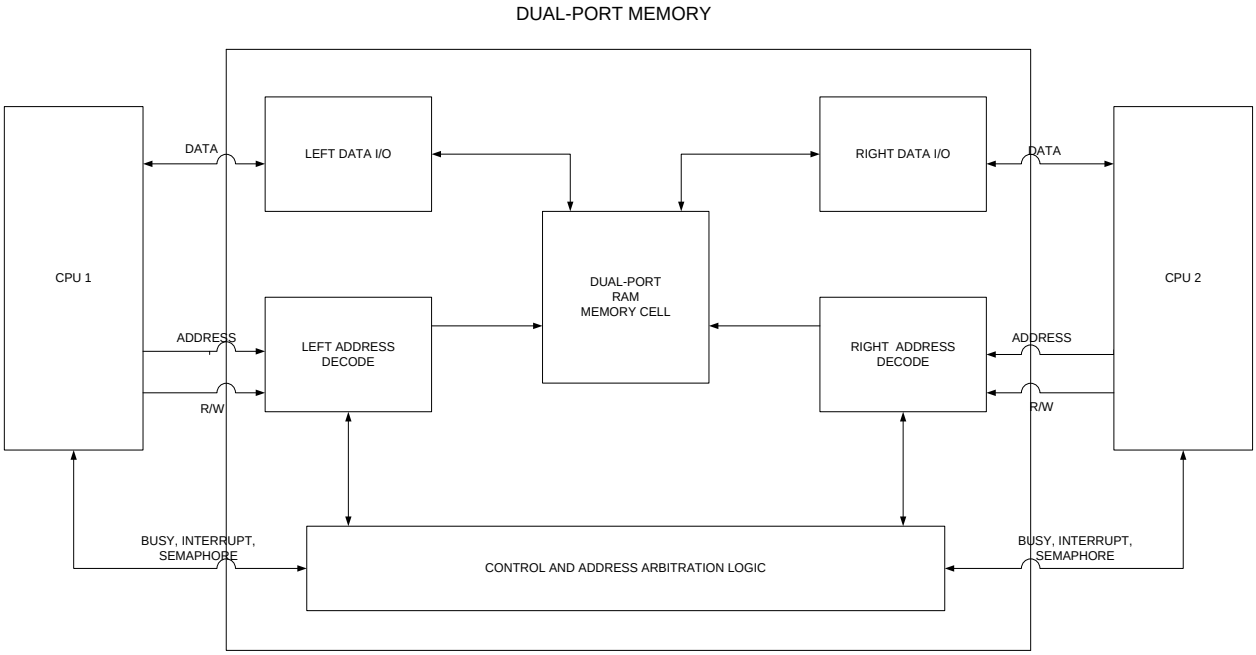
Right Port Configuration			
Bus Match	**Bus Size Select**	**Configuration**	**I/O Pins Used**
0	0	× 36 (standard)	I/O_{0-35}
0	1	× 36 (CE active SEM mode)	I/O_{0-35}
1	0	× 18	I/O_{0-17}
1	1	× 9	I/O_{0-8}

Right Port Operation				
Configuration	**Word Address**	**Byte Address**	**Data Accessed**	**I/O Pins Used**
× 36	X	X	DQ_{0-35}	I/O_{0-35}
× 18	0	X	DQ_{0-17}	I/O_{0-17}
× 18	1	X	DQ_{18-35}	I/O_{0-17}
× 9	0	0	DQ_{0-8}	I/O_{0-8}
× 9	0	1	DQ_{9-17}	I/O_{0-8}
× 9	1	0	DQ_{18-26}	I/O_{0-8}
× 9	1	1	DQ_{27-35}	I/O_{0-8}

Asynchronous Dual-Port RAM Applications

Dual-port memories are used mainly for independent communication between two processors. Cypress asynchronous dual-port memory provides a common memory accessible to both processors (see Figure 11), which can be used to share and transmit data and also to keep track of system status.

