

Second Level Cache Applications of National's BiCMOS SRAMs

National Semiconductor
Application Note 713
Charles Hochstedler
Greg Komoto
August 1990



Second Level Cache Applications of National's BiCMOS SRAMs

Multi-level cache architectures are not simply architectures with multiple caches. Understanding the distinction between multiple caches in a single level cache architecture and a multi-level cache architecture requires that we first consider the complete memory system hierarchy, and then define levels within that hierarchy.

Register files are the closest to the CPU (most often imbedded within the CPU) and may be considered the "zero" level in the memory hierarchy. If a needed datum or instruction is not available in the register file at a given instant, the system looks to see if it is available from the next level of the memory hierarchy. A cache is most often the first level of this hierarchy. In a simple system the main memory may be the second level, with disk as the third and removable magnetic store (e.g. tape) as the fourth level. *Figure 1* shows the typical computer system memory hierarchy with a single level cache architecture. Note that *Figure 1* illustrates a single level cache architecture, even though there are separate data and instruction caches.

Notice that the various levels of the memory hierarchy are chosen to span a range of density, cost per bit, and performance (access time). Each level of the hierarchy is faster than the next higher level. Each level is smaller in density and more expensive on a per bit basis than the next higher level. Table I provides an indication of the speed and density ranges which are commonly used today at various levels of the memory hierarchy.

The purpose of a hierarchical memory system is to provide overall performance (access time) close to the performance of the lowest level, while providing the high density storage needed by the machine. In addition, the cost per bit must approach the cost per bit of the highest level in the hierarchy. The information in one level of the hierarchy is usually a copy of information in another level, or is new information likely to be copied into another level soon. The lower levels are included in the architecture primarily for performance—with bandwidth at or near the processor bandwidth. The higher levels are included to provide significant storage capacity at a reasonable cost per bit, but at the expense of a significantly reduced bandwidth.

Second Level Caches

Multi-level cache architectures are system architectures which split the cache memory subsystem into functional blocks residing at more than one level in the memory hierarchy. *Figure 2* illustrates two such architectures. The first is a uniprocessor system, with separate data and instruction caches at the first level, and a single large second level cache. The second architecture illustrates a multiprocessor system, with private caches for each processor, and a large shared second level cache. If you study *Figures 1* and *2* closely, they should help to clarify the difference between multiple caches at one level, and multi-level caches.

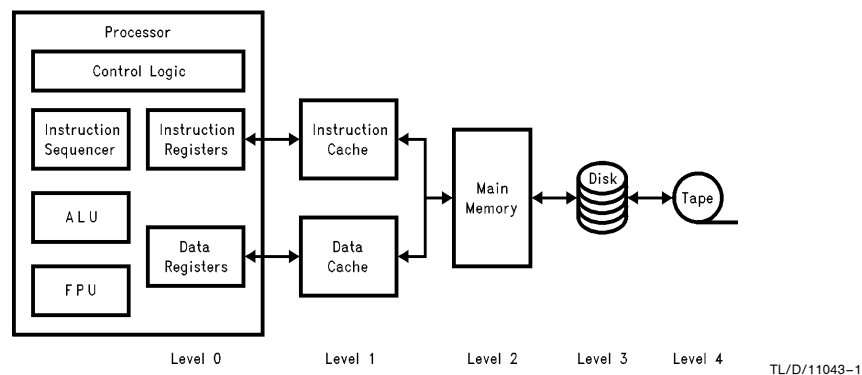
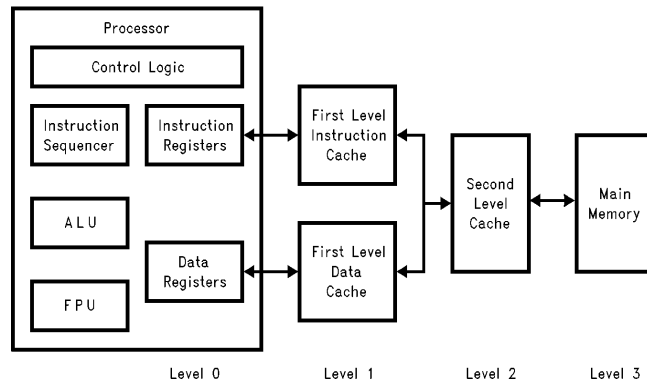


