AN1210

A Protocol Specific Memory for Burstable Fast Cache Memory Applications

Prepared by: Ron Hanson

Cache memory design has evolved rapidly in recent years, taking full advantage of the specialized cache application specific fast static RAMs that are becoming increasingly available. These advanced designs are driven by several factors: faster processor clock rates, larger on-chip processor caches, larger and faster FSRAMs, more efficient processor bus protocols, and more efficient DRAM interfaces.

CACHE MEMORY DESIGN TRENDS

Six key trends can be observed in this evolution:

1. Larger caches to improved hit rates.
2. Faster caches to maintain the desired no-wait state response.
3. Dominance of direct-mapped cache designs over the number of multiple-way set associative cache designs.
4. Minimization of external cache control logic to increase speed.
5. Users are developing their own cache solutions, even though vendors are offering more and more integrated solutions.
6. An increasing use of Application Specific Memories (ASMs).

LARGER CACHES

The latest CISC and RISC processors all have ample amounts of no-wait state cache on-chip or included in the processor chip set. Frequently this cache responds a full clock cycle or more faster than an external memory cache could because it is connected to the processor's highly efficient internal bus. In the case of the MC68040, this is a full Harvard Bus architecture that is at least twice as efficient as the fastest external memory system.

The hit rates of these internal caches are very impressive too. The i486™ provides 8K bytes of on-chip four-way set associative cache as does the '040. Though a small amount of cache, these caches have read hit rates greater than 80%. In short, it takes a comparatively large external cache to improve on the performance of the processor alone and this trend will continue. However, FSRAMS are also getting larger. 256K bit FSRAMs are now in abundance and 1 Megabit FSRAMs are in production. As has always been the case with memories, these new larger FSRAMs will replace the older smaller ones at about the same price relative to their respective product life cycles. In other words, building a cache with the largest FSRAMs available today is no more expensive that building a cache three years ago with the largest FSRAMs available then.

FASTER CACHES

Processor speeds continue to increase and there is no end in sight. There are already 50 MHz production processors. Recently the processors have been designed to be more "cache friendly." Significant protocol improvements were implemented on the '040 versus the '030 and the i486 versus the i386. These include implementing synchronous protocols, adding burst addressing, and reducing the data input set-up times.

However, it still comes down to question of raw speed. Fortunately, the increase in density has also been accompanied by increases in FSRAM speed. Now RAMs with 12 ns access times are available to support the 50 MHz processors. It is increasingly apparent that greater integration will be needed to continue to support the fastest processors. The elimination of logic circuits from the critical cache speed path is being vigorously pursued today.

THE DOMINANCE OF DIRECT-MAPPED CACHE DESIGNS

It has been shown that for any given system, as the size of the external cache increases, the performance advantage of a multiple way set associative cache over a direct mapped cache quickly fades to insignificance.[1] Furthermore, a multiple way set associative cache is always more complex to implement.[2] In a discrete design, this translates to either more cost or a loss in response time, which erodes any performance advantage that might be gained. For an integrated solution, it means relying on a vendor for a purchased proprietary solution. Often, if more performance is sought, it is far simpler and less expensive to just enlarge the cache rather than build in multiple way set associativity.

---

1 Jeff Leonard, "Clever Cache Designs Required to Pace High-Speed RISCs," *EE Times*, March 19, 1990, pp. 56, 68 – 69.

2 Mark D. Hill, "A Case for Direct-Mapped Caches," *IEEE*, December 1988, pp. 25 – 40.

MINIMIZATION OF EXTERNAL LOGIC

This point differs from the comment made on the elimination of logic circuits though integration. The Cache Tag RAM is a good example of integration that eliminated the need for a discrete comparator logic device. This did not minimize the logic required. Synchronous or self timed RAMs accomplish this by greatly reducing the complex logic required during write cycles. This is only the beginning; new protocol-specific memories are on the way that will take their cues from the processor itself and perform the needed RAM functions.

