Revised June 2001

The Memory Gap and the Future of High Performance Memories

by

Maurice V. Wilkes
AT&T Research Laboratories - Cambridge, UK

The first main memories to be used on digital computers were constructed using a technology much slower than that used for the logic circuits, and it was taken for granted that there would be a memory gap. Mercury delay line memories spent a lot of their time waiting for the required word to come round and were very slow indeed. CRT (Williams Tube) memories and the core memories that followed them were much better. By the early 1970s semiconductor memories were beginning to appear. This did not result in memory performance catching up fully with processor performance, although in the 1970s it came close.

It might have expected that from that point memories and processors would scale together, but this did not happen. This was because of significant differences in the DRAM semiconductor technology used for memories compared with the technology used for circuits.

The memory gap makes itself felt when a cache miss occurs and the missing word must be be supplied from main memory. It thus only affects users whose programs do not fit into the L2 cache. As far as a workstation user is concerned, the most noticeable effect of an increased memory gap is to make the observed performance more dependent on the application area than it would otherwise be.

Since 1980, the memory gap has been increasing steadily. During the last ten years, processors have been improving in speed by 60% per annum, whereas DRAM memory access has been improving at barely 10%. It may thus be said that, while the memory gap is not at present posing a major problem, the writing is on the wall.

On an Alpha 21264 667 MHz workstation (XP1000) in 2000, a cache miss cost about 128 clock cycles. This may be compared with the $8-32$ clock cycles in the minicomputer and workstations of 1990 [1]. If the memory latency remains unchanged, the number of cycles of processor idle time is doubled with each doubling of speed of the processor. A factor of four will bring us to about 500 clock cycles.

Hiding the Memory Gap

As the speed of processors increase, the memory gap increases. On the other hand, shrinkage enables L2 caches to increase in size and, to some extent, this balances out the effect of the increased memory gap. However, there are reasons to believe that non-cacheable problems are increasing in importance. On the scientific side there are 3D simulations and similar applications. Again. many database servers used in transaction processing rarely, if ever, manage to establish a working set that will fit entirely within a cache. It must be accepted that, however large the cache memory is made, there will be plenty of problems that defeat it. For such problems the cache actually gets in the way and slows down the running of the program. Readers with a long memory will recall that the CRAY 1, which was designed specifically for such problems, had no data cache.

The use of a cache is only one way of hiding memory latency. Thread changing, is quite effective in servers for hiding the latency due to page faults. At first sight, it would appear appear to have limited scope for hiding the much smaller latency due to the memory gap. on account of the overheads of thread changing. These overheads can largely be avoided if the architecture is based on that of a multiple issue out-of-order processor, modified so that instructions for issue can be selected from a number of active threads, instead of from a single thread. This requires, among other modifications, the provision of multiple program counters. It is then found that the out-of-order execution and register re-naming features of the processor provide a large part of the mechanism required for multithreading. The number of registers must be increased in proportion to the number of threads, and the pipeline must be slightly lengthened to allow for the increased register access time.

The system is known as Simultaneous Multithreading [2, 3]. It has yet to be demonstrated on a working chip, but simulations give hope that good advantage can be taken of multithreading when suitable threads are available. The danger is, of course, that the lengthening of the pipe line might cause the processor to be slowed down appreciable when programs with only one thread are run, but this does not seem to be the case.

Hiding the memory gap provides, at best, a partial solution, and for the long term, we must hope that memory systems, significantly faster than those we have at present, will be developed. Nothing is in sight at the present time and therefore we must press DRAM designers for better DRAMs.

Much delay is incurred in transmitting information from memory chips to the processor chip. However, it should be noted that most of this is associated with multiplexing and takes place on the chips; very little is transmission delay in the wires themselves. For this reason, better physical interconnect systems have a limited role to play. We really require require DRAMs with more pins and wider buses, especially address buses. There is also scope for architectural improvements, for example, the judicious provision of more internal buffering.