FIGURE 1. Memory System Hierarchy

Table I. Speeds and Densities Typical Over the Memory Hierarchy

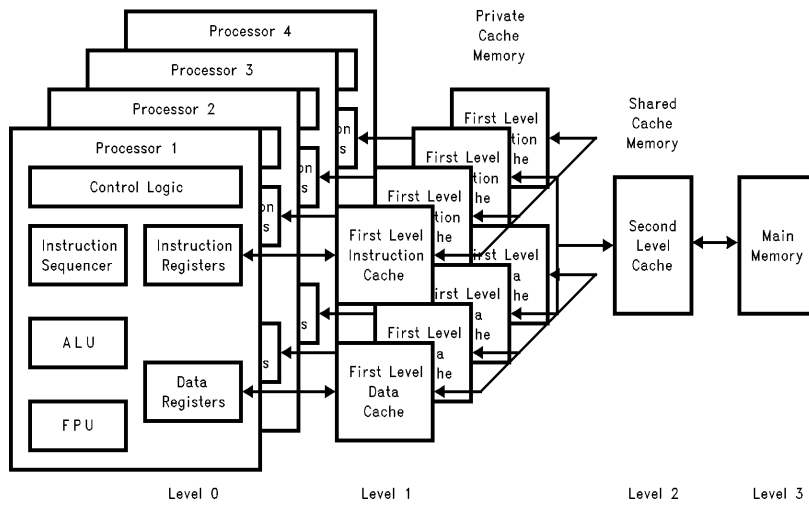
Application	Level	Access Time	Density
Register Files	0	3–15 ns	64–2k Bytes
Caches	1	5–35 ns	2k–256k Bytes
Main Memory	2	25–200 ns	1M–1G Byte
Disk	3	10–100 ms	100M–100G Byte
Tape	3	Minutes to Load	As Many as Needed

AN-713



TL/D/11043-2

FIGURE 2a. Uni-Processor with Multi-Level Cache



TL/D/11043-3

FIGURE 2b. Multi-Processor with Multi-Level Cache

Benefits of Multi-Level Cache Architectures

There is only one fundamental reason for including a second level cache in any given system architecture—system performance enhancement. While there are a few other specific additional benefits, these benefits are really just specific facets of the system performance benefit. For example data coherency in multiprocessor systems may be enhanced by adding a second level cache of shared memory. Yet these benefits also bring along some costs. Second level cache sub-systems cost more money, increase design time, increase time-to-market time, increase power, increase component count, lower system reliability, increase cooling system requirements, and so on. In spite of the costs, the second level cache is growing in popularity. The market's insatiable appetite for increased performance and improved cost-performance ratio has forced system architects to consider and implement such architectures.

One reason for the increased use of second level cache architectures is the recently improved density of very fast SRAMs and the rapidly reducing cost per bit of these devices. National's very-fast, high-density BiCMOS SRAMs are leading the way in changing the performance and cost per bit ratios that are sparking these trends in the industry.

Another force is feeding this increased use of second level cache architectures. The performance levels of processors, both microprocessors and ASIC-based custom processors, are rapidly advancing. The access times of the DRAMs most often used in main memory have not kept pace with advancing processor speeds. In addition, the impact of RISC processing techniques on the bandwidth of processors, and the resulting demand for higher bandwidth memory systems has forced the designers to look for some solution.

The computer market pays a premium for higher performance. This premium is enough to motivate the exploration of any architecture likely to improve performance and especially an architecture likely to improve the system cost-performance ratio. System architects are finding that second level caches do improve performance and cost performance ratios for many high end systems.

SECOND-LEVEL CACHE APPLICATIONS

When the system designer's goals are to build a new system architecture, the view is broad and the options are many. Most systems in design are restricted somewhat by requirements to maintain some measure of compatibility with the company's prior systems. Software compatibility is a common goal. Another common goal is bus compatibility, which allows the use of plug in options already developed. Systems unencumbered by restrictions of compatibility with prior bus structures or software will usually afford the system architect an opportunity to explore new concepts, and seek new levels of innovation. Without concerns for compatibility, he is free to tailor the system for the target price performance range and explore all feasible options.