USERS ARE DEVELOPING THEIR OWN SOLUTIONS

There are many reasons why computer companies from the lowest performance to the highest are developing their own circuits rather than purchasing the ready-made solutions. One is competitive pressures. PC manufacturers using the same processor, coprocessor, mass storage devices, etc., must find a way to differentiate their products. They can do this by designing their own circuits. Another reason is value added. Many of these companies desire to develop their own chip technology to increase their own share of the revenue received for each computer.

Nevertheless, there is still a high demand for standardized memories. The sheer volume a memory can generate if it is adopted as a standard will drive its cost down far below what an individual custom memory could accomplish. Thus, though cache designs are using more specialty ICs, they still rely on multi-sourced high volume memories for cache data storage.

USE OF APPLICATION SPECIFIC MEMORIES

Referring back to the problem of supporting the very fastest processors, it is clear that the cache designer must attack this problem on all fronts. What is needed is a smart flexible, integrated, high density, very fast SRAM. Such products do exist, and the following is a description of one of the latest under development by several vendors that combines all of these features.

THE SYNCHRONOUS BURST PROTOCOL

In an effort to overcome the limitations of memory bus bandwidth, many of the high performance microprocessors have implemented burst memory protocols. Rather than transferring a single memory word per bus cycle, the microprocessor will transfer (burst) several consecutive memory words in quick succession. The number of words transferred corresponds to the length of a line in the microprocessor's internal cache. Burst transfers have been shown to greatly improve bus utilization. The MC68030, MC68040, PowerPC™, i486, Pentium™, MC88200, and AM29000 all employ burst memory transfers of one type or another.

Though the on-chip cache(s) can be very effective, system performance frequently can be improved by the addition of a secondary cache memory external to the microprocessor. There are three good possible reasons to add a secondary cache: 1) in multiprocessing systems, the time spent arbitrating for control of a global bus can severely degrade performance; 2) the system bus may run at a significantly slower rate than the microprocessor bus; and 3) the nature of the code itself may be better suited for larger caches than are available on-chip.

Burst protocols provide a new challenge for system designers. To achieve no wait state performance, it is necessary for the cache to count through the burst sequence. This in turn creates a problem during cache update cycles when wait states must be added to account for slower DRAM access times. Clearly, the designer would benefit from the integration of as much of this logic as possible onto the FSRAM. This reduces chip count and eliminates the propagation delay from discrete devices. Furthermore, by using inputs directly from the processor, it is possible to actually minimize the amount of logic required to manage the burst cycle. The inclusion of this logic creates an FSRAM that is not only processor specific, but protocol specific as well.

THE 32K x 9 SYNCHRONOUS BURST FSRAM

Not surprisingly, the original specification proposal for this burst FSRAM came from a user, Compaq Computer (Houston, Texas). It is a Synchronous FSRAM with an on-chip burst counter (see Figure 1) and special logic that enables the RAM to interface directly to the i486 processor as well as a cache controller. This device is being developed by several vendors for the i486 market.

The device is similar to existing synchronous FSRAMs in the market today. All of the address and control signal inputs to the RAM are held in registers on the chip, which are triggered by the rising edge the clock input (K) or the clock input gated by another input signal. These other signals include the $\overline{\text{ADSP}}$ and $\overline{\text{ADSC}}$ signals that qualify the address input.

The burst counter on chip is designed to count in the sequence used by the i486; however, the on chip count avoids the wait state inserted by the i486 at the beginning of a burst read cycle, thus improving cache performance. The $\overline{\text{ADV}}$ signal advances the counter of the rising edge of the clock, prior to the next memory access. The device uses a data input register to clock in the data on write cycles. Writes to the RAM are self-timed, requiring the minimal amount of control logic.

