The “BAREFOOT” ENGINEER

Presented by John Brock

John Brock is an electronics engineer with over 20 years experience in the electronics industry, with the last 5 years concerned with microprocessor based development. He has designed communications equipment for the South African Air Force as well as time and attendance systems. He also has designed a series of microcontrollers for time and data process applications. In more recent times he has written communications and network applications software for the PC. As an independent consultant he provides development, support and backup facilities for individual and corporate clients.

As a support person you need tools to test and diagnose PC problems. The aim of this course is to provide you with your own tools. These tools you will generate, using the software provided with DOS. Such tools include the use of DEBUG and QBasic or BASIC. An understanding of the hardware and the way it is interconnected will also be provided to help you find the majority of faults. The emphasis will be on the practical aspects.

Most PC problems today stem from incorrect configuration of either the system hardware or the operating system. So I have added a sizeable section on configuration of the machine.

Software tools are available, but at a price and operate usually without the users understanding. You probably know half a dozen users who have a copy of PCTOOLS or NORTON and profess to use it regularly. One user of mine insisted that he had to have PCTOOLS on his hard disk. He got a copy somehow, and promptly formatted his hard disk!

I have found over the years that the best tools are the ones you make yourself, and the next best, the ones you know how they work. So with that in mind, I am going to show you how to build your own tools for testing the PC.

Section 1 Computer fundamentals - building blocks in hardware.

A brief history of the PC

1951 The SAGE project links radar sites in the US at 1300Baud.

1961 Leonard Kleinrock of MIT publishes first paper on packet switching.

1968 Messrs Noyce and Moore leave Fairchild to found Intelligent Electronics (INTEL).

1971 4004 developed by INTEL developed for a calculator company called Busicom.
This first microprocessor was introduced by Intel in 1971 and was a 4-bit machine called the 4004. This was the first of its kind; a 4-bit machine, oriented towards calculators but capable of very much wider application.

1972 8008 8 bit processor follows the 4004.
The latter was superseded by the 8080, a ‘N.M.O.S.’ processor with a 2μS instruction cycle and 70 instructions. The power dissipation is only 600 mW and there is a full range of r.o.ms and r.a.ms, clock and interface receivers and drivers so that a complete system could easily be built.

1973 Bob Metcalfe invents the “Ether Network” based on CSMA/CD.

1974 Vinton Cerf & Bob Kahn produce the protocol suite TCP/IP.

1975 MITS introduced the Altair 8800 using the 8080 256 bytes of memory, toggle switches for programming and LED front panel.
Z80, 6502 and 6800 all developed. Z80 became available in Feb 76.
KIM-1 single board computer 1k bytes of memory, 2k bytes of rom
IBM announced a 50-Pound briefcase size computer called the 5100. It had 16k Bytes of memory, BASIC and a 16 line by 64 character display with a magnetic tape storage system.
Bill Gates and Paul Allen found Microsoft and write microcomputer BASIC on punched tape!

1976 Apple 1 based on the 6502 built by Steve Jobs and Steve Wozniak. The Apple 1 had no case or power supply or peripherals. They proposed personal computer designs to both Hewlett Packard and Atari, both were rejected!
Texas Instruments released the TMS9000 a 16 bit microprocessor.
Cromemco released the Z-1 a Z80 system with 8k bytes of ram and serial I/O.
RCA introduced the 1802 microprocessor the first CMOS microprocessor.
Shugart released the "minifloppy" disk drive to use 5.25" disks.

1977 The Apple II is released using the 6502 and 16k bytes of memory expandable to 64k, 16k bytes of ROM, a keyboard, cassette interface, eight slot motherboard, game paddles, and colour capable graphics/text display.
The TRS-80 was brought out by Tandy Radio Shack. It used a Z80 and had 4k bytes of ram, 4k bytes of rom (including BASIC), a keyboard, display and cassette interface.
Gary Kildall develops CP/M which becomes the standard operating system for 8080/Z80 based pc's.
Commodore Business Machines unveils the PET based on the 6502.

1978 Epson introduces the MX80 dot matrix printer.
MicroPro international is founded and releases a word processor package called WordMaster.

Apple and Radio Shack both announce 5.25" disk drives.
Texas introduces Speak and Spell.

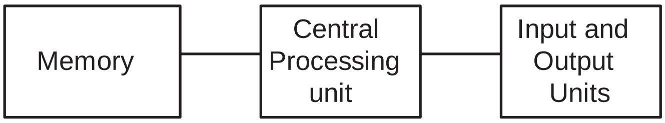
1979 Visicalc shown off by Dan Bricklin and Bob Frankston
Hayes releases a 110/300 Baud Micromodem which autodials and autoanswers.
Intel announces the 8088 and 8086.
Wayne Ratcliff develops the Vulcan program, which will later become dBase II.
Xerox, DEC and Intel announce Ethernet.
3Com is founded by Bob Metcalfe.

1980 Clive Sinclair releases the ZX80 based on the Z80 with only 1k ram, 4k rom Basic which plugged into a standard TV.
Commodore Vic 20 released.
Apple III finally released.
Shugart starts selling the Winchester drive hard disk.
CP/M 86 by Digital Research released.
WordPerfect for Data General computers.

1981 Osborne 1 portable/Luggable.
The IBM PC released.
Clive Sinclair produces the ZX81 to replace the ZX80.

The components of the microcomputer.

All microcomputers contain similar components. These are the Central Processing Unit or Microprocessor, read only memory, read-write memory and the Input and Output (I/O) units.



The Central Processing Unit (CPU)

The CPU consists of three components the ALU (Arithmetic Logic Unit), The Control Unit and the Registers. The ALU performs all calculations, comparisons and logical operations, using the registers for data manipulation. The coordination of the microprocessor, and the microcomputer, is done by the Control Unit. The interpretation of the programming instruction code and the generation of the internal and external control signals are also handled by the Control Unit. The CPU requires instructions before it can do anything, and these are placed into Read Only Memory by the manufacturer of the motherboard. Memory is accessed by the CPU by means of an address, data and control bus connecting structure. Input and Output "ports" are accessed in a similar manner.

The Memory

The memory that is used in microcomputer systems is split into two types, the read-write or Random Access Memory (RAM) and the Read Only Memory (ROM). The main function of the memory is to store instructions or programs and data for use by the microprocessor.

The Input Unit

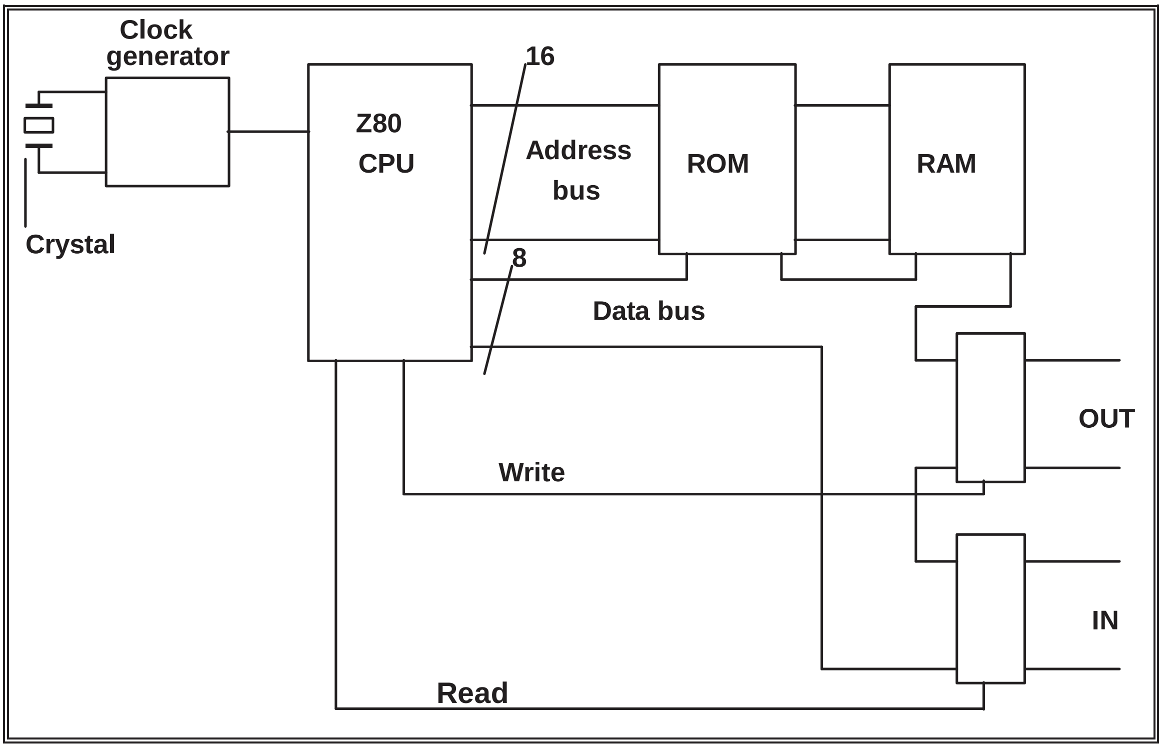
Input is the way data is put into memory. The Input unit is a device that interfaces the microcomputer system to the outside world. Input units generally used are the keyboard, a mouse, serial interface, joystick adapter, etc.

The Output Unit

When the micro computer has completed its computations, it must have some method of displaying the results. This method is known as output and the output units handle the display of information. The common output units include the monitor and the printer.

Devices such as hard and floppy disk drives, and modems are known as input/output devices as they are used for both input and output.

A simple microprocessor system

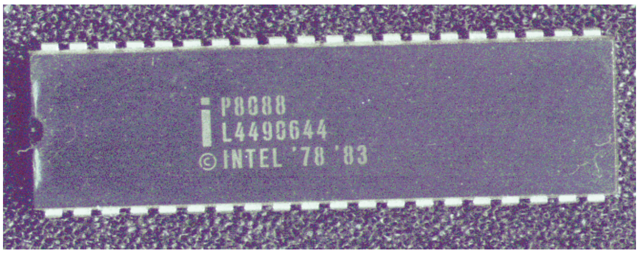


Our simple microprocessor starts operation by setting the address bus "bits" all to a logical zero. This corresponds to an address of zero. It then does a memory read of what is at the first location and this is placed onto the data bus by the ROM. It looks at the code it finds there up in an internal table, and proceeds to execute that instruction. The instruction may refer to the next location, if so it will set up the address bus accordingly to a logical one, and do another memory read. It will keep reading instructions from memory until it is either reset or it is told to wait for something to happen. One method of checking out the Z80 system was to tie the data bus to logical zero, and watch the address bus step from zero to 65535. This relied on the fact that the code for NOP, or no operation, was zero. Unfortunately that does not apply to the 8088 series of microprocessors.