Cache Architecture Choices in a New System Design

A second-level cache should be examined as one option. System metrics should be predicted (simulated) for various sizes and architectures of single and multi-level cache architectures. The system architect can compare a wide range of cache architectures, such as:

- Single Level Cache Architectures
 - Direct Mapped Cache
 - Two Way Set Associative Cache
 - Four Way Set Associative Cache
- Multiple Caches
 - Separate Data Cache and Instruction Caches
 - Direct Mapped Data Cache and Two Way Set Associative Instruction Caches
 - Two Way Set Associative Instruction and Data Caches
- Multi-Level Cache Architectures
 - Direct Mapped Second Level Cache with Separate First Level Data and Instruction Caches

Through the comparison of a range of architecture options the system architect may determine the few architectures most likely to best fit his design goals.

The validity of simulation results is largely dependent on the accuracy of the system models and on the applicability of the traces used for the modeling exercise. (Traces are basically lists of the address references generated by an application software program as it runs.) Knowledge of the intended applications of the system being developed is key to selecting appropriate trace material. Careful examination of the validity of modeling assumptions, and careful choice of trace material should yield useful results, indicative of the relative performance of the options being explored.

Unfortunately there are a host of variables to be considered in the study and simulation of the most promising architectures. Overall cache size, cache block (or line) size and replacement algorithms are examples of cache architecture tradeoffs in addition to the choices of the level of associativity and the number of cache sectors.

Another level of tradeoffs to be considered is the range of suitable devices available for the physical implementation. Performance, size (density) and cost of the memory devices must be considered. How much faster is fast enough for the best choice to be a smaller and faster SRAM? How much more cost per bit, for a higher speed device, still results in a better system cost-performance ratio? How will the component cost and performance change over the production life of the system?

The astute system architects and designers consider the cost performance ratio of candidate devices, forecast over the high volume years of the system life cycle. This sometimes causes the early development and part of the system debugging to be done with an earlier generation multiple sectors, overall cache size, block size, replacement policies,

and more. The right way to decide is to simulate several options and use the results to indicate the preferred implementation for the particular machine. Most systems in development today with second level caches are opting for large sizes, 1 to 4 Megabytes, and a straightforward direct mapped implementation. This is generally large enough to contain frequently called portions of the operating system, as well as large segments of application code and data. The second level caches going into new multiprocessing systems tend toward the higher densities, as may be anticipated.

SRAM CHARACTERISTICS FOR SECOND LEVEL CACHE APPLICATIONS

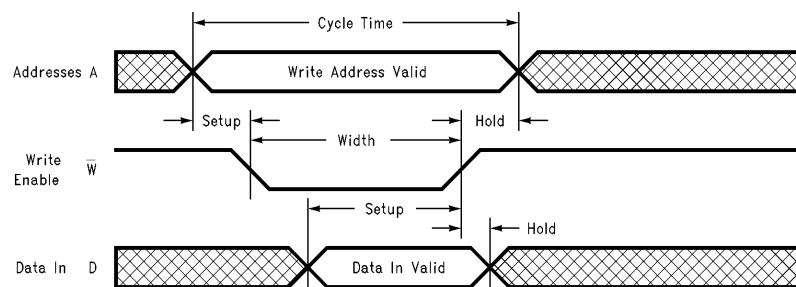
There are many characteristics which may help to make a given SRAM better suited for cache applications. Very fast access time certainly is the first to come to mind. Other attributes are also becoming critical.

Many SRAM vendors are becoming increasingly aware of the potential for switching noise problems in arrays of very fast SRAMs. Recently, JEDEC has approved a new family of SRAM pinouts intended to ease this problem by providing multiple power and ground pins in the center of the package. These new pinouts, called the "revolutionary" pinouts, have been approved for devices from 256k to 4 Megabit densities, in bit wide, 4-bit wide, and 8-bit wide organizations, with common and separate data I/O, in synchronous and asynchronous versions, and with TTL I/O and ECL I/O.

As high speed SRAM vendors migrate to this new family of pinouts, some of the noise problems will be alleviated.

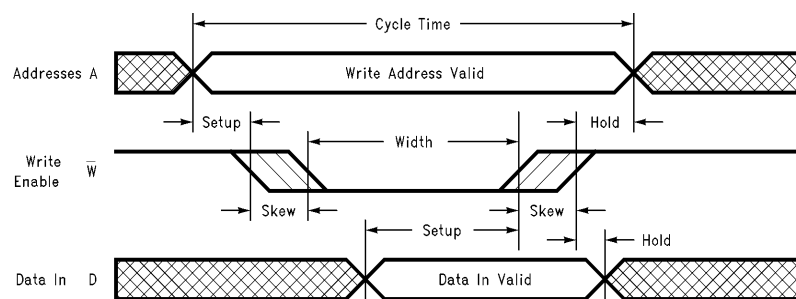
The primary characteristic needed is clearly speed. However, the important speed is the speed realized in the system, not the speed that the SRAM vendor claims on the data sheet. System variables will dictate how close the speed in the cache is to the data sheet speed. There are, however, some key parameters which often cause the system to run slower than the potential. Awareness of these potentially speed degrading problems at the time of SRAM selection may help considerably in system performance.

