ABSTRACT
While the SAS® System for personal computers will run on a PC/XT™ or compatible with 10 megabytes of disk space and 640 K of RAM, it is not recommended. This system will deliver extremely sluggish performance and will have little work space available. This paper recommends two machines for running the SAS System and discusses how to configure each of these to achieve the kind of performance with which you will be satisfied.

INTRODUCTION
The SAS System is a large and sophisticated computer package and it requires a powerful PC to perform well. The best machines for running the SAS System are AT® or PS/2® class machines powered by the Intel® 80286 or 80386 Central Processing Units. The configuration of the machine is vital to its performance. PCs have become modular units that can be pieced together in several ways. Hopefully, this paper sheds some light on the mysteries of configuring a PC.

MACHINES
Recommended machines for running the SAS System are:

• 286 machine:
  A 286 CPU running at 10 megahertz or faster with 640K - 1 megabyte of RAM on the mother board.
  A 40 megabyte or larger hard disk, running at 40 milliseconds or lower average access time.
  • Optional components are:
    An Expanded Memory Specification (EMS) board with 1 megabyte or more of RAM.
    Disk-caching software capable of running from Expanded Memory.
    A 287 math coprocessor

• 386 machine:
  A 386 CPU running at 16 megahertz or faster with 1 megabyte of RAM on the mother board.
  Software capable of converting Extended memory into Expanded memory. (Note: This is recommended with a 386 CPU only).
  A 60 megabyte or larger hard disk, running at 30 milliseconds or lower average access time.
  • Optional components are:
    An additional megabyte or more of RAM on the mother board.
    Disk-caching software.
    A 387 math coprocessor.

Both of these machines achieve good results. The 386 machine will deliver superior performance but will obviously cost more. A good configuration for most users is to equip either machine with 1 megabyte of EMS and a 1 megabyte disk cache. Effectiveness of a math coprocessor depends on the type of jobs run.

PC COMPONENTS
It is important to understand the weaknesses of present day PC systems in order to understand how to configure them. The principal problem is the inability of PC DOS to allow more than 640K memory for program execution. This affects both the capacity and speed of the PC. Its effect on the capacity is obvious, since an application program such as the SAS System has less than 640K available in which it must load both executable code and data. This means that the amount of data that can be loaded is severely limited.

This affects the speed of the PC because the SAS System is large and many of its components will not fit into available RAM without being broken up into smaller pieces. Each piece is loaded into RAM, executed, and then removed so that another piece can be loaded. All the pieces taken together constitute a system (such as a SAS PROC). This process is known as overlaying. The problem is that pieces are constantly being loaded from the hard disk into RAM and the PC is slowed because it may spend more time loading pieces from disk than it does executing what it has loaded.

The first decision to make is whether the area that most needs improvement is speed or capacity. You are the only one who can answer this with the detailed knowledge of the type of work which you will be doing. In this paper, capacity is discussed first, then speed.

EMS
The workings of EMS are discussed later in this paper in some detail. The important thing to understand about EMS is that it allows you to run larger jobs on your PC. There are two types of situations where EMS can make a real difference:

• When you are running a job in which your data set has a large number of variables.
  • When you are running full-screen procedures such as SAS/GRAPH®, SAS/ASSIST® or SAS/FSP® software.

EMS is generally not regarded as a speed enhancement, and in some cases EMS may actually be detrimental to speed. There are many factors affecting speed so it is hard to predict. The important thing to remember is that the primary reason to install EMS is capacity, not speed.

If you are on a budget and are willing to take advantage of the Display Manager System, a significant capacity gain can be made by running the SAS System in noninteractive mode. This will save approximately 130K of RAM, and enable you to run proportionally larger jobs. Consult the SAS Language Guide for Personal Computers, Release 6.03 Edition or SAS Technical Report P-171, Changes and Enhancements to Base
SAS Software for Personal Computers, Release 6.03 for instructions on running the SAS System in noninteractive mode.

Normally, you should use 1 megabyte of EMS, but there are situations when more might be required. These would typically be when running SAS/ASSIST software or large SAS/FSP or SAS/AF software jobs. For these situations, 1.5 or even 2 megabytes might be appropriate. It is not the case that if a little EMS is good, more is better.

With a 286 CPU, an EMS card must be purchased in order to install EMS on the machine. With a 386 CPU however, an EMS card is not necessary. Memory can be added directly to the motherboard, and a third party EMS driver can be purchased for a nominal price. This will not work with the SAS System on a 286 CPU and should not be attempted.

**DISK CACHING**

The only reason to install a disk cache is to increase speed, which it does remarkably well. A disk cache will not improve capacity and will actually reduce it by the amount of memory that the caching software takes, but this is usually minor.