The Z80 processor grew out of the development of the 8080 by INTEL. The operating system that became an industry standard based on both processors was the CP/M system. The diagram shown is a simplified version and does not show any "GLUE" chips. These chips would have been used to interconnect the major integrated circuits used in the complete system. These PC's preceded the advent of the IBM PC in 1981, and were still being sold and used in 1985.

Intel’s microprocessor family

The microprocessor used in the IBM compatible microcomputers is based on a CPU manufactured by INTEL or one of its competitors. Several microprocessors have been manufactured by Intel for the IBM PC range, these are the 8086 and 8088, the 80186 and 80188, 80286, 80386, 80386SX, 80486, 80486SX, the Pentium and Pentium Pro. The range of microprocessors also includes the 8087, the 80287, the 80387SX and the 80387. The x87 ranges of microprocessors assist the x86 range by handling floating point calculations. The coprocessors have been incorporated into the cpu from the 80486 up.



Plastic 8088 processor

8086 & 8088

The 8088 microprocessor was used to develop the original IBM PC. Other manufacturers of microprocessors have designed and marketed, other compatible microprocessors, under licence from Intel. These are NEC (V20 and V30), which amazingly could also be switched into a Z80 mode of operation.

The 8088 uses 16 bit sets for instruction and data manipulation, but will only output data in an 8 bit format. It is a 16 bit instruction, 8 bit data microprocessor. A 20 (A0 to A19) line address bus is used to access memory therefore the maximum amount of memory that it can access is 1024Kb or 1Mb. This is the reason PC DOS and MS DOS could only access 1Mb total memory without the aid of device drivers or Dos Extender software.

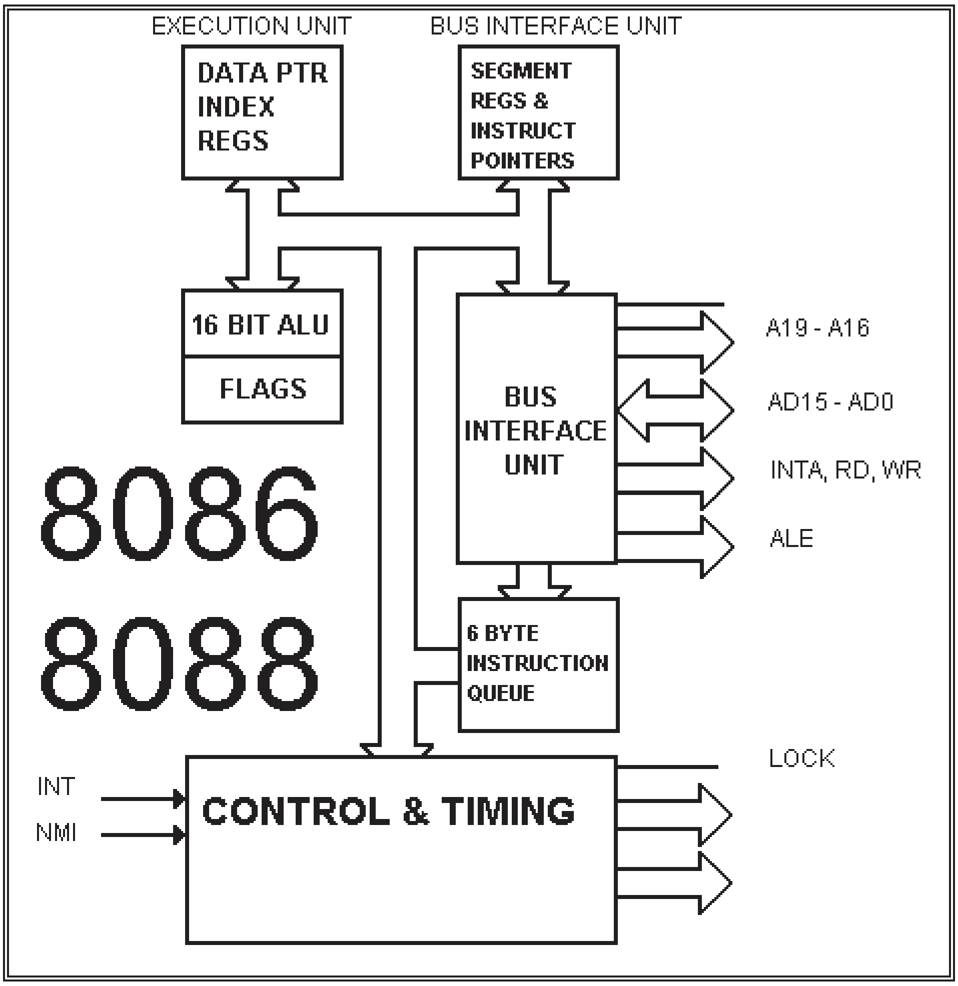
As the microprocessor runs it takes from a memory location an instruction, executes the instruction and then, unless the instruction directs otherwise, goes to the next memory location and repeats the process. If an instruction directs the microprocessor to access another area of memory, then the microprocessor will move the instruction retrieval to the new memory location and start processing from the specified point. Roughly 1 million instructions per second can be executed.

The differences between the 8088 and 8086 are that the 8086 will output data as 16 bit information. The 8088 is known, by Intel, as an 8 bit processor and the 8086 is a 16 bit processor. This is the only difference between the two processors. The 8086 is totally "code compatible" with the 8088. The 8086 was used in the IBM PS/2 model 25 and in some of the AT&T (Olivetti) microcomputers.

The original 8088 could run at a clock speed of 4.77MHz (Millions of Cycles per Second), however the 8088-2 and later versions ran at clock speeds up to 12MHz. These processors could be slowed down to 4.77MHz, by using a slower clock speed. This has been marketed as the ‘Turbo’ option. This is no longer a real option and turbo buttons should be disabled to stop users from accidentally slowing the pc.

The NEC V20 and V30 microprocessors were designed as plug in replacements for the 8088 and the 8086 and use much of the additional features of the 80188/6. Intel stopped production of the 8088 and 8086 microprocessors during 1988 and the Fujitsu and NEC have also stopped production of the 8088's manufactured under licence.

Block diagram of 8088/86



One point to note about the diagram is that the address bus lines, AD0 to AD15 are shared with the 8 or 16 bit data bus. This was because the chips were made in 40 pin dual in line packages therefore not providing sufficient pins for connection. Later processors had more pins.

NOTE: The 6 byte instruction queue is a form of caching.

8088/8086 Registers

High 8 bits	-combined name-	Low 8 bits	Use of register
AH	-AX-	AL	ACCUMULATOR
BH	-BX-	BL	BASE
CH	-CX-	CL	COUNT
DX	-DX-	DL	DATA
	SP		STACK POINTER
	BP		BASE POINTER
	SI		SOURCE INDEX
	DI		DESTINATION INDEX
	CS		CODE SEGMENT
	DS		DATA SEGMENT
	SS		STACK SEGMENT
	ES		EXTRA SEGMENT
	IP		INSTRUCTION POINTER

FLAGS REGISTER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

All the following processors including the Pentium II contain these registers with some additional registers and enhancements.

80188 & 80186

Intel incorporated some extra features into this design and it was used in some XT type machines. It featured:-

an enhanced 8086-2 CPU
clock generator
2 independent, high speed DMA channels
a programmable interrupt controller
3 programmable 16 bit timers
programmable memory and peripheral chip select logic
programmable wait state generator
local bus controller

It was available in versions with clock speeds of 8, 10 and 12 MHz. It still addressed only 1 megabyte of memory. It was however totally "code compatible" with the 8088/6. This processor is generally used for SCSI controllers and dedicated control functions.

80286

This microprocessor was used as the CPU for the IBM PC/AT series of microcomputers. The 80286 for upward code compatibility contains the same instruction set as the 8086, but has increased and enhanced the total instruction set. This means that the 80286 can run the same operating systems and programs as the 8086/8. Several of the microcontroller support functions were also built into the 80286 these are:

Memory management circuitry that allowed the 80286 to use Protected mode and to access up to 16MB of memory.
Interrupt controller circuitry.
DMA control circuitry.
Bus cycle decoding circuitry.

These functions were previously handled by separate support microcontrollers on 8086 and 8088 motherboards. The result of this incorporation is that the 80286 has had the effect of allowing the 80286 motherboards to become smaller. Baby or XT sized AT motherboards are an example of this. Though the original AT motherboards were very large requiring special cases.

The 80286 has four internal units, which provided the enhanced functions. These are:

The Memory Manager allows the 80286 to access 16Mb of physical memory.
The Bus Controller controls a restrictive bus mastering concept.
The Instruction queue prefetch processor that speeds up the instruction usage.
The Execution unit was the same as the 8088 and 8086.

The external and internal bus widths of the 80286 are both 16 bits wide, this makes the microprocessor a true 16 bit processor. This was later referred to as the Industry Standard Architecture or ISA configuration. The Instruction processor enables the microprocessor to handle program code faster, as the next instruction from memory is fetched while the previous instruction is being executed.

With the Execution unit being similar to the CPU of the 8088 and 8086, the emulation of the earlier microprocessors was done at up to six times faster, when using the same clock speed. The clock speed was also variable and could range between 8MHz and 25MHz. The external clock frequency was twice the actual clock frequency used internally.

The protected mode of the 80286 was not used by DOS as this would have meant a complete rewrite of the operating system. Microsoft, in conjunction with IBM, based the original OS/2 operating system kernel on the protected mode. OS/2 was not available when the IBM PC/AT was released, so the standard operating system was PC DOS.

During this period software became able to use more and more memory. A typical program being Lotus 123. To aid the XT users a memory scheme called Expanded memory was devised. The three companies that developed the Expanded memory specification were Lotus Corporation, Intel and Microsoft. This became known as the "LIM" standard. This hardware was quite often added to AT motherboard designs.

80386

With the release of this microprocessor the 32 bit standard that most new operating systems use was created. The 80386 has enjoyed a greater acceptance because the operating systems and programs would not have to be upgraded to be run at full capacity on later microprocessors.

The 80386 has the same functions as the 80286, but they have been improved and enhanced. The memory management unit can access 4 Gigabytes of memory directly and 16 Terabytes of virtual memory. Enhancements to the Bus controller allow full bus mastering to be controlled by the microprocessor. The real power of the 80386 lies in the Virtual mode of the microprocessor, enabling the microprocessor to emulate multiple 8086/8 microprocessors.

