



## Frequently Asked Questions

Answers to the Most Frequently Asked Questions  
Regarding ATP Memory Products

### **What is the difference between unbuffered, buffered and registered DIMMs?**

A buffer is used with EDO memory, and a register is used with SDRAM memory. They both work similarly to achieve the same function, but a register uses a synchronous SDRAM clock to lock in the signals, and a buffer simply buffers the EDO signals, as EDO memory has no synchronizing clock.

Most SDRAM and EDO DIMMs sold today are the unbuffered variety. When four or more heavily loaded, high-density DIMMs are used in a system memory, registers or buffers are added to each DIMM module to reduce the address and control signal load. Registers and buffers present a small load to the memory controller on the motherboard, yet each buffer or register may drive many chips (loads) on a DIMM module. DIMM modules can have as many as 36 memory chips on them, such as in stacked chip configurations used in very high-density memory.

A large buffered EDO server system can contain 16 DIMM modules and support 4GB of system memory with great reliability. The necessary downside is that a buffer will add about 5 nanoseconds to the system timing budget, and a register will add one CAS latency. A current rule of thumb: if your entire memory system load is near or exceeds 64 chips (4 DIMM modules x 18 chips/module = 72 chips) you will almost certainly need registered SDRAM or buffered EDO memory modules. Your system may run with 4 fully loaded unbuffered DIMMs, but it will likely have little or no noise margin left. Some systems won't even boot with a total of 64 chips installed on 4 unbuffered DIMMs.

### **What is ECC?**

ECC stands for Error Correction Code that can correct a bad bit in main system memory. Unlike the older parity bit, which would tell you if a bit was bad or not, ECC will correct 1 bit out of 64, and detect but not correct 2 bits bad out of 64 bits (64 bits is the standard DIMM data bus width today). In an ECC memory bus there will be 8 bits of ECC information along with 64 bits of data for a total memory bus width of 72 bits. That is why ECC memory typically has a "72" somewhere in the module configuration part number, and non-ECC typically has a "64".

### **Why does one need ECC, especially if most memory shipped in home PCs today has neither ECC nor parity?**

There are three somewhat interrelated reasons:

1. Cosmic rays can cause soft errors in DRAM memory, and can change a 1 bit to a 0 bit (never the reverse). The memory is not damaged, and that bit can be later written with a 1 and the memory will work fine just as before. That is not to say that a seriously powerful cosmic ray couldn't permanently damage memory, but that would be a very, very rare event at sea level. But how rare is this normal soft error type of event? Much effort over the years has been devoted to measuring and reducing the effect of cosmic rays on system memory. As a result, some very rough guidelines can be given for the sea level error rate in a 128MB

DIMM module, and that can be about 13 soft errors per year, or about one per month. Note that this rate is very dependent on how the memory is designed, and could easily be four times better than what IBM observed with some 4 Mbit Japanese EDO memory in one of their carefully done experiments involving millions of device hours of testing. Suppose, assuming a conservative estimate of cosmic ray induced soft events alone, your system with 128MB of memory crashed or suffered serious corruption about four times a year, would that be acceptable? I suspect you now have a feeling for why ECC is so important.

2. PC systems are using more memory nowadays, and soft error rates scale linearly with the amount of installed memory, all other things being kept equal. Systems with 32MB are being replaced with 64MB and 128MB, and soon 256MB systems will be common. Servers easily range from 1GB to 4GB of installed memory, and the uncorrected soft error rates for these larger systems are universally regarded as intolerable.
3. The die sizes for memory are shrinking, and it is easier for a puny cosmic ray to cause a memory bit to flip to zero. In fact, as memory sizes continue to shrink, there is an increasing probability that not one bit, but a cluster of bits will flip to zero. ECC will do a wonderful job of correcting a single bad bit, but in the future a more advanced form of ECC, which IBM calls chipkill correct, will be required to correct an entire memory chip that goes bad. IBM's Netfinity servers have chipkill correct, and all high-end servers and critical application systems require better correction than ECC. The good news is that normal PC systems will do just fine with ECC, and it may be a few years before more advanced correction will be needed here.

#### **What is the difference between 16MX8 chips and 16MX4 chips in a 256MB memory module?**

Memory modules can be made from a variety of different chips, but not all possibilities are permissible. A 256MB ECC DIMM memory module can be made with 18 16Mx8 chips or 36 16Mx4 chips. But it could also be made with 18 32Mx4 chips, 9 32Mx8 chips or 5 32Mx16 chips. (For a more complete understanding of what and why, please refer to the reference "Understanding DIMM Module Configurations, with Contemporary Examples.")

#### **What is the difference between different CAS Latencies, for example, between CL2 and CL3?**

A read from memory involves a row address strobe (RAS) followed by a column address strobe (CAS). The term CL refers to Column address strobe Latency (note that there is no difference in write performance for CL2 or CL3 memory). In a CL3 memory the read data is available 3 clock cycles after the CAS. In a CL2 memory the data is available 1 clock earlier, in only 2 clock cycles.

In typical systems in mid 1999, a CL2 PC100 memory will show around a 1 to 4% improvement in system performance over a CL3 memory. With even faster CPU speeds, or specialized graphic intensive applications, CL2 performance improvements can exceed 4%. One can have CL4 memory, and with the forthcoming DDR memory, one can even have CL2.5 and CL3.5.

#### **What is the relationship between the CPU, chipset, bus, and memory?**

A contemporary CPU does not talk directly to the main system DRAM memory, but through one or more intermediaries. One reason for the complexity is that as CPU speeds increase they are waiting longer and longer periods for the slow DRAM memory to respond. A second reason is that fast memory (the kind that fast CPUs would prefer) is very expensive compared to DRAM.