Two Industries

It might have been expected that the above developments would come naturally with the striving for better performance. However, this has not happened, and the reason lies partly in a different attitude to product development in the DRAM industry compared with that in the processor industry. It is necessary to understand this differences in order to see why pressure on the DRAM industry is necessary to secure better DRAMs.

The division had its origins in the mid 1980s when the still infant silicon chip industry in the United States was experiencing economic difficulties. These arose from the existence of excess production capacity worldwide, along with heavy overseas competition, in particular from Japan. The consequence was that semiconductor companies in the United States decided to pull out of DRAM manufacture and concentrate on processors, where economic pressures were less severe. However, there are, as I have already indicated, technical reasons why DRAM manufacture should remain distinct from processor manufacture. Not only is the process optimized for high storage density rather than for high transistor quality, but some of the steps in the manufacturing process are different.

Large production volumes are essential to the economy of the semiconductor industry. The processor side of the industry finds its volumes in well established products. These include products for the embedded market. The need to maintain a high volume of production has not prevented processor manufacturers from adopting an innovative policy towards the development of new high performance processors. Indeed, such developments are seen as the source of large volumes in the future.

By contrast, in the DRAM world the emphasis from the beginning has been on maximizing the number of bits that could be stored on a single chip. Since the demand for larger and larger memories has been universal, DRAM designers have been able to meet the needs of all their customers with essentially the same design. Advances in storage capacity have taken place by periodical shrinkage and the industry has grown accustomed to a situation in which new designs would reach high volumes almost immediately.

Both sections of the industry exist in worlds that are highly competitive, but competitive in different ways. The DRAM industry is a mass market commodity industry in which sales are very sensitive to price. It survives on a knife edge, and finds it difficult to provide funding for major innovations that do not lead rapidly to large volume production. As I pointed out above, the processor industry is the exact opposite. It long term survival depends on innovations for which the immediate market is small.

Combined Design of Memory and Processor

It is fair to say that, within the constraints under which it operates, the DRAM industry has done what it could to meet the need for high performance chips. However. this has been achieved by modifying main stream chips in ways which do not involve a large investment, for example, by increasing the bandwidth to the cache and providing some form of burst mode [4]. These help, but what the high performance workstation and processor designer really wants is lower access latency. In practical terms this means, as I said above, major modifications to the chip and its packaging that will provide more pins, especially address pins.

In the presently existing situation, designers of processor chips intended for a high performance application can influence the design of the memory only indirectly. This means that they are unable to master mind the design of the processor and memory as a single whole. This matters more now than it did in the past, because it is becoming much harder to devise innovations, within the processor itself, which will enable more and more performance to be squeezed out of a workstation or a server.

Collaboration between the two sections of the industry will perhaps alleviate this situation. However, it is natural to ask whether one day it will not become possible to abandon the DRAM with its analog basis and change over to a purely digital memory, namely, one based on the same kind of logic circuits that are used in a processor. The SRAM is, of course, such a logic-based memory, but at the present time a switch to SRAMs would be out of the question, since it would involve using 16 times as many chips for the same amount of memory as we have now. However, in other fields there are cases in which a switch from analog to digital has been dismissed as impossible, and remained that way almost up to the moment at which it actually occurred.

A switch from DRAM to SRAM could occur naturally, if not entirely painlessly, if it were to come about that the capacitor technology used in DRAMs proved incapable of further shrinkage, but SRAMs continued to shrink for some further time. This appears to be unlikely, but if it did happen, it would only take a shrinkage by a linear factor of four for the capacity of SRAMs to catch up on that of DRAMs. A smaller shrinkage would be sufficient if the development of a smaller, but acceptable, SRAM cell proved possible.

What Next?