One good example is seen in the way that many SRAM vendors specify the write cycle timing. It is common to find the write pulse width specified at a value equal to or almost as wide as the cycle time specification. The sum of write pulse width, and the longest setup and hold times is the best possible write cycle time. System timing realities add skews to the SRAM timing, lengthening the cycle time in the system, and degrading bandwidth. In the write cycle example, in order to meet the minimum write pulse width under all conditions, the design will result in the nominal write pulse width being wider, and widest possible write pulse wider still. Since setup and hold times are referenced to the write pulse, the uncertainty, or skew in the write timing adds directly to the write cycle time. Degradation of the write cycle is illustrated in the timing diagrams of Figure 3.



TL/D/11043-4

3a. Data Sheet Write Timing



TL/D/11043-5

3b. System Write Timing

FIGURE 3. SRAM Write Timing Skews

memory are straightforward options. It is conceivable to offer main memory controller on each memory card which can support interleaving in main memory for the higher end systems. In the lower end systems, the ASIC memory controller could be simplified to reduce or eliminate the interleaving. Across all systems, the memory controller could provide other memory functions such as: parity or SECDED (error correction), diagnostics, error logging, DMA support, and refresh control (assuming DRAM). The economic benefits of commonality throughout the family of machines may bring economies of scale which help to offset the cost of some performance imbalances within any one given machine in the family. It is also practical to sell the machine at a slightly higher price, in effect charging the customer for the benefit of an easy future upgrade path.

Second level cache and higher speed CPU can be a practical solution to the very real problem of extending a system architecture into higher performance models, beyond the capabilities of the system backplane bus structure. If the majority of system applications are compute bound (not I/O bound) this may be an economical and practical upgrade path. One benefit is that a line of peripheral and I/O cards need not be redesigned to support the new faster model. Depending on the machine architecture, the upgrade to a higher speed CPU with second level cache on the card may be relatively straightforward.

With this in mind, National offers a full line of asynchronous and synchronous BiCMOS SRAM products. National's product line grows smoothly to fit your needs. For example, our 16k x 4 and 64k x 4 ECL I/O SRAMs come in pin for pin compatible replacement flatpak packages. National offers an array of densities and organizations, from 18k to 1 Megabit, in an array of x1, x2, x4 and x9 organizations.

SECOND LEVEL CACHES IN MULTIPROCESSOR SYSTEMS

Second level caches may bring an additional and significant benefit to multiprocessor systems. In multiprocessor systems there is a strong desire to provide each processor with a private cache, for high bandwidth to and from memory. However, cache coherency can become a major design challenge.

If each processor has a private cache, as was shown in *Figure 2*, there exists the very real problem of how to manage the eventuality of any specific datum being cached in more than one cache at the same time. More to the point, a problem arises when one processor writes to a datum in its private cache, when that datum was also cached in another processor's private cache. In this case, the second processor's cache now contains "stale" (invalid) data.

For proper system operation, this condition must be handled by a set of protocols. These protocols are usually implemented by the cache controller and memory management hardware design.

There are a variety of cache coherency protocols being used today. A general case, the MOESI model, is useful for discussion. The MOESI model derives its acronym from the set of possible states attributed to each cached datum. These states are: Modified, Owned, Exclusive, Shared, Invalid. The cache controller can be designed to keep state information for each cache block. Even caches in uniprocessor systems needed to implement at least a valid/invalid state bit. With multiprocessors, additional state information is needed for proper control. Combinations of these states make it possible to determine the proper action under any condition.

A datum may be owned and not shared when it is first fetched from main memory by a given processor and cache. Later, if it is requested by another cache, the first processor's cache controller needs to change the status to owned and shared. The second processor's cache sets the status as unowned and shared.