Typically the CPU will talk first to L1 cache that is a small amount of very specialized static RAM memory running at the same speed as the processor. The L1 cache memory typically talks to a

larger amount of specialized L2 cache static memory residing on or near the CPU, and under the direction of an onboard L2 cache controller. L2 cache is physically located on the chip in the new Celerons of Intel, or next to the CPU on what is called the backside bus for the slot 1 and slot 2 Intel cards, or on a bit of nearby motherboard real estate in older Pentium and 486 systems. The L2 cache then talks with the main DRAM memory through the intermediary of the "North Bridge" chip of the motherboard chipset. The CPU to L2 cache backside bus typically runs at half the CPU speed, but occasionally as fast as the CPU as in the case of the Celerons. The CPU to motherboard North Bridge chipset bus is called the Front Side Bus (FSB), and typically runs at 100MHz, although 133MHz is in the works, and 66MHz was the old, prior standard. The North Bridge chipset bus to the main DRAM memory is called the memory bus, and typically runs at 66MHz to 100MHz, and 133MHz (expected in September 1999).

If you think this arrangement is complex, you're right, but it is likely to get more complex in the future. An L3 cache between main DRAM memory and L2 cache is being considered for some of the newer systems. Note also the trend is to put more and more functions into the main CPU chip. I expect soon to see the DRAM memory controller, currently located in the North Bridge chip, to be integrated into the CPU silicon for a substantial increase in memory performance.

#### **Why do gold and tin contacts make a difference? Isn't gold better?**

Whenever dissimilar metals are placed in contact with each other, galvanic action can and usually does take place. You can think of a gold to tin contact as a very tiny, very low power battery. With time, crud can form in the gold to tin interconnection, very much like the crud you will find on ordinary batter terminals that are several years old. If the contact degradation becomes bad enough, you will have an intermittent or dead memory system. The solution is to use gold with gold, and tin with tin. One connector manufacturer claimed that in some rare situations tin to tin is best, but any contact expert knows that properly done gold to gold is by far the best, long term, low voltage connection.

#### **What kind of memory do Intel servers take? For example, the Intel SC450NX- does it have to use 4 pieces of memory together?**

Yes, and that is because this powerful memory system is 4-way interleaved for increased throughput. Note that many low end to medium servers of Intel (for example, MS440GX, T440BX, N440BX, L440GX and C440GX) are just like ordinary PC systems in that you can install a single DIMM and they will work fine.

#### **What are the advantages of EDO DIMM? Why is Intel server SC450NX designed for buffered 3.3V EDO DIMM instead of SDRAM?**

SDRAM is an improvement over EDO in that it synchronizes the DRAM operation between the clock of the chipset controller and the CPU clock. However, this benefit has often been overrated, and the saving due to synchronization is only on the order of 1 to 5%, all other things being kept equal. Other memory considerations can easily outweigh this small advantage. As an example, large server memories, such as the 4GB maximum of the SC450NX, require much greater system reliability than standard PC systems.

The design and qualification of the DRAM must conform to stringent guidelines for background radiation tolerance to prevent grossly unacceptable soft error rates in such a large system. When the engineering team selects a memory type for a high-end server, they may not be able to choose the latest technology if that technology has not proven itself with regard to acceptable soft error rates. I am not privy to the SC450NX design considerations, but I expect that well-established soft error rates for the chip types available at design inception were one of the more serious considerations for their choice.

### **What is SPD?**

SPD (Serial Presence Detect) is physically a tiny non-volatile memory chip on a memory module that can be read by the memory controller on the motherboard, and contains the critical parameter information for that module. This information includes memory type, size, speed, voltage interface, number of row addresses, column addresses, module banks, and other critical parameters. This data is used by the system BIOS to correctly configure the memory system.

A memory controller can be very sophisticated in probing what is on each memory module during boot time, but when an SPD is not used on the module, the memory controller must make assumptions about many critical timings. Unfortunately, the memory controller must be very conservative about such guesses, or the system might become unreliable, or in worse case it could hang forever during the boot cycle. An SPD, which is now required on all PC100 and PC133 modules, tells the memory controller, among many other things, what the fastest timings are that will work reliably.

When SPDs first came out, many motherboards would not bother to read what was in the SPDs. But in the last couple of years, most if not all motherboards read and rely on this information. Some motherboards have become so particular about certain parameters in the SPD that, if they are not what is expected, they may halt the boot cycle, or ask the user to manually agree to a choice of parameters during boot. SPDs have quickly transitioned from a largely optional memory feature to becoming an essential part of a fast and reliable system memory.

### **What is Rambus?**

Rambus is a radically new memory architecture designed to substantially increase effective memory throughput. In an attempt to accomplish this it reduces the traditional 64 to 72 bit data bus width to only 16 to 18 bits wide, but it runs this bus at much higher speeds, up to 400MHz. It clocks the data at both rising and falling clock edges for an additional speed boost to an equivalent 800MHz in best case. The commands and address signals are also unusual in that they are coded into packets so they can be sent over an independent 8 bit bus, and when they arrive at the memory they are decoded and acted upon. The technology has turned out to be unexpectedly expensive and difficult to implement, and many well-informed people seriously question its once-bright future. It will have to compete with the new PC133 memory, and soon with the very high throughput DDR memory.

Questions answered by  
Don Dilley,  
Manager of Engineering,  
ATP Electronics, Inc.  
September, 1999

***Need to Learn More about Memory?  
Please E-Mail Your Questions to Us at  
[info@atpusa.com](mailto:info@atpusa.com)***