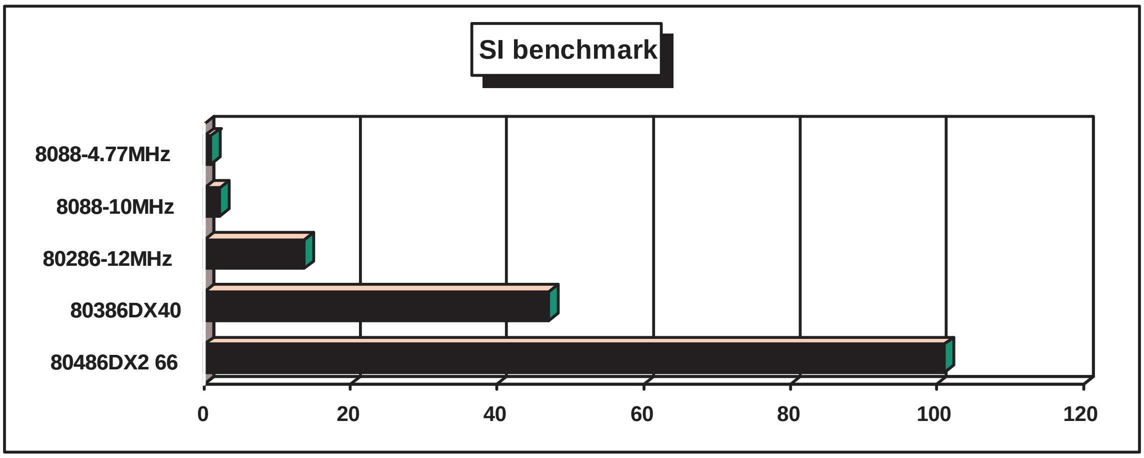
The speed range of the 80386 was also been improved to allow a speed of up to 33MHz. This enables many operating systems and programs to enjoy a faster and more efficient execution.

The enhanced features of the 80386 have been used by software developers to allow true multitasking to be done. OS/2 has been upgraded to take full advantage of the facilities and functions now available, as have Novell and Santa Cruz Operation (SCO) unix.

With software and operating systems now utilising the 32 bit structure and architecture of the 80386, a replacement for the 80286 microprocessor was developed. This new microprocessor was not better than the 80386 but rather was a downgraded version of the 80386 and is known as the 80386sx processor. The only major changes made to the 80386sx was the restriction of the maximum direct memory access to 16Mb and the restriction of the speed range to between 16MHz and 25MHz. The pricing of the 80386sx is similar to that of the 80286 and, with the virtual mode still available, is a more than adequate replacement for the 80286. American Micro Devices (AMD) have produced versions of the SX which can run with clock speeds of up to 33MHz.

The downward compatibility of the 80386 and the 80386sx microprocessors allows the use of operating systems and software that were originally written for the 8086/8 and the 80286. The operation of the 80386 based microcomputer is restricted to the use the software makes of the 80386 capabilities.

How the processors improved



80486

Because of a greater demand from microcomputer users for faster and more efficient processing, the 80486 microprocessor was developed. As with the 80386 the 80486 had all of the features of its predecessors, with a few added extras

The Math's Coprocessor was integrated into the chip.
8 kilobyte onboard cache memory was also integrated into the design.

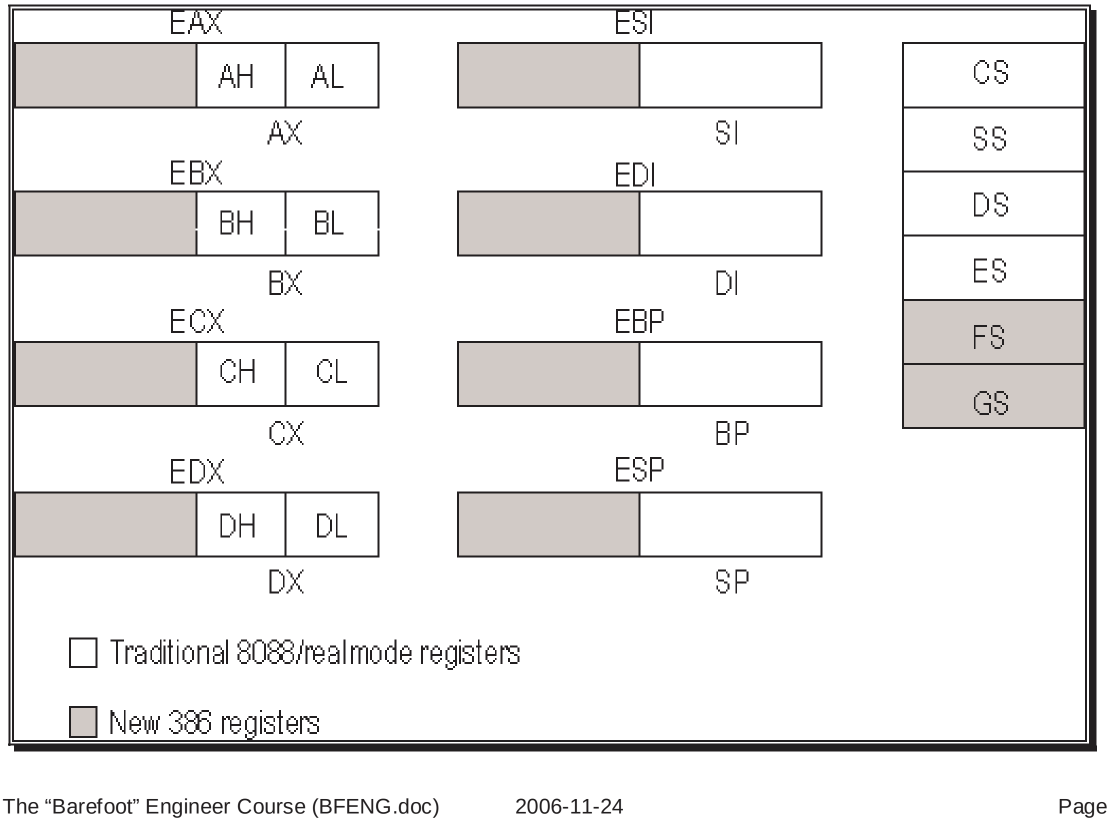
The math's coprocessor is a separate CPU and enables the 80486 to handle floating point calculations. This situation gives a greater flexibility to high end computer users, as the speed of the microprocessor is enhanced considerably.

The Cache memory further speeds up the microprocessor operation, by keeping heavily utilised data in the microprocessor. This also increases the effective speed of the microprocessor. The clock speed range of the 80486 is from 20MHz to 50 MHz. Internal doubling of the clock signal as used in the DX2 type 80486, can take the effective clock speed to 100MHz. The DX4 types which use clock tripling have taken the standard to 100MHz internal, 33MHz external. AMD produced an 80486DX4 120MHz processor which whilst externally clocked at 40MHz, runs at 120MHz internally. This is a replacement for the widely used 80386DX40 cpu.

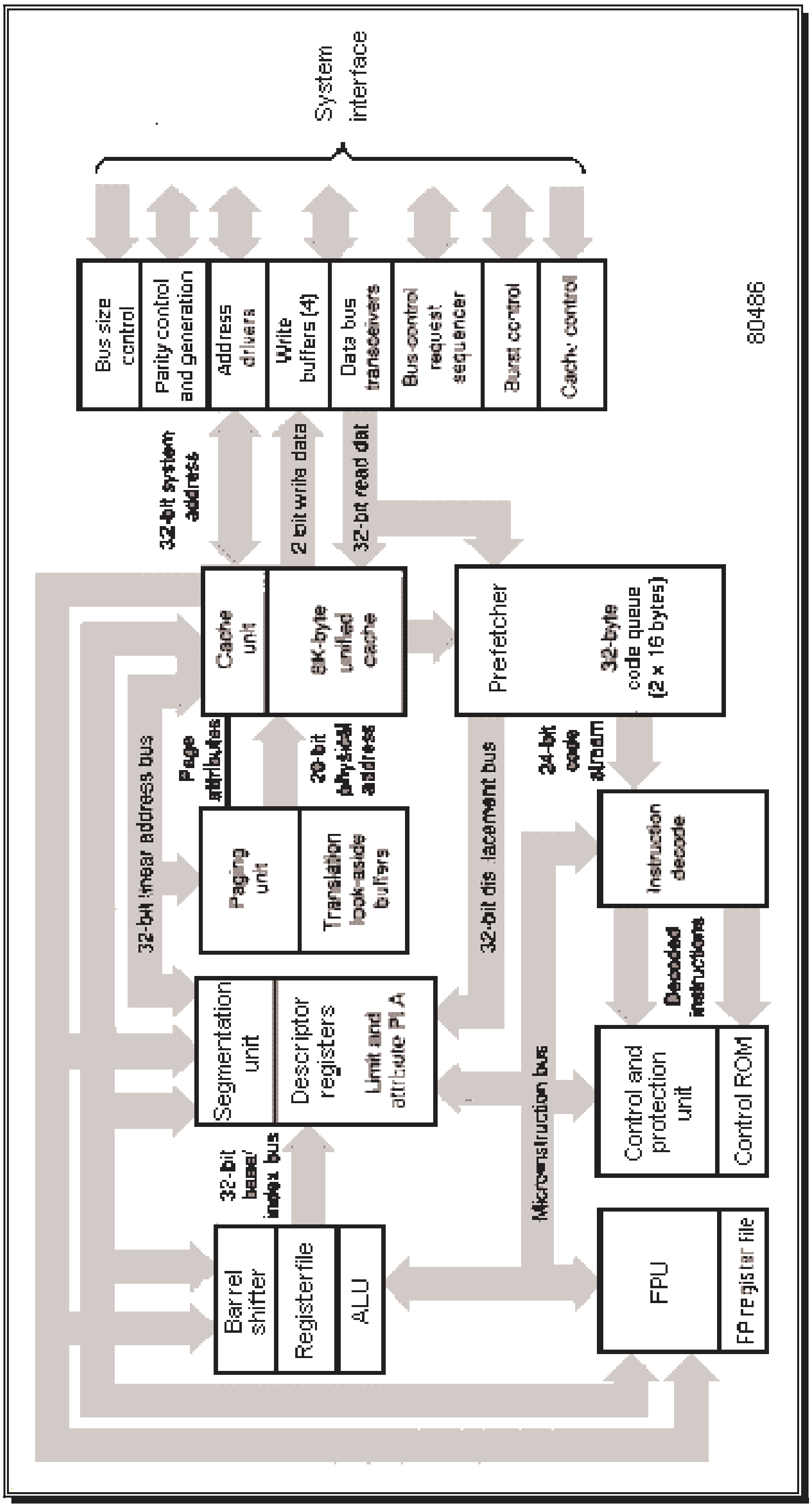
The 80486 is a 32 bit microprocessor, to keep the standard consistent, as such uses a similar architecture to the 80386. This allows most microcomputers to upgrade from an 80386 microprocessor to an 80486 with the minimum of cost.