There is more than one way to handle the problem of writing to a shared cache block. The cache write scheme is involved, also. If a write through protocol is chosen, the act of writing the new data back to main memory can be observed by all the cache controllers, and they can correct their copy if they have the same cache block. For reduced bus traffic it is desirable to implement a copy back cache scheme, where the cache block is updated only in the cache, and copied back to main memory only when the block is flushed from the cache. Thus, writing to an owned and unshared cache block is not a problem; it simply becomes owned, unshared and modified. The modified state indicates a need to copy it back when it is flushed.

In the case of a copy back scheme, writing to a shared line may be handled by one of two protocols. The address may be broadcast by the owner, to flag the sharers to invalidate their copy of that cache block. In this scheme the owner must supply the block whenever any processor/cache requests it, since it has the only up to date copy. Alternatively, the owner may broadcast the address and the newly changed byte(s), allowing the sharers to update their copy. Stepping back from the implementation details of the protocols, the system architect can see the need for high bandwidth from cache to cache, in addition to cache to main memory. As the number of processors grows, the bandwidth demand increases. A second level cache can be very helpful in reducing the demands on bandwidth to main memory; providing high bandwidth from a large shared second level cache. For example, using only 18 very fast 1 Megabit SRAMs organized 256k x 4, a 2 Megabyte second level cache of 256k x 72 could be conveniently implemented. An ASIC cache controller and some bus buffers/drivers would complete the majority of the required components. Bandwidth in the neighborhood of 300 Megabytes per second should be achievable with a 72-bit datapath and 15 ns SRAMs.

A rigorous designer will carefully consider the range of cache variables for the second level cache architecture: direct mapped versus set associative, with single or memory device available at that time. Final system development and production volume ramp-up can occur with the desired (new) generation device. The device which was not available at the start of the system development, may provide the best cost-performance ratio when the system production is reaching full volume.

Yet another level of system complexity tradeoff needs to be considered. The cost and performance tradeoff between increasing cache sub-system performance and increasing main memory performance should be explored. Interleaved main memory may be effectively applied to improve memory bandwidth. Overall system performance may or may not be enhanced as much as putting the same cost into a larger cache, or a second level cache. Vector machines, as one example, may find that deep vector registers yield cost-performance dividends superior to that of a larger second level cache.

Many variables must be considered in the definition of a new system architecture. Increasing performance almost always means increasing cost. The ideal machine is a careful balance of CPU performance, the memory system perform-

ance, and bus performance, all consistent with one another. Higher performance in any one area of the machine generally increases the machine performance only slightly and decreases the cost-performance ratio.

With these factors in mind, National offers a wide range of very-fast, high-speed BiCMOS SRAMs in a variety of densities and speed grades. Plus, National offers enhanced memories which help ease the design, implementation, and performance of cache sub-systems. National's BiCMOS SRAM product line continues to grow and expand to meet your future memory system needs. Together, they form a full line of BiCMOS SRAMs designed to meet your needs now and in the future.

Second Level Caches in an Upgrade of an Existing System

Most systems developed today are upgrades of prior systems. Considerations for system performance enhancement upgrades are typically quite constrained. Maintaining backplane bus compatibility is a common constraint for an upgrade. In this case the system designers should look at the bandwidth limitations of the existing bus structure.

A higher performance CPU will demand higher memory system bandwidth. If the bus to memory is likely to become a significant bottleneck, a large second level cache may be worth considering. Exploring the cost and complexity differences between larger first level caches, and smaller first level caches with a large second level cache is necessary. In either case, the benefit of an improved cache sub-system is a reduction in bandwidth needed between the main memory and the cache(s).

For some systems, the processor and first level cache(s) may be integrated on one or a few very high density ASICs. It's strongly desirable to have the first level cache on the same ASIC as the processor logic, to eliminate wasteful I/O and board crossing delays. However, current technology limitations leave most high performance system architects wishing for larger caches than are practical on the same silicon as the processor. A large second level cache that is external to the ASIC based processor but still on the same CPU card may be an excellent solution.

For example, a 512k byte cache organized 64k x 72 for an ECL RISC workstation can be readily implemented with an ASIC controller and 18 RAMs (National's 64k x 4 BiCMOS ECL I/O SRAMs). Today, National's industry leading 64k x 4 BiCMOS ECL I/O SRAMs have access times as low as 10 ns. In the near future, speed leading 16k x 4 devices can achieve 6 ns to 8 ns access times. Devices in this speed and density range are quite suitable for large second level caches; a cache that supports the smaller first level caches that are integrated with the processor logic. A series of simulations should easily demonstrate whether this type of architecture upgrade is a good choice for a particular system.