While it is true that the larger the cache the better, experience has shown that the performance improvement is most cost effective at about 1 megabyte of RAM. If your machine has more RAM you should use it, but it is probably not cost efficient to buy more for this purpose. A 1 megabyte disk cache will provide a spectacular speedup of your PC.

Most commercially available disk caching software now runs from EMS or extended memory (discussed later). This presents an opportunity for owners of machines with 1 megabyte on the motherboard to enhance performance with little cost. Such machines have 384K of memory above the DOS 640K barrier and memory can be configured as a disk cache. Since the memory is usually not being used by any other application, you can get a significant performance increase for the price of the disk caching software. This is a spectacular performance buy. Several manufacturers now include disk caching software with the machine, (IBM® includes it with PS/2 model 50 or higher) so check the documentation that came with your machine.

If your machine is running from a Local Area Network (LAN) with the SAS System resident on the server, you may well have disk caching as part of your network services (this is almost always true with Novell®). If this is the case, the only speed increase available is from caching the local hard disk that holds the SAS work files. While this will result in some speed gain, it is probably not worth the cost of buying additional RAM. If you have a 1 megabyte machine with a spare 384K however, the disk caching software is still a good buy for this purpose.

**MATH COPROCESSORS**

A math coprocessor can provide a speed boost for your PC, but the size of the speedup depends heavily on the type of jobs you run. If you run mainly DATA steps that handle lots of character data and produce reports, a coprocessor may not be worth the cost.

However, if you use a lot of statistical or graphical procedures, a coprocessor can significantly increase speed. In certain cases of heavy numeric calculation, coprocessors can more than double speed.

There are a few pitfalls to avoid when installing a coprocessor. The coprocessor chips are rated for speed and if a chip is run faster than its rated speed, its calculations are unreliable. It is vital that you determine the speed at which your PC drives the coprocessor before you order or install a coprocessor. Generally, the coprocessor is driven at a fraction of the main processor speed. The only way to determine the speed at which your PC drives a coprocessor is to check the documentation or contact the manufacturer. Under no circumstances should you install a coprocessor rated slower than the speed your PC drives it.

In order to understand the process of configuring a PC in more depth, the rest of this paper discusses the internals of EMS and disk caching in some detail.

**EMS DETAILS**

Expanded Memory Specification is the solution to a problem that arose in the PC industry in the mid-1980s. To understand the solution it is important to first understand the problem.

Consider the logical memory map represented in Figure 1. This identifies three kinds of memory:

- conventional memory, the area from 0 - 640K
- reserved memory, the area from 640K to 1 megabyte
- extended memory, the area above 1 megabyte.

Note that extended memory does not exist on the XT™ type machine. This is because the 8088 and 8086 CPUs used in the XT are incapable of addressing past 1 megabyte. Thus, the XT is constrained by hardware from addressing the area above 1 megabyte, and an XT cannot have extended memory.

The problem was that PC users were unable to run large problems with their application software, especially spreadsheets. An easy solution would have been for software writers to agree on a standard and use extended memory, because this would have allowed up to 15 megabytes of additional memory. However, at the time there were many more XT's existing than AT's. Any solution that was to help the majority of the market would have to work on XT's.

The problem of allowing more than 1 megabyte of memory to be used on an XT that contains a CPU that will address only 1 megabyte was not an easy one. Consider the physical memory map represented in Figure 2. This shows that while the conventional memory area from 0 - 640K is physically on the motherboard (on some machines, 512K is on the motherboard and the rest on an add-in board), the reserved areas from 640K - 1 megabyte is vacant, except for memory that resides on add-in cards and is mapped to the reserved region. Examples of cards that contain memory mapped to the reserved area are display controllers (CGA, EGA, VGA), disk controllers, and most other add-in cards, especially EMS cards. Figure 3 shows the physical memory map for a PC with 1 megabyte of RAM on the motherboard.

The scheme that was devised is represented in Figure 4. Note that the EMS card contains a series of memory pages that are 16K each. It also contains a set of chips called paging hardware. The paging hardware is able to map any 16K page in the EMS page frame, which is a pre-determined section of reserved memory (640K - 1 Meg). The paging hardware is controlled by software called an Expanded Memory Manager (usually...
abbreviated as EMM.SYS). Remember, an XT normally can only
address 1 megabyte of memory, but now it can actually see all
the memory (up to 8 Meg) on the EMS card. It can not see it all
at once, but it can see whatever pages are mapped into the page
frame. Since the EMM software can control the paging hardware
and tell it to map any 4 of the 16K pages into the EMS page frame
(where it will be visible to the CPU), the entire 8 megabytes of the
EMS card is now available to the CPU. Because an AT or PS/2
class PC emulates an XT, this scheme works essentially the same
on all PCs.