A scaled down version of the 80486 was available, designated the 80486SX. The 80486SX restrictions include; a restriction of speed to 33MHz and the loss of the internal math's coprocessor. Clock doubled versions of the SX were available, though these ceased production mid 1995.

The extensions to the 8086 registers in the 80386/80486



The 80486DX Block diagram



The “Pentium”

The recent processor from INTEL is the “Pentium”, which would have been called the 80586 if they had kept to their original numbering scheme. The reason for the change is their inability to stop their competitors from using the number, as it cannot be trademarked as their own. So we have the “Pentium”.

The processor is quite different from all the rest in design and has a great many features to enhance performance.

- Approximately 3 million transistors
- 8k code and data write-back caches to increase performance.
- 64 bit data bus with additional burst mode operation.
- A branch target buffer predicts which way execution will branch and executes the opcodes without delay.
- Two instructions can be operated on at the same time in parallel pipelines.
- Built-In multiplier, divider and adder units so floating point instructions can be executed in one clock cycle.
- Performance is 100 Million instructions per Second at 60MHz operation.

The Pentium surpasses the 486 in speed and power by using 256-bit data paths, pipelined processing that lets operations in all components of the microprocessor happen at once, and instruction processing split into dual arithmetic logic units.

The processor is a 73 pin package arranged in a 21 x 21 pin grid array with an additional pin for keying. The chip incorporates 3.1 million transistors through the use of 0.8 micron CMOS technology. The processor uses a 32 bit bus, but its external data bus to memory is 64 bits wide. A burst mode permits 256 bit data segments to be read by the data cache in a single bus cycle, which produces a maximum memory transfer rate of 528 million bits per Second.

NOTE: To obtain maximum performance programs are required to be re-compiled or re-assembled using a “Pentium” compiler/assembler. Otherwise the performance upgrade from 486 is less than inspiring.

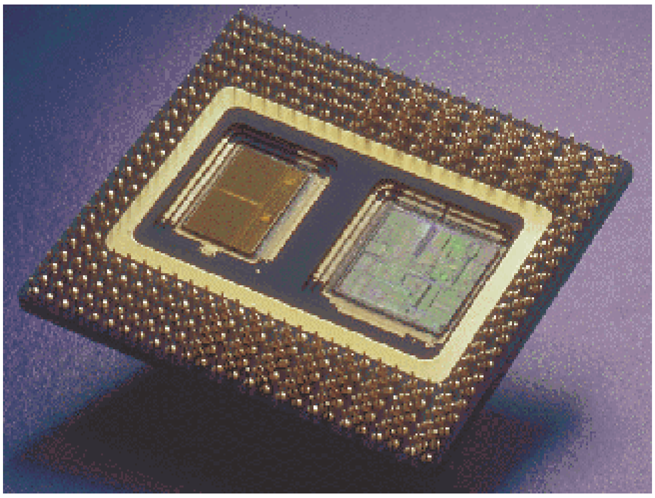
The P6 now “Pentium Pro”

The Pentium Pro has been constructed with the level 2 cache in the chip carrier with the processor. It has complete code compatibility with previous Intel 80x86 processors. It delivers superior performance through an innovation called Dynamic Execution. It also provides support for enhanced data integrity and reliability features: ECC (Error Checking and Correction), Fault Analysis & Recovery, and Functional Redundancy Checking.

The Pentium Pro integrates about 5.5 million transistors on the chip, compared to approximately 3.1 million transistors on the Pentium processor. It operates at clock speeds of 166MHz up to 200MHz.

It initially was produced on the same high volume 0.6 micron process currently used for the 90 & 100 MHz versions of the Pentium processor, and was then moved to a 0.35 micron process.

It is intended for use as in a multiprocessor machine that will deliver performance that will scale up to 1000 MIPS with four processors. It is of course fully backward code compatible, however it has been noted as not running 16bit applications very well. Its 32bit performance is exemplary and is suited to NT, Windows 95 as well as Unix platforms.



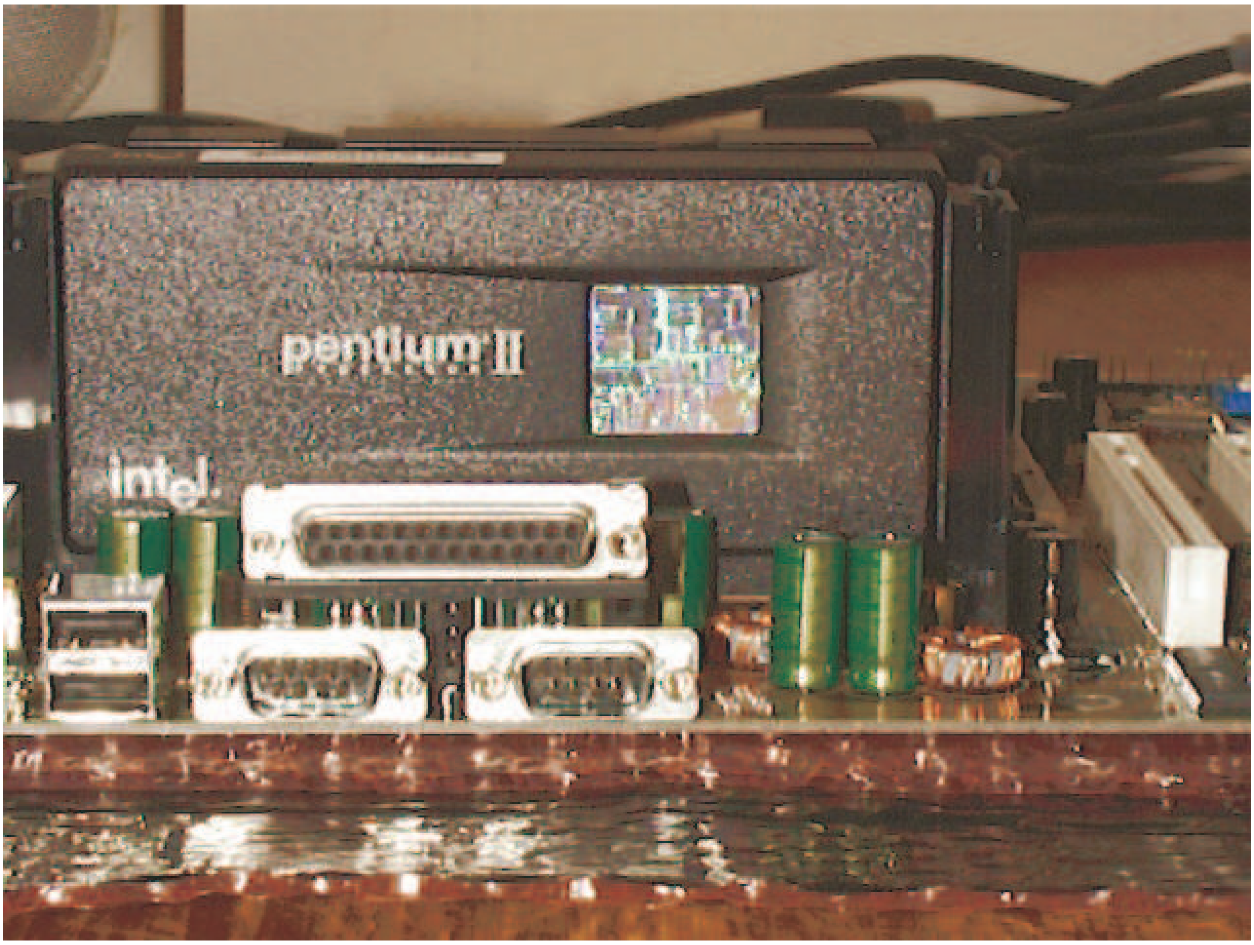
Klamath and MMX processors

The latest in a long line of processors, the multimedia/graphically orientated processors, Pentium MMX, which speed up graphical operations was released in Jan 1997. Three versions were produced, 166MHz, 200MHz and 233MHz. The P166MMX outperforms the standard Pentium by about 10%. With the Pentium being obsoleted in June/July 97 this processor replaced the standard Pentium as the ‘entry level’ processor. It didn’t last long, on the 21st of October 1997 it ceased production leaving the pc buying public to make a jump in cost to the Pentium II. Stocks of Pentium MMX cpu’s are expected to run out around June 1998.

AMD and Cyrix have competing products, though neither has really made it in South Africa. In Europe AMD have made serious inroads into Intel’s market with the K6.

The Pentium II

This is really a Pentium Pro with corrections applied to the 16bit performance and the MMX instructions added to the design. The socket design is quite different from the previous chips and uses a patented and proprietary socket. This stops any competing processor being plug-compatible. The processing performance is considerably improved over the previous standard Pentiums but leaves the basic PC design 'I/O bound' as the peripherals do not match the processing speed.



The Pentium II has clock speeds of 233MHz to 450MHz, though Intel has shown a version running at 700MHz. The original version came with a large heatsink and dual fans which made the cpu unit heavy and unstable when mounted normally in a motherboard. Newer versions are being produced with single fans and circuitry to check for fan failure.

The Pentium II has also changed the design of the motherboard from the XT/Baby AT design. The new ATX design seen above has the serial and parallel ports as well as USB and PS/2 connections cluster mounted on the motherboard. The motherboard also has power down and power off functions built-in. This all means that the case and power supply is a new design as well allowing for these functions.

With the machine being 'I/O bound', Intel have brought out a high speed graphics attachment bus called AGP. More details further on in the display peripherals section.

*** More ....

Multiprocessor computers

Multiprocessor machines can be purchased, consisting of Pentium type processors. This type of pc may well be the future standard for “power” boxes. It is already becoming the ‘norm’ for file server and database server type machines.