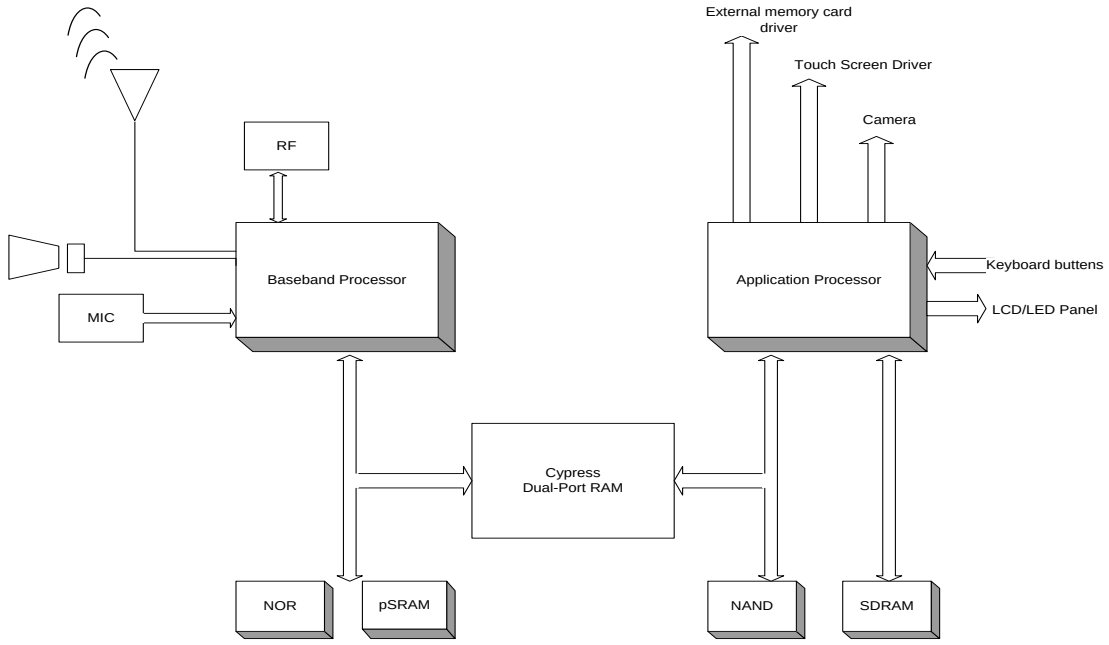
Figure 11. Processor Communication with Dual-Port Shared Memory



Dual-port RAMs are used in set-top boxes in which the board includes a CPU in addition to other peripheral components, such as hard drives, Internet interfaces, and infrared send-and-receive components. The main function is to transmit and store digital television programs. The board can also connect to the Internet through infrared components to control the tuner front-end using an I^2C interface and to complete data exchange through the dual-port RAM.

You can also connect dual-port RAMs seamlessly to any of the popular baseband or application processors in smartphone devices (see Figure 12). A typical smartphone incorporates various multimedia functions and faster data transfer. The baseband processor, usually DSP, handles voice processing while the application processor, typically a RISC processor, manages MPEG video compression and decompression. The communications link between these two processors, which share data, is a common bottleneck. Thus, you can achieve optimal performance only with shared memory.

Figure 12. Use of Dual-Port RAM in Mobile Devices



References

1. Dijkstra, E. W., “Solution of a Problem in Concurrent Programming Control.” CACM, Vol. 8, No. 9, Sept. 1965, p. 569.
2. Dijkstra, E. W., “Co-operating Sequential Processes.” Programming Languages, F. Genyus (Ed.) Academic Press, New York, 1968, pp. 43−112.

Summary

Cypress asynchronous dual-port RAMs offer the reliable, high-speed data sharing that is required between two devices, ensuring compatibility with networking components, videoconferencing devices, graphics accelerators, and other devices. Dual-port RAMs are available at multiple operating voltages (5 V, 3.3 V) and have the following features: mailbox, arbitration, semaphores, data-width expansion, and bus matching. For complete information on Cypress asynchronous dual-port RAMs, visit our website at www.cypress.com.

About the Author

Name:	Sheetal Chennapnoor
Title:	Applications Engineer
Contact:	smch@cypress.com

Document History

Document Title: AN70118 — Understanding Asynchronous Dual-Port RAMs

Document Number: 001-70118

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3828464	SMCH	07/19/2013	Obtain spec. # for note to be added to spec. system.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com