Busses, memory, and CPUs may be the primary areas of concern, but there are certainly several other areas requiring some study, and possible upgrade. System power supplies and cooling must be reviewed. As integration levels increase, the total power consumed is generally reduced. However, larger and more complex caches, plus more complex processor logic running at higher speed may result in an increase in power demand. More subtle power supply characteristics may also need to be reviewed. For example, the supply decoupling of a higher speed upgrade may become more critical, due to the higher speed devices and reduced transition times.

The cooling system may require some upgrades, also. Hot spots in the system are almost surely going to change. Changes such as adding impingement air cooling for a hotter CPU card may be needed. Possibly just a fan change is enough. A significant system upgrade requires a recheck of almost everything, and then improvements and redesigns where indicated.

Another real world occurrence is that some systems are planned from the onset to allow a family of machines to be sold from the basic architecture. This approach will tend to disrupt the system performance balance and the cost performance some. Economic performance, however, in terms of return on investment, may be better served by such a family of machines approach. The goal is to scale the CPU and memory system performance in reasonable steps across a range of system performance capabilities.

It is common to maintain a single backplane bus design throughout the product line. It is not easy to implement backplane busses which scale over a range of bandwidth capabilities. However, the bus cost is typically the least hardware intensive and therefore least expensive of the major system blocks which could become the performance limiting bottleneck. An overdesigned bus on a lower end model machine might not be an excessive cost burden for the system.

In contrast, it is quite practical to offer a set of CPU and memory system options which span a range of performance. Several processor speeds and sizes of

It is quite desirable for the cache to support writes at the same bandwidth as reads. It is possible to design the cache controller to extend the write timing, but that is a cumbersome solution. It is more straightforward to design the cache such that a write cycle requires only the same time as a read cycle. Clearly, the read cycle time is desired as fast as practical, and as fast as is needed to meet the machine bandwidth target. An SRAM with a minimum write pulse width which is in the range of 50% to 70% of the cycle time is desirable. It allows some time for realistic system skews without forcing the system cycle time to be much greater than the data sheet cycle time. National's BiCMOS SRAMs offer a write pulse width that leaves about 33% of the cycle time for system timing skews.

Similarly, SRAMs with zero or small setup and hold times are also easier to utilize in fast cache applications. Be careful when looking at setup times; some setups are referenced to the beginning of the write pulse (e.g. address setup), and other setup times are referenced to the end of the write pulse (e.g. select setup). Data setups are specified either way depending on the vendor and the device. National's asynchronous BiCMOS SRAMs feature zero setup and hold times to make them easier to use in fast cache applications.

A vendor specifying zero data setup referenced to the beginning of write may actually require more setup time than one specified for several nanoseconds of setup time referenced to the end of the write pulse. For example, a 12 ns SRAM with a write pulse of 8 ns and data setup of 6 ns referenced to the rising edge of the write pulse, is easier to utilize at high speeds than a 12 ns SRAM with a write pulse of 7 ns and a data setup of 3 ns referenced to the falling edge of the write pulse. In the second case the data is required at least 2 ns earlier, and that is true only in an ideal system. When the write pulse skew is accounted for, the data may be required about 4 ns earlier.

Another write parameter which may become a concern as speeds are pushed ever higher is one which most vendors today do not even specify. The minimum disable time between two write cycles can limit the speed bringing new information into the cache. To understand this parameter, consider the implications of the address setup and hold time specifications. The write line can only drop an address set-up time after the address stabilizes at the new address. Similarly, the write line must rise and address hold time before the address begins to change. This, in effect, stipulates that the write line must be high whenever the address bus is changing.

Indeed, either the write, or the select, or both, must be high whenever the addresses are changing. If both the select and the write line are low while the addresses are changing, several address locations may be corrupted, not just the prior address location and the new address location. The parameter which most vendors do not bother to specify is the minimum length of time for which either the write, or select, or both, must be high. *Figure 4* provides an illustration of this parameter. National provides this information so the cache designer has one more piece of information

which may help him to achieve the highest performance possible by using National's SRAMs.