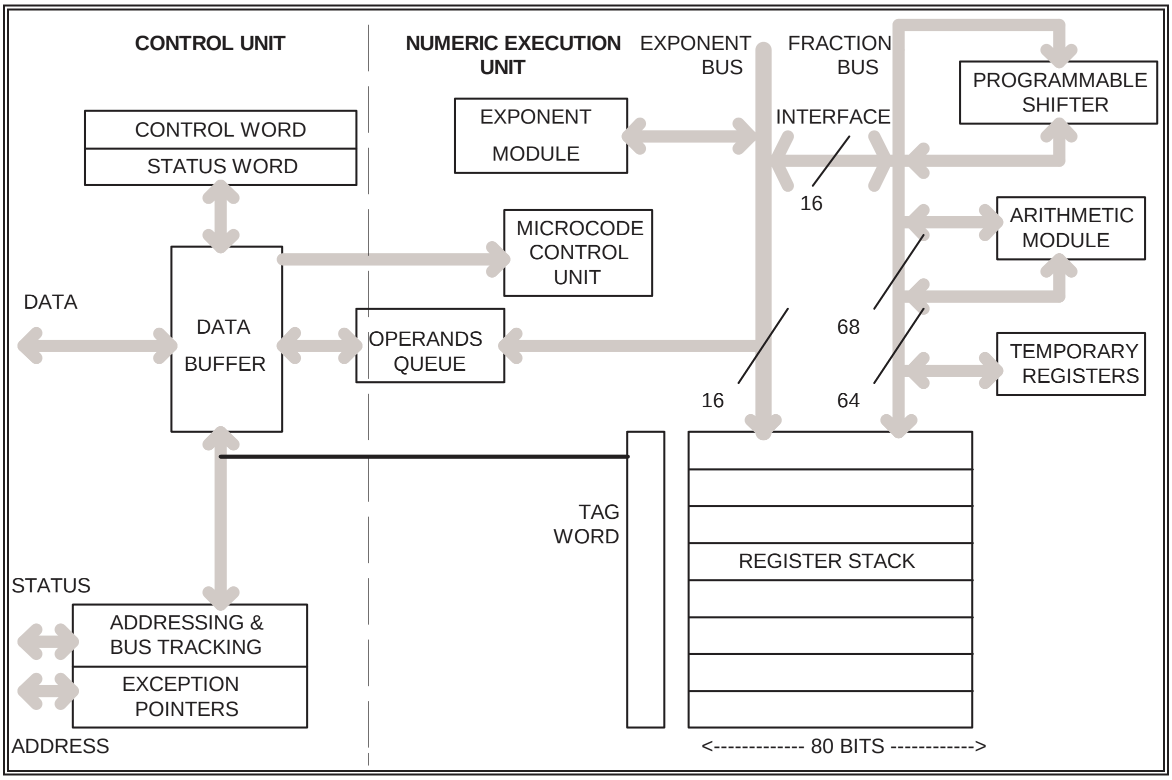
8087, 80287 and 80387 Floating point coprocessors

The math’s coprocessor was designed to provide a library of fast floating point routines. The processor requires no additional hardware I/O or DMA transfers. It is accessed using escape sequence codes in software. Most modern compilers have this support built in and some automatically detect the presence of the coprocessor.

The coprocessor has its own set of registers and instructions for access appear in the 8088/6 processor instruction set. There are eight data registers of eighty bits each, a status and control register and an instruction and operand pointer. The registers are also organised as a stack and register addresses are relative to the top of the stack.

Programming the coprocessor is straightforward as the processor has only sixty-nine instructions. The additional processing speed attributed to the use of a coprocessor can be quite amazing. A typical calculation can be reduced in time by thirty times or more by the addition of a coprocessor.

8087 Coprocessor block diagram



Support Microcontrollers

The purpose of the support microcontrollers is to assist the microprocessor in the control of the microcomputer system. The 8088 and 8086 rely on separate microcontrollers mounted on the motherboard. The more complex 80286, 80386 and 80486 have many of the support functions built into the microprocessor chips.

DMA (Direct Memory Access) Microcontroller

DMA is a concept whereby the CPU is not involved in the transfer of data from a peripheral device to memory, vice versa or memory to memory. The microcontroller usually used, in 8086/8 based systems, is an 8237. The 8237 has four DMA channels that can be individually programmed. Two of these chips were designed into the AT design to provide seven DMA channels 0 to 7. Channel 0 has the highest priority and 7 the lowest. Channels 0 to 3 perform 8 bit transfers and channels 5 to 7 can perform 16 bit transfers in the AT design. Channel 4 is used by the motherboard to cascade channels 0 to 3 to the processor. Channels 0, 3, 5, 6 and 7 are reserved, 1 is for SDLC and 2 is for disk use.

Clock Generator and Timing Crystals

All microcomputer systems require an accurate timing signal, to synchronise the various devices. A timing signal is generated by a quartz crystal oscillator. This signal is made into TTL compatible square waves by a Clock Generator chip. Timing signals of lower frequencies are generated by divider circuits internal to the clock generator chip.

In the case of the 8088 based microcomputer systems, the crystal produces a frequency of 14.31818 MHz. This in turn drives an oscillator circuit that supplies the 8284 Clock Generator. The original frequency is divided by 3, to provide a timing signal of 4.77 MHz, for use by the 8088. The 4.77 MHz signal is divided again by 4 to produce a 1.19 MHz timing signal for use by the timer counters.

The 8284 also handles a logic signal for use by the microprocessor, the Power Good signal. On IBM power supplies this signal is generated when all power levels are stable. The power good signal is used to reset the microprocessor, and start its normal operation.

PIT (Programmable Interval Timer)

This timer/counter microchip is an 8253, and has three 16 bit timer counters attached to the microcomputers bus. The PIT generates an interrupt signal that is used to keep the time of day clock and to operate the speaker. The 8253 also handles the triggering of the DMA microcontroller to start the memory refresh cycle.

Programmable Peripheral Interface (PPI)

An 8255 microcontroller was used for the PPI in the XT. The 8255 is a very versatile device in that it provides 24 programmable input/output lines. The PPI is used for several purposes, these are:

Reading the value of the motherboard configuration switches.
Interfacing with the keyboard.
Providing the speaker output control.
Controlling the NMI enable circuitry.
Supplying control lines for the 8253.

The 8255 functions have not been incorporated into any microprocessor design and was not used in the AT and later motherboard designs.

The Keyboard interface controller

On the AT/ISA motherboards this is generally a dedicated microcontroller based on an Intel 8042. The chip is programmed to respond to serial data from the keyboard and pass the data to the main cpu via the keyboard interrupt. This controller chip is also responsible for controlling the A20 address line, which allows access to extended memory.

Interrupt Controller

This microcontroller is an 8259. Most microcomputers are driven by interrupts. In the case of the 8088, only two interrupt connections/pins are available, the Non Maskable Interrupt and the Maskable Interrupt. The interrupt controller intercepts interrupt signals from all the devices, attached to the microcomputer system bus, allocates a priority to the interrupt signal and passes on the interrupt to the microprocessor. The 8259 enables 8 extra interrupt levels to be used on 8088 based microcomputer systems. Two cascaded 8259's are used in 80286 machines to provide 15 extra interrupt levels.

The Floppy disk drive controller

This microcontroller chip was originally an Intel 8272 floppy controller and was later upgraded to a Zilog 765 or NEC chip. The controller is designed to support high density disk drives up to 1.44MB and low density drives down to 160k single sided floppy disks.

The hard disk controller

The AT/ISA design used a hard disk controller adapter card based on a Western Digital design. The later cards used a WD1006 chip set which allowed for an interleave of 1:1 on the hard disk and had track buffering. This “Standard” design has been used as the basis for most IDE controller interfaces and is the required standard for use with Windows 32 bit disk access. Performance originally was around 240k Bytes per Second rising to 500k Bytes per Second with the WD1006.

Even though most of the functions, handled by the various support microcontrollers, have been incorporated into the 80286 and above, many microcomputer systems still use these microcontrollers for hardware compatibility.

Wait state

The main consideration in the manufacture of a microcomputer system, is the matching of speed. Many systems do not match components correctly, therefore compromising the efficiency of the microcomputer system. The term Wait State describes the number of cycles the microprocessor is inactive waiting for a reply from another device. The device can be memory or an input/output device. Thus the overall performance of a microcomputer system relies on the speed of the slowest device. The AT had one wait state inserted on memory access, which allowed the use of 150 nano Second ram chips.

Memory, memory addressing & structure

Memory

Memory forms an important part of any computer system, as it is used to temporarily store data and programs that are being utilised by the microprocessor. Two types of memory are used these are ROM and RAM.

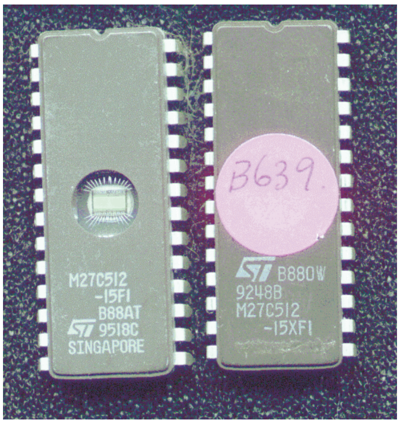
ROM (Read Only Memory)

ROM contains programs and data that is required by the microcomputer at start up. The contents of ROM cannot be changed, under normal conditions. The programming of ROM is done at the manufacturing stage. ROM is split into further types, these are:

PROM	Programmable Read Only Memory
EPROM	Erasable Programmable Read Only Memory
EEPROM	Electrically Erasable Programmable Read Only Memory

A PROM is basically ROM that has not been programmed. The programming is done by the user, by using a special PROM programming unit. Once programmed the PROM cannot be reprogrammed.

The EPROM may be reprogrammed, in the same way as the PROM, but only after the contents have been erased. This is done by subjecting the microchip to ultra violet light for 40 to 45 minutes.



Eproms

EEPROM allows the reprogramming to be done without having to erase the previous memory contents.

ROM chips on the motherboard, may contain the BIOS (Basic input/output System), the initialisation routines for the microcomputer system, diagnostic routines, the routine for handling the diskette boot procedures, system configuration and BASIC (Beginners All-purpose Symbolic Instruction Code) interpreter. In most cases the POST (Power On Self Test) is also included.

Other uses for ROM on the motherboard are to store programs that run directly on power-up, such as dedicated word processing and proprietary control software.

ROM has a disadvantage in that the memory access is generally slow, in comparison to RAM. This has lead to some manufacturers programming the BIOS to copy the ROM into RAM on startup, providing an improvement in performance. This “SHADOW” RAM is however lost to the operating system as a result.

Flash eprom technology has already been used for network adapters. It may soon be the norm for motherboards allowing BIOS upgrade from a disk. Some Pentium motherboards already have this as an additional feature.

RAM (Random Access Memory)

RAM is split into 2 further types; these are SRAM (Static Random Access Memory) and DRAM (Dynamic Random Access Memory). RAM generally has faster access than ROM, however RAM has the disadvantage of being volatile. That is: when power is removed from static or dynamic RAM, the contents disappear.

Dynamic RAM uses small capacitors mounted on a grid of lines. The capacitors hold a charge that signifies a binary 1. When RAM is read, the capacitors are automatically recharged. DRAM capacitors discharge fairly quickly (4mSec), so a refresh cycle is required to automatically recharge the small capacitors. The refresh is really a memory read on a block of cells, and this is done transparently by the memory refresh hardware many times per second.