This FSRAM has a special built in wait state on write cycles (see Figure 2). This conforms with the i486 write timing. Furthermore, the RAM only advances its internal counter when told to by the controller, which is simultaneously acknowledging the previous transfer to the processor. The RAM can insert wait states whenever needed and, more importantly, it can hold address and count and switch from read to write mode in the event that a cache read miss occurs.

The real value of the BurstRAM™ is its simple processor interface (see Figures 3 and 4). The on-chip Address Register is controlled by the clock input and the processor's valid address signal. Thus, the RAM only registers the address when told to by the processor.

Using inputs from users on Motorola's MC68040 microprocessor, a similar device for '040 has been developed. This version, the MCM62940A, can also interface with the MPC601 (PowerPC), MC88200 and AM29000 RISC processors.

This version of the BurstRAM naturally has a modulo four burst counter to stay in step with the '040 and MPC601. No-wait state Write Burst Cycles at very high clock rates are attainable on both '040 and PowerPC platforms.

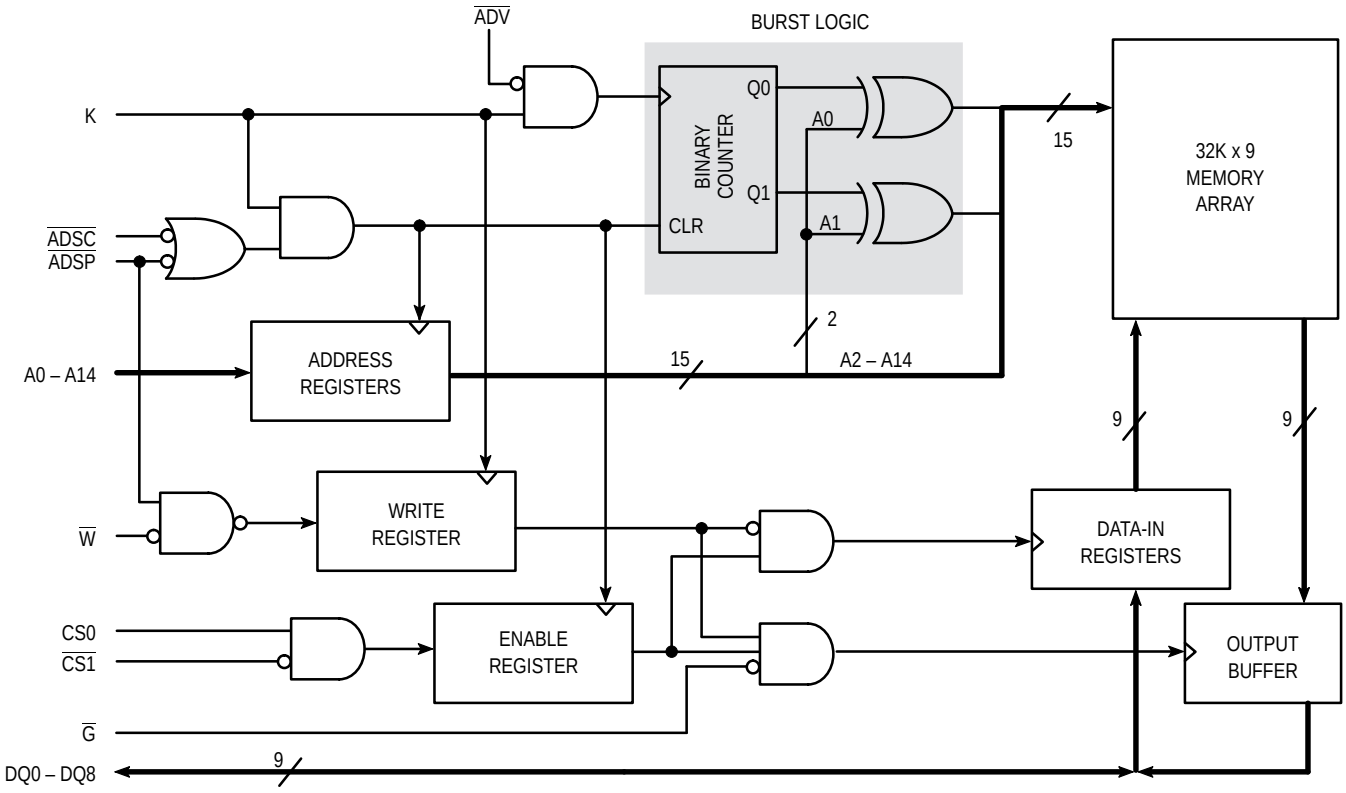

Figure 1. MCM62486A 32K x 9 BurstRAM Block Diagram

The removal of the wait state from the beginning of the write cycle actually simplifies the control logic since the conditions under which the BurstRAMs internal counter is advanced are now identical for both read and write cycles.

The conditional registering of the address input is especially useful when interfacing to a processor with multiplexed address and data buses such as the MC88200 or shared address buses such as the AM29000.