It should now be obvious how expanded memory differs from
extended memory. In Figure 1, you can see that expanded mem­
ory is simply memory above 1 megabyte. In Figure 4, you can see
that expanded memory consists of four components:

- RAM
- paging hardware
- a 64K page frame
- an Expanded Memory Manager

If any of these components are missing, you have not fulfilled the
EMS specification and cannot use EMS.

This picture was somewhat complicated by the arrival of 386 PCs.
The 386 chip contains memory management hardware that can
do the same thing as the EMS paging hardware. Thus, one of the
components of EMS already exists on a 386 PC. Consider Figure
5, which portrays EMS installation on a 386 machine. Note that
the 4 components of EMS still exist, but 2 are in different form.

- The paging hardware is now on the 386 CPU.
- The RAM is now on the mother board (or a special add-in
extension to the mother board).

The other components of EMS, the expanded memory manager
and the EMS page frame, are unchanged. Note that with this con­
figuration, it is not necessary to add an EMS board, additional
memory can simply be added to the mother board.

At this point, it is important to understand how the SAS System
takes advantage of EMS. Figure 6 shows a typical 286 EMS
installation. Note that the SAS System loads executable code into
the RAM pages. The SAS System then tells the Expanded Mem­
ory Manager what module it wants loaded and the EMM controls
the paging hardware to map the appropriate page into the EMS
page frame so it is visible to the CPU.

EMS increases the capacity of the SAS System in two ways. First,
because SAS executable code is loaded into EMS memory, it
frees conventional memory to hold data, thus increasing capac­
ity. Second, if a large amount of SAS executable code is required
to be loaded, such as when running SAS/ASSIST software, the
entire EMS memory is available to load it.

DISK CACHING DETAILS

The principal way to increase the speed of your PC is with a disk
cache. After you have installed EMS, you will probably want to
install a disk cache to increase performance along with capacity.

The first thing to understand is that mechanical motion is much
slower than electronic switching. The strategy of disk cache soft­
ware is to eliminate the mechanical motion of the hard disk and
replace it with electronic switching used in RAM. To understand
how to do this, you need to understand how a hard disk sub-

system works. Figure 7 represents such a system. Note that the
disk is divided into sectors.

1. When a program requests data or executable code from
the disk, it requests it from DOS
2. DOS then passes the request on to the disk controller
3. The disk controller physically retrieves the requested
sector from the hard disk
4. The contents of the sector are then read into RAM and
used by the program

Figure 8 shows how things happen when a disk cache is present.
Note that the cache program is loaded into conventional memory.

1. The application program requests a file from DOS as
before.
2. But when DOS requests the sector from the controller,
the request is intercepted by the cache program.
3. The cache program then checks the cache area (which is
represented here in extended memory but can also be in
EMS) to see if the requested sector is stored there. If it
is not, the sector is retrieved from the disk and stored in
the cache area before being passed on to DOS to be
loaded into RAM.
4. If the sector is stored in the cache area, it is loaded
directly from the cache area to RAM.

Simply moving the sector from one section of memory to another
is much faster than reading it from the spinning disk and this is
how the cache saves time.

Most disk cache software uses some form of a least recently
used (LRU) algorithm to determine which sectors are requested
the least. If the cache is full, it will cache a new sector over the
sector that has been requested the least. Thus, the cache eventually
fills with the sectors that the program uses the most. This can
lead to dramatic time savings, especially with a system like the
SAS System that uses the disk heavily.

Most disk caches are called "write through," meaning they cache
sectors that are read from the disk but do not cache sectors that
are written to the disk. To do so would save some time, but it is
dangerous if the program hangs or the machine loses power.
Data might not get written back to the disk.

CONCLUSION

In the PC environment, most users are forced to be system ana­
lysts and hardware maintenance people. Hopefully, this paper
has removed some of the mystery from PC systems and will pro­
vide assistance to users who want to purchase a new PC or
improve the performance of an existing PC.

SAS, SAS/SHARE, SAS/AF, and SAS/GRAPH are registered trade­
marks and SAS/ASSIST is a trademark of SAS Institute Inc.,
Cary, NC, USA.

Novell is a registered trademark of Novell, Inc.

Intel is a registered trademark of Intel Corporation.

IBM, PS/2, and AT are registered trademarks and PC/XT and XT
are trademarks of International Business Machines Corporation.
Figure 1
Logical Memory Map

Figure 2
Physical Memory Map
Figure 3

Physical Memory Map for a 1 Megabyte Mother Board

Figure 4

286 EMS Installation
Figure 5
386 EMS Installation

Figure 6
The SAS System and EMS
Figure 7
A Hard Disk Subsystem

Figure 8
A Hard Disk Subsystem with Disk Cache