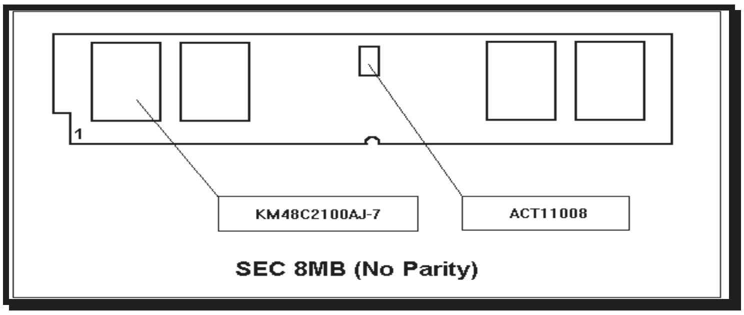
Static ram uses transistor flip-flops that can store data for long periods of time, as long as power is supplied to the RAM chip. The access time for SRAM can be extremely fast. The CMOS (Complementary Metal Oxide Semiconductor) RAM that is used to store system information on 80286 systems is static ram. The content of the CMOS RAM is variable according to system set-up. The SRAM is used because the memory chip only requires a low level of power, and this can be supplied by a battery pack.

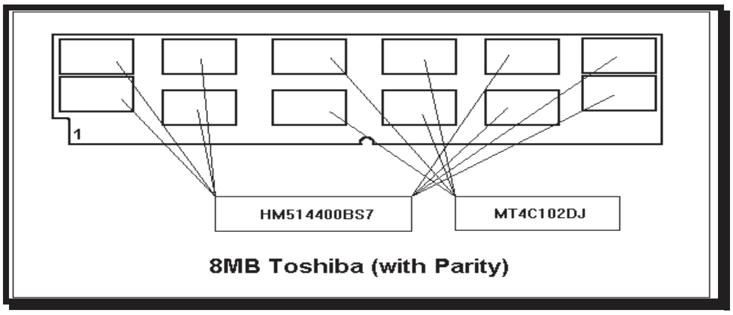
NOTE: Both ROM and RAM memory chips are sensitive to static electricity and to protect these components they should be carried in anti static tubes or containers.

Parity - Hardware error detection

In any sequence of bits parity can be calculated on those bits according to even or odd parity. In the PC this is done by a hardware chip and stored into the ninth memory location. Should the bit generated by the parity check chip not correspond to the bit stored in the ninth bit, a parity alarm signal is generated. The parity alarm signal is connected to the Non Maskable Interrupt line and produces the message "system parity error : system halted".

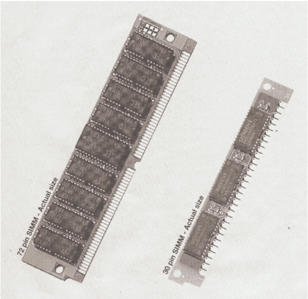
Unfortunately some SIMM manufacturers make SIMMs with and without parity. This leads to problems, as you cannot mix the two types on the motherboard. You may disable parity checking if you have a late model motherboard, allowing both to be used. How do you tell which SIMM has parity or hasn’t? Count the number of chips on the SIMM. If a SIMM has 9 chips or a multiple of 9 then it will have parity.

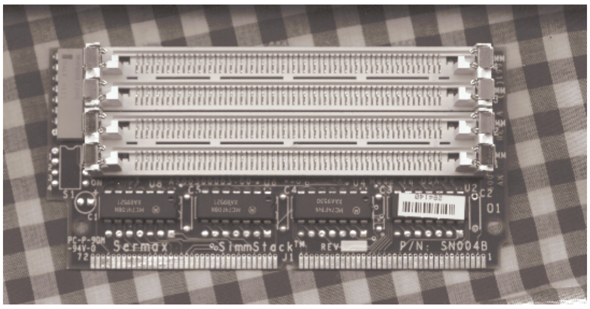




Mix-n-Match its not

Mixing SIMMS of different manufacturers is also a recipe for disaster, as access time differences will produce parity errors or worse, unexplained crashes. Similarly different construction SIMMs will produce errors. The OLD SIMMs would have nine 1M Bit chips on a SIMM, whereas the new variety have two 4 bit chips and a 1 bit parity chip. These cannot be mixed reliably in a machine.





72 pin and 30 pin SIMMs

A ‘SIMM-VERTER’ allows you to use up those 30 pin SIMMs

Memory Addressing

Each memory or port component in the PC has a unique address. A given I/O adapter may have both an I/O address and memory addresses. The memory addresses are for the memory on that card. For example, a VGA card has a given I/O address for the 6845 video controller, and the RAM on the card, used for the text and graphics, has a memory address.

The lowest memory addresses, below 640k are for motherboard RAM and are used to store the interrupt vector table, the BIOS data area, the DOS working area and the user’s programs.

An 8086 processor has 20 address lines resulting in the addressing of 1,048,576 bytes of memory. The 8086 having only 16 bit registers has to add two registers together to obtain the required 20 bits. This is done with a segment register and an offset register. The address in the offset register is offset 4 bits from the physical address and the two values are then added to obtain the physical address. This 20 bit address is then placed onto the address bus.

The 80386 and 80486 has 32 address lines and can address 4,294,967,296 bytes of memory in its virtual protected or native mode.

NOTE: Its called ‘real mode’ because the address specified in the segment and offset, actually refer to a physical address in memory.

CPU addressing - how it works. Byte addressing and Word addressing.

The 8088 had 16 bit registers and an external 8 bit data bus. The 8086, 80186 and 80286 all had 16 bit registers as well as 16 bit external data buses. This meant that these processors had an advantage of being able to transfer 16 bits at a time over the data bus, rather than two 8 bit bytes. This speeds up the retrieval of data from memory. With the 80386 and up this becomes 4 bytes per memory access. This all supposes that the data is aligned on either a word boundary or double word boundary. If it isn’t the cpu will have to have two or four retrievals to obtain the data. Compilers and assemblers can be instructed to place the data on word or double word boundaries, thus making programs perform better when a higher processor is used.

Intel Notation

The original 8080 (not 8088) processor stored its 16 bit words with the high byte in the higher memory location. This method has carried on throughout the Intel range of processors. Other manufacturers of processors, such as Motorola use the lower memory location for the highest byte. This leads to interesting data conversion problems between systems of different processors. If a 16 bit number is read from memory you will have to transpose the right byte value and the left byte to obtain the correct value. A 32 bit number is stored in a similar manner, with the highest byte of the four bytes, stored highest in memory.

Example: The pointers stored in the interrupt table are stored like this :-

[OFFSET]	:	[SEGMENT]
[low byte][high byte]	:	[low byte][high byte]

Looking at Memory addresses with Debug

Debug has been provided with DOS since version 1.0. It provides a system monitor program which can be used for ‘debugging’ programs. For a detailed manual see the document DEBUG.txt.

Debug commands and functions include:

Compare 2 blocks of a system memory
Display contents of system memory
Display and modify the content of system memory
Fill the memory with data or data from a string
Go and execute the program from an address.
Specify break points in a program and display system status
Load data from a diskette into memory
Move the contents of memory from one location to another.
Send data to an I/O port.
Display the contents for the processor registers and flags.

Search system memory for data specified in a string.
Trace the execution of a program and display processor CPU status.
Disassemble an assembly language program.
Write a block of system memory to diskette.

Debug is a command driven program and as can quickly be seen, it was designed for programmers. The hyphen "-" is used as the prompt. To obtain some help, type a ? and press enter.

To examine a block of memory start DEBUG from the DOS prompt and after the "-" appears, type in D40:0 and press Enter. This will give a "D"ump of the memory at segment 40 hexadecimal in both hexadecimal and ASCII.

A memory address is a physical location within the total memory, at which data can be stored. Under DOS, memory is divided into segments each of which comprises 64 k of contiguous bytes of memory. An individual memory address within a segment is called an offset. The first byte in a segment is offset 0 the second is offset 1 etc. Individual memory addresses are identified by their segment address and offset. Debug knows nothing about extended memory and cannot be used to examine memory above 1MB.

The PC’s Memory Structure

The XT & AT System memory map

Decimal (k = 1024)	Hexadecimal (absolute physical)	Function
960k	FFFFF to F0000	64k system ROM BIOS, Power On Self-Test and possibly BASIC.
816k	EFFFF to CC000	Reserved for ROM attachments on adapters.
800k	C0000	VGA ROM bios attachment
736k	B8000	CGA adapter display memory.
704k	B0000	Monochrome & Hercules adapter display memory.
640k	A0000	EGA / VGA adapter graphics display memory.
	9FFFF	DOS & USER memory (Read / Write memory.)
1024 0000	004FF 00400 00000	BIOS DATA AREA Interrupt vector table

Port & I/O Addressing

Port addressing is done by the processor in exactly the same manner as memory addressing. The difference being that the processor asserts the IOREQ control line. The 8088 can address 65536 individual ports using the lowest 16 bits of the address bus to select the required port. The IBM XT designers however provided only for 1024 ports and grouped them in blocks of eight. This was changed in the AT design to the full 64k ports and grouped in blocks of 32. Some adapters designed for the XT do not decode the port address fully and can be found at several addresses on the 64k map.

XT (8088) I/O Address map

Address range (Hexadecimal)	Usage
3F8-3FF	Asynchronous Communications (Com1)
3F0-3F7	Diskette controller
3E0-3E7	Reserved
3D0-3DF	Colour Graphic display adapter. (CGA)
3C0-3CF	Reserved
3B0-3AF	IBM Monochrome Display adapter.
3A0-3AF	Reserved
380-38F	SDLC Communications adapter
378-37F	Parallel Printer
320-32F	Fixed Disk controller
300-31F	Prototype Card (USER I/O)
2F8-2FF	Asynchronous, Communications (Com2)
2F0-2F7	Reserved
278-27F	Reserved
220-24F	Reserved
210-217	Expansion Unit
200-20F	Game Control & Tape streamer hardware.