Burst FSRAMs are not a new concept; when the ’030 first introduced the burst protocol in a microprocessor environment, a burst protocol FSRAM specification was developed. Unfortunately, the timing constraints of the ’030 placed the performance goals of the FSRAM beyond the technology available at the time. The only way to build a no-wait state cache at the higher speeds was to utilize the bus retry cycle to rerun any memory access in the event of a cache miss.[3] To the RAM, this meant having to count backwards in the event of a cache miss and adding pins and logic to control this adjustment. Furthermore, the 15 ns access times needed were not feasible at the time.

THE FUTURE DIRECTION OF PROTOCOL SPECIFIC FSRAMs

Clearly, with the technology being developed today, it will be quite feasible to fully integrate all of the elements of the cache (data storage, address tag storage, and control logic) onto one chip. This will be the least cost approach, and if offered by a vendor, it will represent the least amount of user design resources. However, this approach will severely limit cache options and product differentiation. Furthermore, this approach will never perform as well as on-chip caches, which are growing in size. Thus, discrete FSRAMs of some kind will continue to be used in cache memory design.

Protocol Specific FSRAMs will increase in usage, but they will not completely replace standard products if for no other reasons than the versatility advantage of a standard device and its smaller packages. The densities of both will have to increase, though it appears that wider RAMs will be preferred for the new designs.

---

[3] Richard Crisp, Brian Branson, and Ron Hanson, “Designing a Cache for a Fast Processor,” *Electronic Design*, October 13, 1988, pp. 111 – 118.