In discussions of the memory gap it is often suggested that hybrid techniques, whereby DRAM cells and a processor can be put on the same chip, offer a way ahead. According to the way the design is optimized, the result can be viewed either as (1) primarily a processor with some DRAM memory on the same chip, or (2) as primarily a memory chip with some processing power included. The former would constitute a complete computer on a single chip, and would undoubtedly find useful applications. However, it could hardly have enough memory to compete in high performance applications with current systems using multi-chip memories. The idea that a memory built with chips of the later kind—an intelligent memory—could in some way solve the fundamental problems of memory latency discussed in this article has been with us for some time. So far, it has proved a will-o'-the-wisp, and I expect it to continue in that way.

Among the fundamental physical problems encountered as further shrinkage of CMOS takes place, is the onset of quantum mechanical tunneling in very thin layers of insulation. It is tempting to suggest that the later might be exploited to make a tunnel diode memory, thus turning a problem into a feature. Tunnel diode memories were much talked of in the 1960s and even used to a small extent. They depend on the fact that diodes formed from two conductors separated by a very thin insulating film can, in virtue of the tunneling effect, show a negative slope resistance. In series with a resistor of suitable value, a diode of this type can function as a flip-flop. It is possible to construct matrix memories in which each cell consists of such a flip-flop together with a conventional diode for access. Tunnel diodes once appeared to have a promising future for memories and even for logic circuits. The first (instruction) cache to be built used tunnel diodes. However, tunnel diodes proved difficult to make and were swept away by the rapid development of planar transistors and integrated circuits. There is still plenty of activity on tunneling among physicists, as papers appearing in physics journals show, and from time to time there are references to possible applications. Unfortunately, there are no clear indications that practical tunnel diode memories will emerge either in the immediate or in the medium term future.

There is also interesting experimental work in progress towards the ultimate development of single-electron devices in which the presence or absence of one electron marks the difference between a zero and a one. This work is being done at a very fundamental level and, even if it is successful, it will inevitably be many years before it has any impact on the computer field.

Final Comments

So much for the long term. It will be interesting to see what happens in the shorter term. Moore's law has still some way to go, both for processor chips and for DRAM chips. As I suggested above, the end may come sooner for DRAM chips than for processor chips. In each case there must be an end, simply because of the laws of physics. The end has been heralded for some time by the steep way in which fabrication costs are going up. How it will all come about, no-one knows.

The memory gaps is certainly increasing, and there appear to be more and more economically important problems that will not go into any reasonably-sized cache. This would suggest that computers optimized for such problems will again appear—I say *again* because, as I have already mentioned, the CRAY 1 was of this kind. Perhaps, like the CRAY, they will have no data cache, or perhaps they will have means for bypassing it selectively for those accesses for which it would do harm rather than good. Workstations with dual mode operation will perhaps appear and appeal to those engaged in computer-intensive scientific research.

A final closing of the memory gap—as distinct from merely curbing its growth—could lead to a great simplification in computer systems. Perhaps it is too much to hope that such complications as data caches and branch prediction would disappear altogether, but they would certainly become de-emphasized.

Acknowledgments

I am indebted to many colleagues for helpful discussion of the topics treated here, especially to Dr C. P. Thacker, Dr T. Mudge, and Dr D. E. Roberts. However, the views I express are my own.

References

1. Hennessy, J. L., and Patterson, D. A. Computer Architecture - a Quantitative Approach. 1st Edition, p 408. Morgan-Kaufmann 1990.

2. Tulsen, D. M., Eggers, S. J., Emer, J. S., Levy, H. M., and Stamm, R. L. *Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor.* Proc 23rd International Symposium on Computer Architecture, p 191. ACM May 1996.

3. Eggers, S. J., Emer, J. S., Levy, H. M., Lo. J. M., Stamm, R. L., and Tulsen, D. M. *Simultaneous Multithreading.* IEEE Micro, p 12 September/October 1997.

4. Cuppu, V., Jacob, V. B., Davis, B., and Mudge, T. *A Performance Comparison of Contemporary DRAM Architectures.* Proc 26th International Symposium on Computer Architecture, p 222. ACM, May 1999.