0Ex	Reserved
0Cx	Reserved
0Ax	NMI Mask Register (x indicates don't care) (bit 7)
080-083	DMA Page Registers
060-063	PPI 8255
040-043	Timer 8253
020-021	Interrupt 8259
000-00F	DMA Chip 8237

NOTE: When powering-up the NMI (Non Maskable Interrupt) for the 8088 is switched off.
The switch (mask) can be set and reset as follows:-
Set mask: write hex 80 to I/O Address hex A0 (enable NMI)
Clear mask: write hex 00 to I/O Address hex A0 (disable NMI)

AT & ISA I/O Address Map

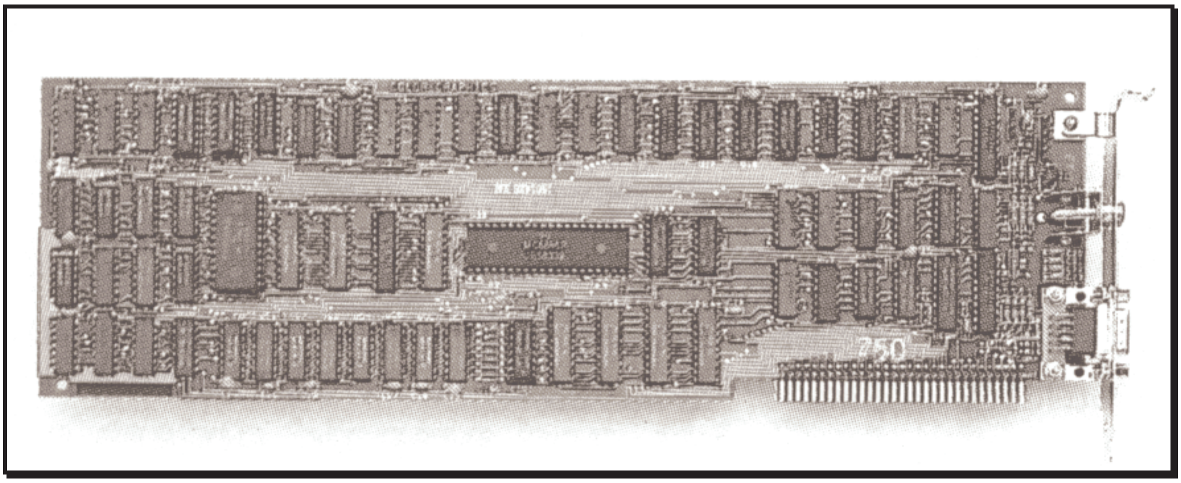
Address range (Hexadecimal)	Usage
	space for other adapters ?
378-37F	Parallel Printer port 1
320-32F	not used
300-31F	Prototype Card (USER I/O) or Ethernet adapter card.
2F8-2FF	Asynchronous, Communications (Com2)
2F0-2F7	Reserved
278-27F	Parallel printer port 2
220-24F	Reserved
210-217	Reserved
20C-20D	Reserved
200-207	Game Control & Tape streamer hardware.
1F0-1F8	Hard disk controller.
0F0-0FF	Math coprocessor I/O
0C0-0DF	DMA Controller no 2 8237
0A0-0BF	Interrupt controller 8259 Slave
080-09F	DMA Page Registers
060-06F	8042 Keyboard controller
040-05F	Timer 8254
020-03F	Interrupt controller 8259 Master
000-00F	DMA Controller 8237

Peripheral Devices

Display adapters

[Speed of graphics, why 16 bit has fewer wait states, why mono/cga is poor for graphics speed. Why local bus is really good for graphics.]

Color Graphics Adapter *

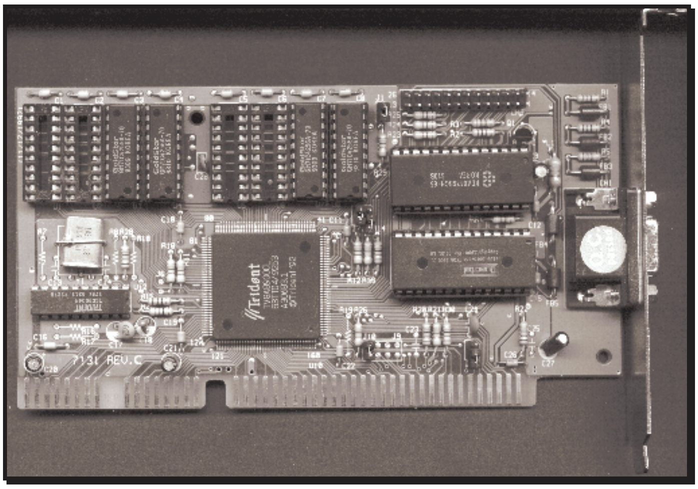


[* Yes I know it is spelt wrong!]

Monochrome and CGA adapter cards worked by making the processor have complete access to the display memory. These cards relied upon the processor being patient as the CGA card would give “SNOW” if it was accessed during a normal scan display time. The processor therefore had to wait for a flyback time to access the memory, several microseconds later. The BIOS was written to allow for this or alternatively the CGA adapter would be “switched off” during update. This gave rise to a blinking or flashing display which was awful to watch.

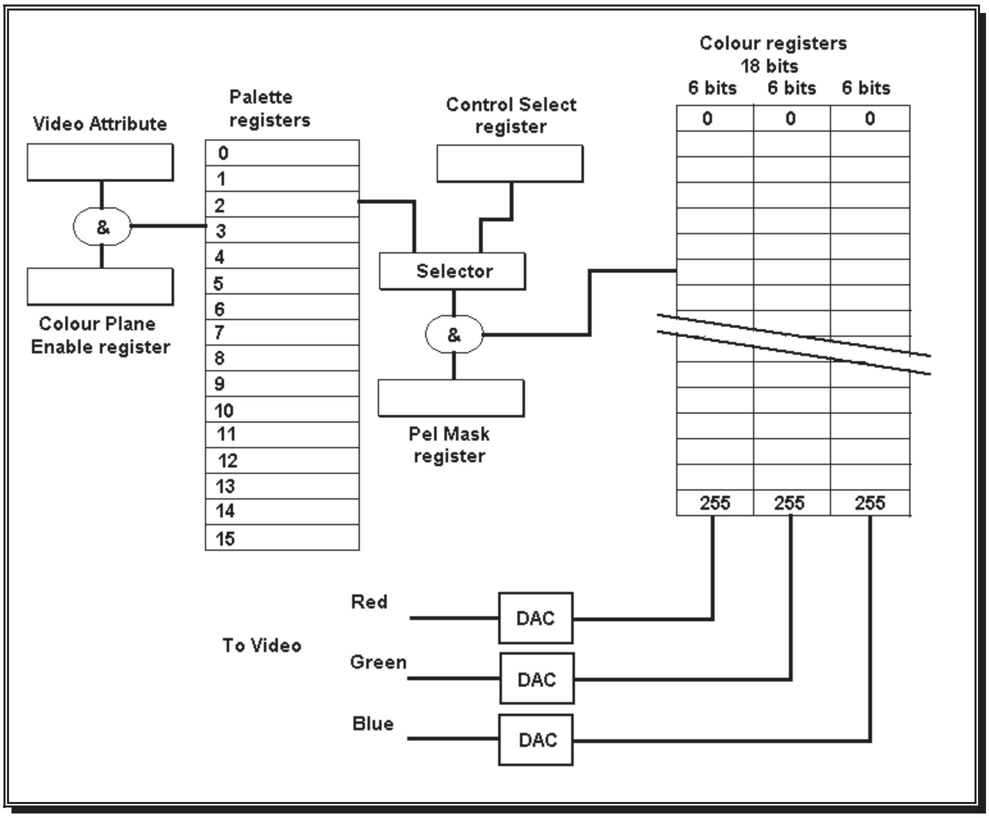
Later adapters, both Hercules and CGA clones, used the WAIT signal on the processor to stop the CPU from accessing the memory until it was available without causing “SNOW”. The EGA (Extended Graphics Adapter) had this wait state circuit built in and was snow free as a result.

The VGA adapter



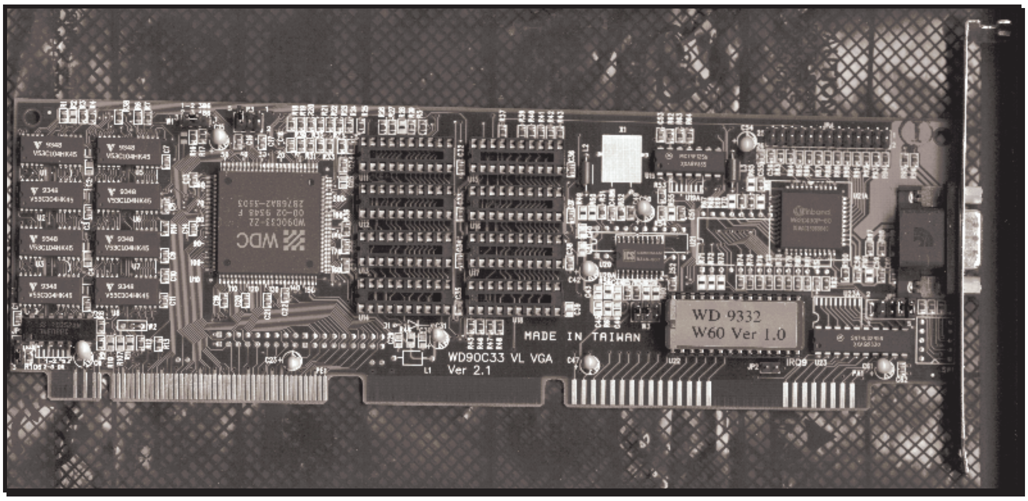
An improvement brought about by IBM for the PS/2 range of computer, the VGA adapter has stuck with the industry. The successor the XGA has not caught on. This adapter is an ANALOGUE type of video display where previous display adapters were DIGITAL. There is a sort of perverse logic to this as the Digital to Analogue converters have improved to 24 bit from the initial 6/8 bit converters allowing millions of shades of the primary colours. This renders picture quality approaching photographic quality. It is certainly far better than television quality.

Original VGA adapter - allows 18 bit colour registers, 6 bits per primary colour giving 262,144 total colours.



VGA eliminated most of the wait states used in previous cards and allowed the processor the full 16 bit bus access to the screen display memory. This improved the performance dramatically from typically a few hundred characters per millisecond to 700 characters per milliSecond for a 256k VGA card. The performance improves with amount of memory as well as cpu speed and a 512k adapter can give up to 2,900 characters per milliSecond on an ISA bus.