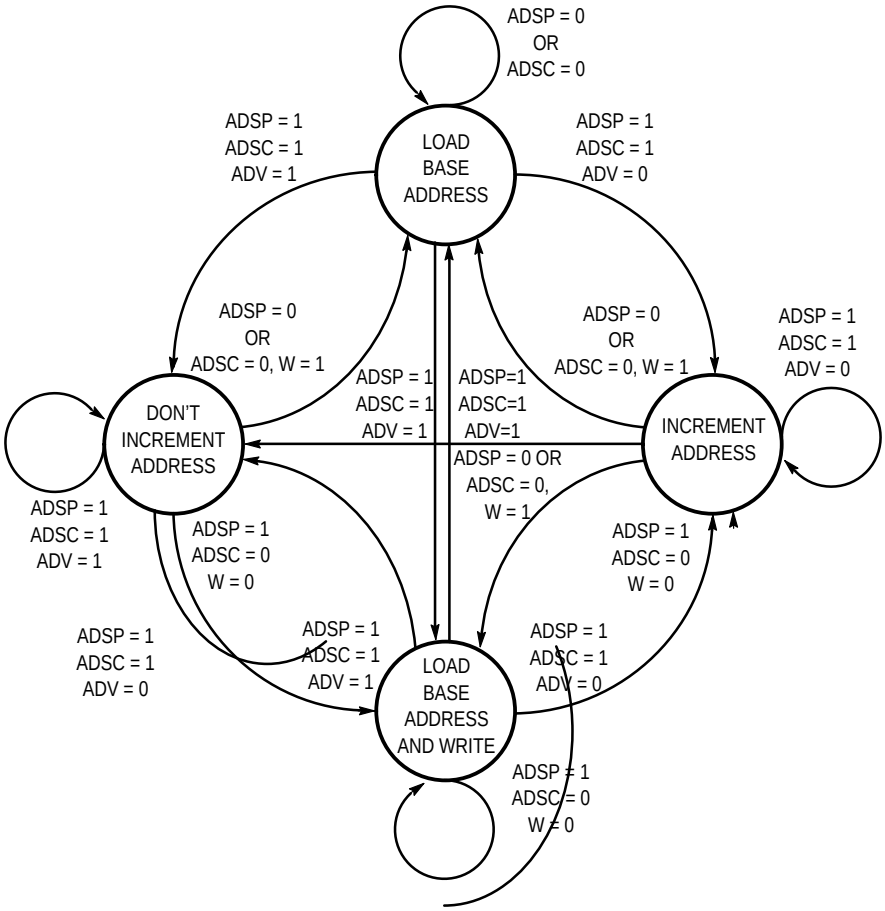

Figure 2. State Diagram for Address Determination on the MCM62486A BurstRAM

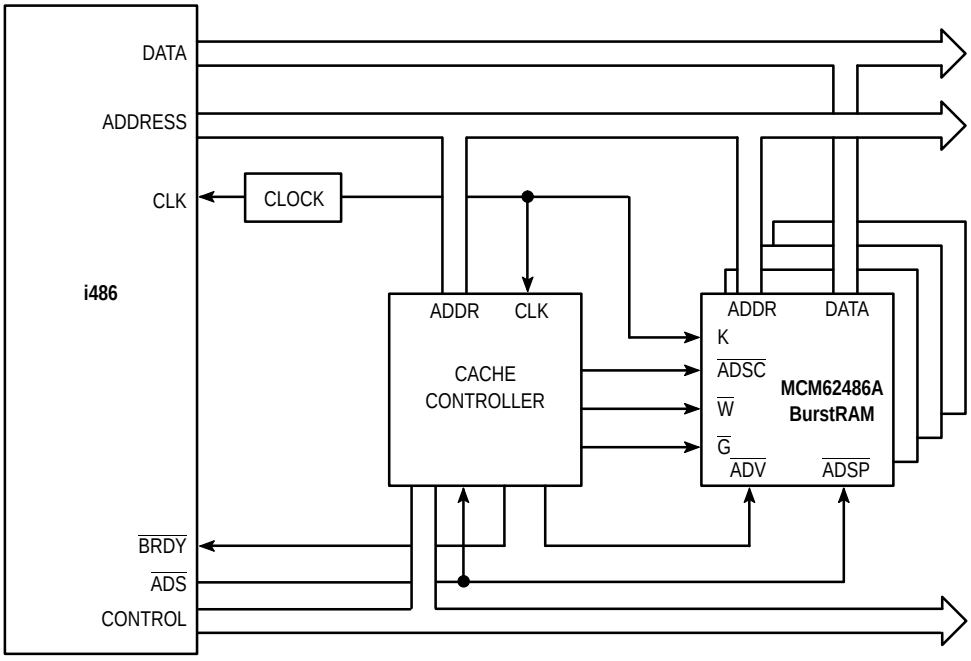