As system speeds increase, another parameter may become critical. The read cycle time may be limited by the output hold time from address change. Some vendors do not even specify this parameter. For devices with it specified, it is usually a few nanoseconds, minimum. Inexperienced designers may question why this parameter makes much difference. At slower speeds it certainly is not critical. Using 17 ns access SRAMs and trying to meet 40 MHz (25 ns cycle time) is an example where it may become critical. If the vendor specifies output hold from address at 3 ns, the devices can supply valid data for 3 ns longer than one where it is unspecified (must be assumed to be zero). This 3 ns may make a big difference between the SRAM outputs being valid for about 8 ns versus about 5 ns. Those outputs need to, at a minimum, propagate across some board trace and into the cache controller ASIC (or a discrete external register) where they can be registered and subsequently output on the bus in the next cycle. *Figure 5* illustrates the impact of output hold time on the read cycle.

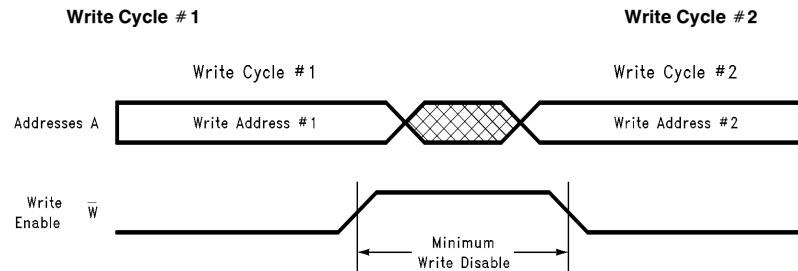
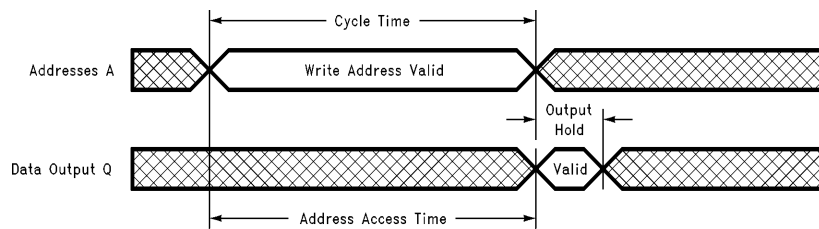


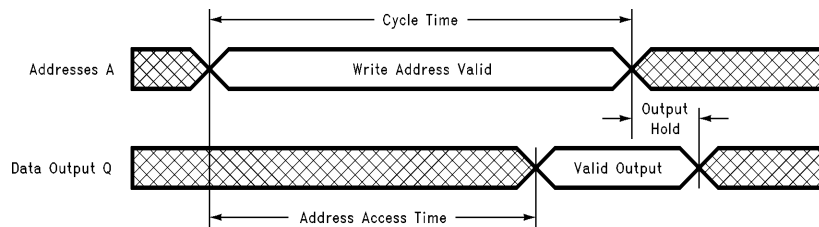
FIGURE 4. Minimum Write to Write Timing

TL/D/11043-6



5a. Data Sheet Read Timing

TL/D/11043-7



5b. System Read Timing

TL/D/11043-8

FIGURE 5. System Output Hold Timing Requirements

In general, the faster the system the more critical these minor parameters become. For tomorrow's systems, fast SRAM access time alone is not nearly enough. Currently, synchronous self-timed SRAMs like National's Advanced Self-Timed SRAMs are rapidly gaining favor for cache applications in very fast ECL systems. They help significantly improve cycle time achieved in the system, as compared to prior generations of synchronous SRAMs.