Figure 3. i486 128K Byte Burstable Cache Memory Block Diagram

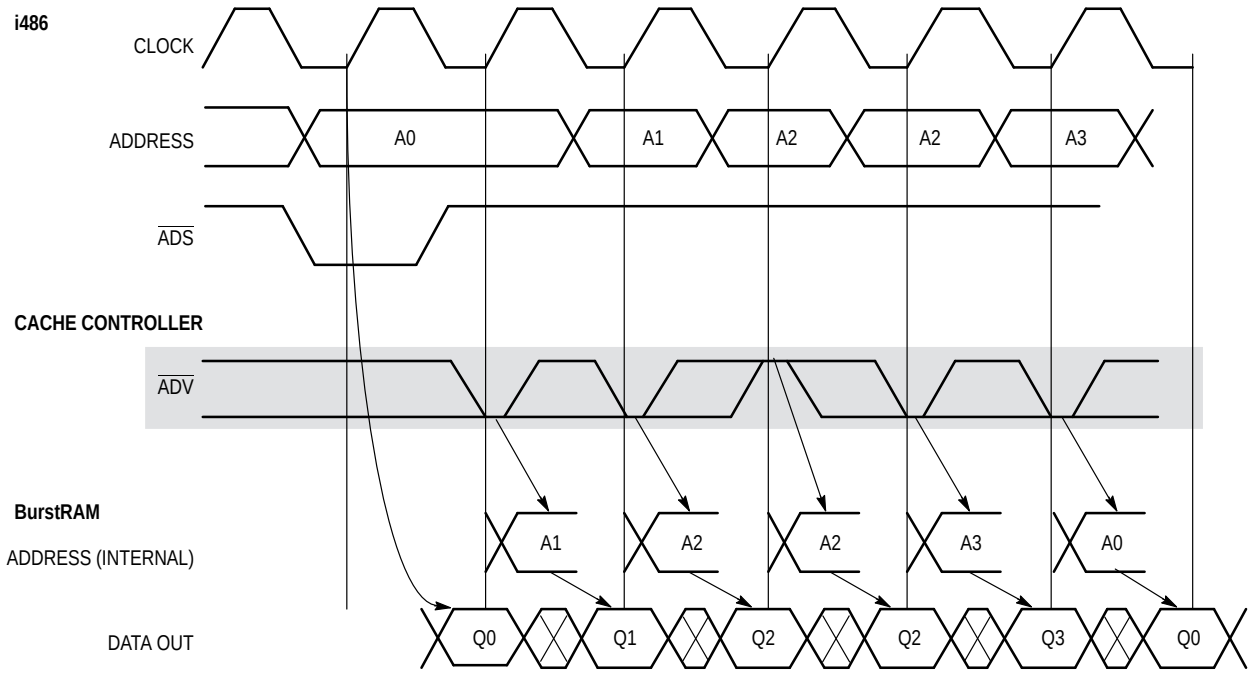

Figure 4. Timing Example of a 2/1/2/1 Burst Read

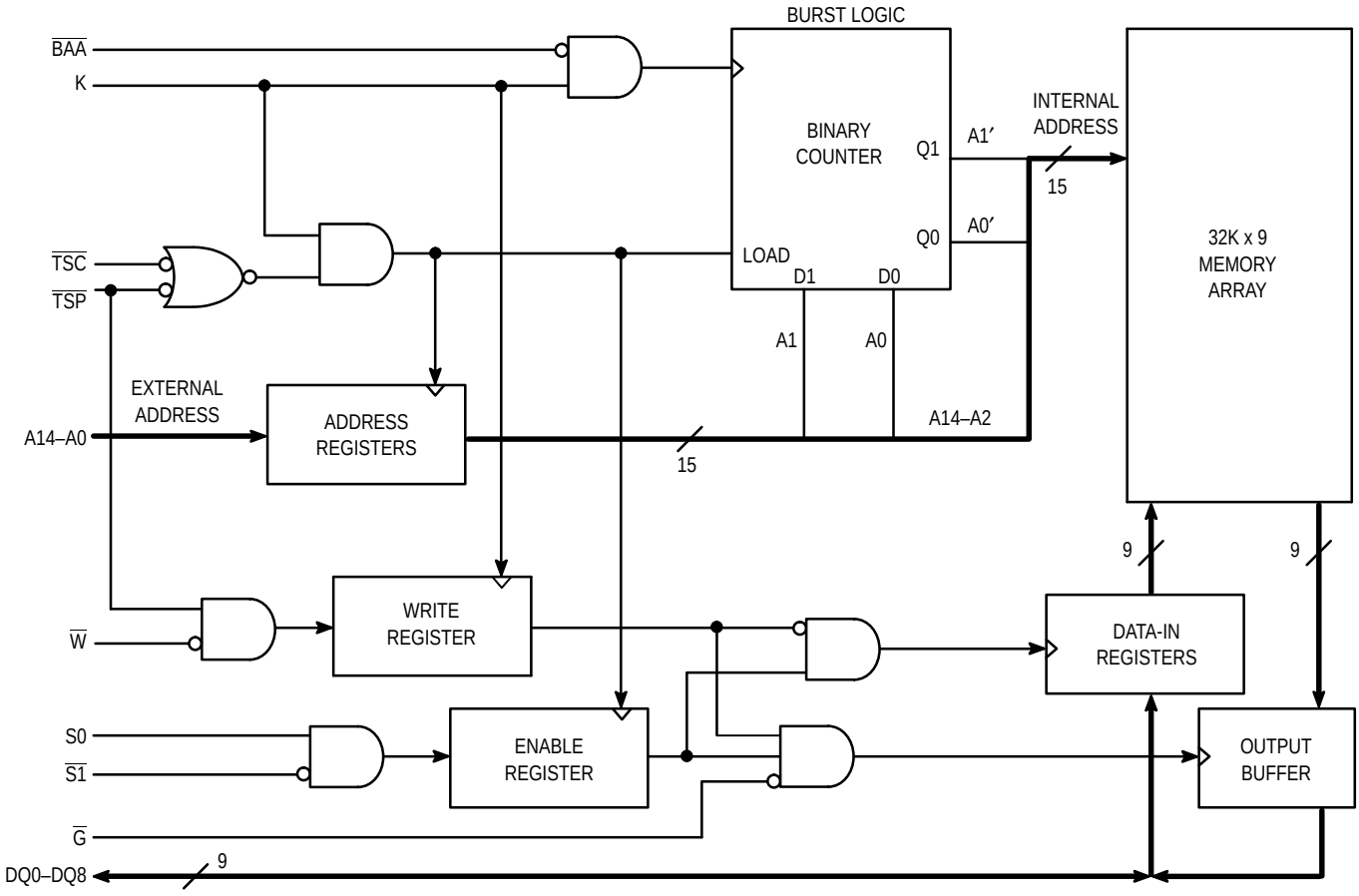

Figure 5. MCM62940A 32K x 9 BurstRAM Block Diagram

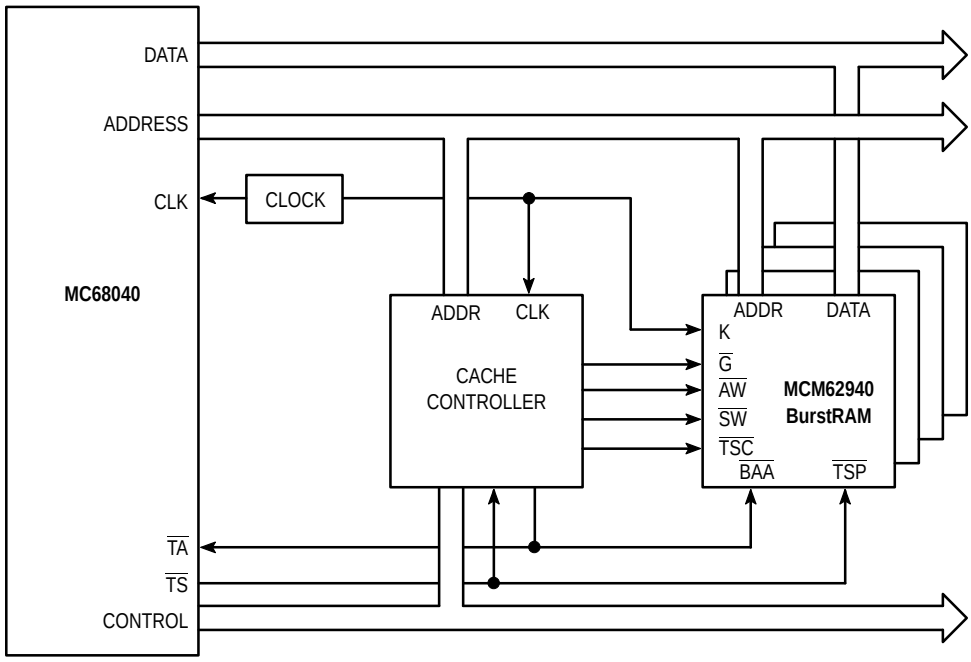

Figure 6. MC68040 128K Byte Burstable Cache Memory Block Diagram

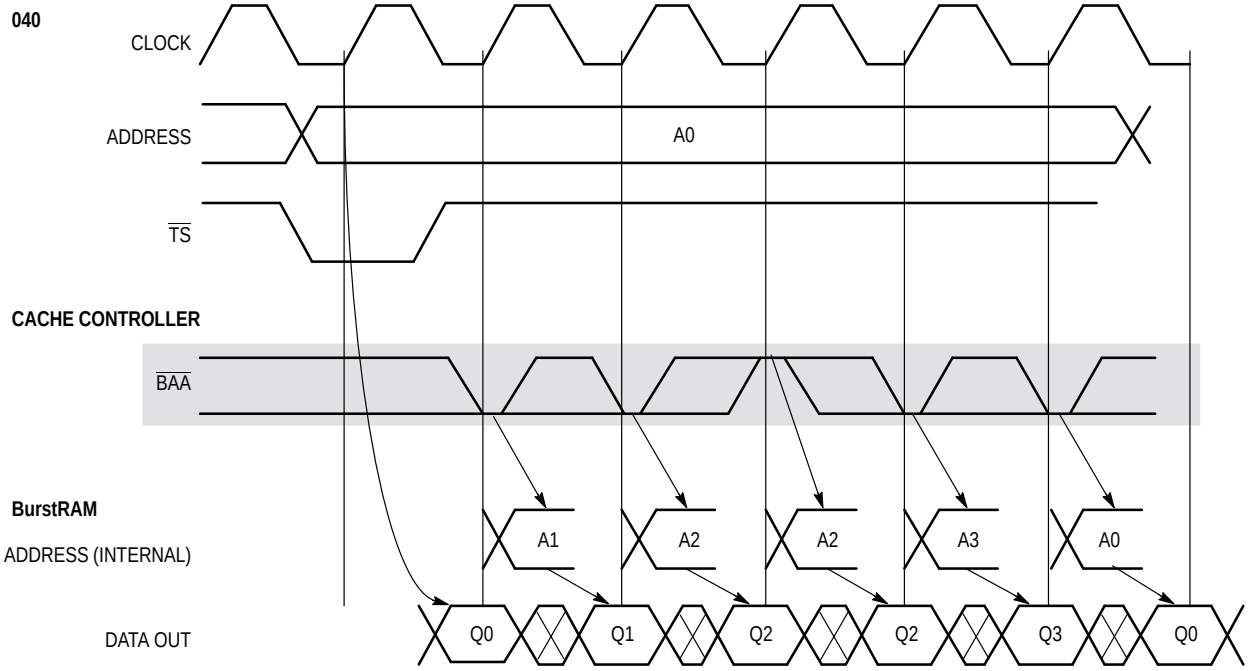

Figure 7. Timing Example of a 2/1/2/1 Burst Read

Literature Distribution Centers:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Centre; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.