SUMMARY

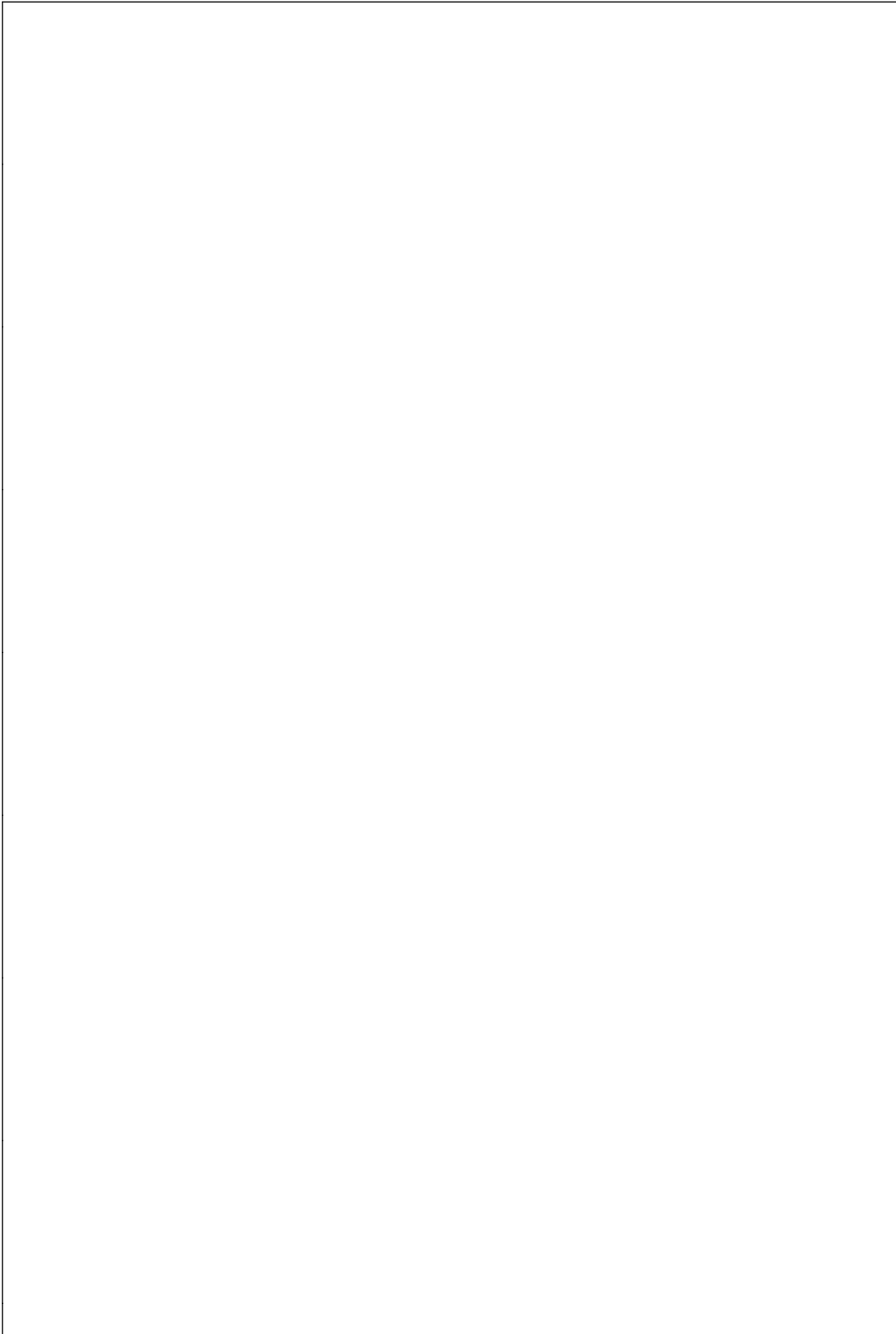
Second level caches are a relatively recent innovation in system architecture. They are used to improve memory system bandwidth in some higher speed systems. They may be appropriate in a wide range of machine architectures; new systems and family upgrades alike. They may help significantly in multiprocessor systems, when main memory bandwidth can not economically provide sufficient bandwidth to meet the demands of all the CPU's combined.

Designing an efficient second level cache is similar to designing any cache. Several architectures should be simulated to determine the most suitable for a given machine and its anticipated set of software applications. Typically, today's systems implementing second level caches are opting

for large and simple architectures; on the order of a few Megabytes of direct mapped cache. The SRAM selected for the second level cache must be fast enough to meet the bandwidth goal. SRAM data sheet speed, however, is not the same as the speed achieved in the application. National's asynchronous BiCMOS SRAMs provide the benefits of a set of specifications which help to ease difficulties the system designers face in implementing very fast caches.

Second level caches are only one architectural technique for improving the bandwidth of the system memory hierarchy. Other options include larger first level caches, interleaved main memory, or simply faster main memory. The apparent rapid growth in popularity of the second level cache indicates that it is a preferred solution with superior cost performance for many system architectures. The second level cache is likely to become a common system feature over the coming years.

Acknowledgement: Much of this material was excerpted from a paper published by C.M. Hochstedler of National Semiconductor—the paper was presented at the SDNC Conference in May, 1990.



LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
19th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.