Local Bus VGA adapters



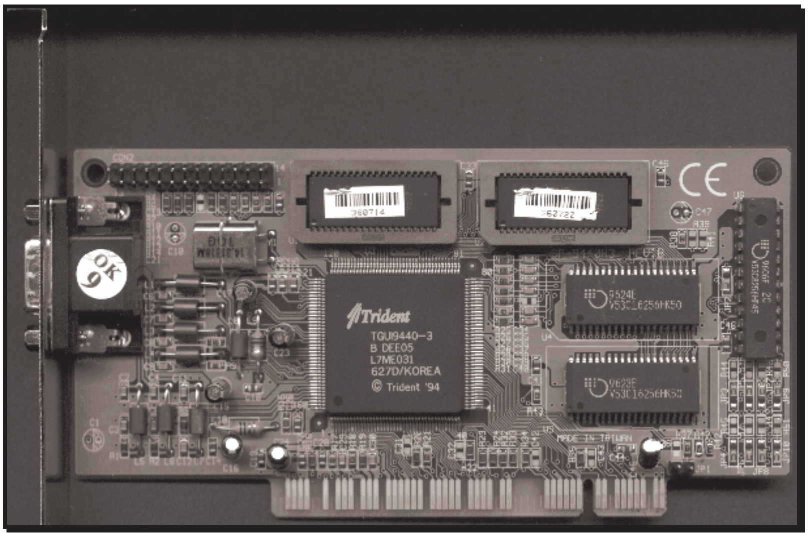
Tests carried out by us indicated that a four to five times improvement could be achieved with a local bus VGA card over a standard 512k VGA ISA card. The fastest we tried, the WD90C33 VL BUS VGA card gave nearly 10,000 characters per milliSecond compared to 2,500 characters per milliSecond for a Tseng Labs or Trident VGA adapter card. The price difference is minimal if you compare costs for a 1MB Trident VGA adapter against this card which comes supplied with 1MB.

Some adapter cards provide extra processing capability in the drawing of graphics shapes and as a result provide extra speed in operation. These are generally referred to as “Accelerator” video adapters.

An example of an accelerator card was recently tested by us, the Chips & Technologies “Wingine” provides a number of performance enhancing routines in its bios and caches fonts as well. It is a local bus adapter and costs no more than a standard VGA 1MB adapter. Its performance measures around 16000 characters/mSecond.

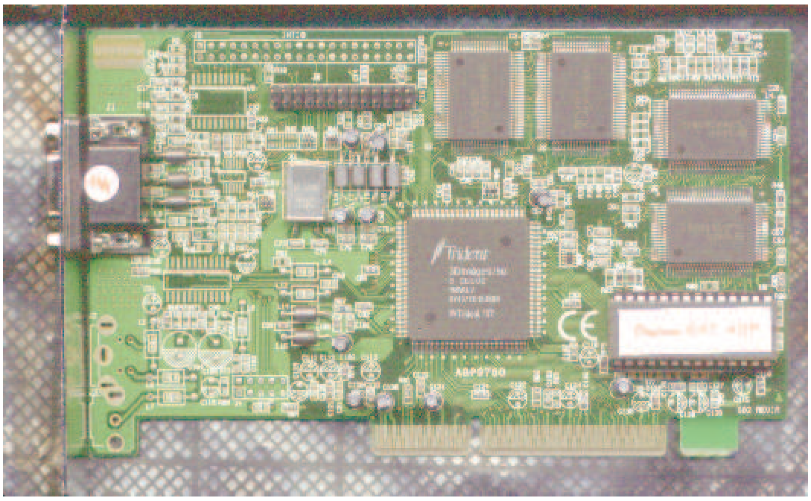
PCI VGA cards

Most PCI VGA adapters have moderate performance, lower than the VL-BUS types. However these are now the standard. For top notch performance use a Diamond Stealth 64 card.



PCI VGA adapter

Note the card is the other way around from an ISA card. This is because the PCI slots face the opposite way to stop incorrect insertion of ISA cards. It also loses one available slot (ISA or PCI) as the slots that are together cannot both have cards in them.



Trident AGP card. Note the staggered connections on the connector.

Accelerated Graphics Port - AGP

With the Pentium II comes a dual bus operation that puts the cache memory connected to the cpu via a 1G bits per second bus and a front side bus that runs at 528M bps. The PCI bus is connected via a chip set to the front side bus. This bus in most machines has to contend with a vast amount of data traffic going to the display adapter especially when displaying video.

By making the front side bus switched Intel have provided a 528Mbps connection to the main memory, a 528Mbps connection to the AGP port and a 132Mbps connection to the existing PCI bus. Keeping the high volume data traffic off the PCI bus will improve the peripherals speed of operation.

Parallel and Serial communications

Voltage levels

Signalling between equipment is generally done at low voltages and currents to ensure safety of test personnel. In the next section I discuss the RS232C connection between equipment and in the section after, the printer port connection.

Parallel transfer of data is generally done at least 8 bits at a time and sometimes 16 bits at a time. A single line extra is used to signal the receiving device that the data is valid. This technique is used to transfer data to and from the cpu as well as to peripheral devices. It is generally faster than serial communication of data. It is used by the highly successful LapLink and other file transfer programs to transfer data from one pc to another.

Serial data transfer is carried out one bit at a time, which means slower data throughput than parallel transfer. It also means more complicated hardware requirements, usually a UART or USART.

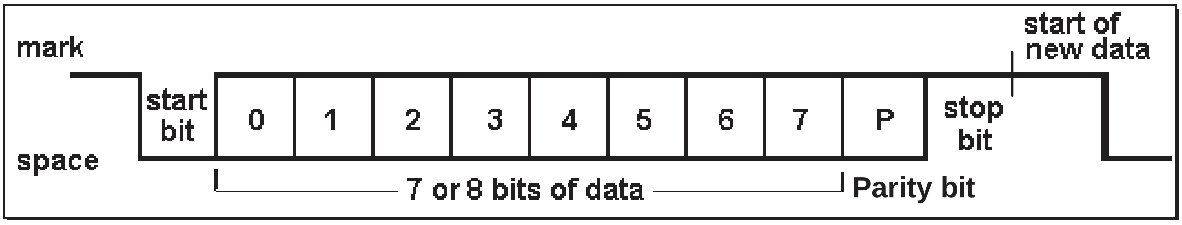
Serial data communications and UARTS

Serial data is transferred one bit at a time. Morse code is a form of serial data. Morse was superseded by 5 bit teleprinter codes commonly called Baudot. In America they will say Murray code. Five-bit code was and still is being used day in day out to transfer textual information from one side of the world to the other. Send a telegram to someone and the post office will use a teleprinter to transfer the message using five-bit Baudot code.

In more recent times it was more desirable to send more detailed text, using lower case as well as capitals. ASCII the standard character definition has become the order of the day. ASCII is truly a seven bit code, allowing for 128 possible characters. Extensions to ASCII were introduced by IBM in the PC to allow some line characters as well as some accented characters.

ANSI the American National Standards Institute are busy with a global version of ASCII, which may replace ASCII in computers of the future.

Serial data stream



The diagram represents a series of data bits against a period of time. The data rate is described as either "bits per second" or the "Baud" rate. Baud from Msr Baudot, the inventor of the five bit code. Obviously the rate description is dependent upon how many bits are used to contain a character. This could be 8 data bits with even parity and two stop bits, giving a total of twelve bits. So when quoting a data rate ensure that the number of data bits etc are stated.

The start bit is used in asynchronous data to signify the start of a character and allows the receiver to synchronise its local clock. The receiver will then wait a half bit period to see whether it is a true start bit. This provides some immunity to noise as most noise pulses will not last a half bit period. If the bit is still low, the receiver will then wait a full bit period and check for a high or low line voltage. If low it is a "0", if high it is a "1". The corresponding bit value is then placed in a bit of a register. The process is then repeated for all the expected bits of data.

The parity bit (P in the diagram on the previous page) provides some further error checking for the complete data byte and is set according to whether ODD or EVEN parity is being used. The receiving software will usually discard the byte received if the parity bit received is incorrect. The receiving chip will usually set a bit in a status register to tell the cpu that a bad parity data byte was received.

The stop bit is a hangover from the old days when the receiving device was a mechanical teleprinter which took a fair amount of time to "digest" the character received. It is not really needed when UART chip to UART communication is carried out. Unfortunately it is not an optional setting and is still transmitted. A UART is an acronym for Universal Asynchronous Receiver Transmitter, which is the chip used by the computer to send and receive data serially. The "norm" is one stop bit for computer to computer transfers and two stop bits for transmitting to a printer.

Seven or eight data bits may be transmitted and received and quite a few UARTS allow five or six bit operation as well. The UART used by IBM in the IBM PC was the 8250 type. This chip had an unfortunate habit of emitting spurious interrupt signals and the BIOS of the PC and XT was designed to allow for them. When IBM designed the AT 80286 machine, they used a later version of the chip called an 16450. This UART does not emit spurious interrupts and can be run at higher data rates.

NOTE: Quite a few XT serial adapter cards have been installed in AT's, much to annoyance of the service or support persons who have had to replace them later. If the UART is an 8250 in an AT, then assume that replacement may well cure problems with lost characters etc.

When the PS/2 was released by IBM in 1987, the serial adapter used a 16550 UART. This chip is a vastly improved and featured version of the 8250. Unfortunately IBM did not know about the manufacturing problems and large numbers of PS/2's could not make use of the features in the UART. The 16550A replaces the faulty chip and provides a FIFO buffer arrangement that dramatically improves the performance. Windows 3.1 can make use of the FIFO buffers in its serial drivers.

To receive data bytes at 1200 Baud, a cpu must read data from the UART every six milliseconds. Not an arduous task for an AT ! But at 9600 Baud, it must read a byte every 83 microseconds. With the cpu being used for more intensive processes, it does not have a lot of time to get characters from the UART and stuff them into a memory buffer. In a multitasking operating system the cpu may be too busy to read data from the UART. The FIFO buffers are internal to the 16550A and can hold 16 bytes of received and transmitted characters. This means that the cpu only has to read every 16 character periods apart. This reduces the load such that a 9600 Baud link will seem like a 600 Baud link to the cpu.

The 8250 register set

As I said above the 8250 UART uses registers to provide an interface to the cpu. These registers appear at certain addresses with the two most common being standard. Com1 or communications port 1, is at address 3F8 hexadecimal. Com2 is at 2F8 hex. This can be relied upon. Unfortunately Com3 and Com4 cannot. 3E8h, 2E8h and 2E0h have all been used in some designs.

NOTE: You cannot have all four ports operating in a machine with the interrupts assigned as standard. The PC design only assigned two interrupts, IRQ4 for Com1 and Com3, IRQ3 for Com2 and Com4. The design of the PC does not allow sharing or "daisy chaining" of interrupts. If you want more then use another unused IRQ line. If you use IRQ5 and IRQ7, Windows 3.x can be set to use these ports quite successfully. However you won’t be able to use these if your machine has a new ECP/EPP printer port.

RS232C

RS232C is the third revision of the Recommended Standard for connections between DTE and DCE issued by the Electronic Industries Association (EIA). DTE is an acronym for Data Terminal Equipment. Terminals usually fall into this category. DCE is an acronym for Data Communication Equipment. Modems and Line drivers fall into this category.

Asynchronous and synchronous communications use voltage levels of plus and minus 15 Volts with a minimum of +/- 3 Volts. A voltage less than +3 Volts or greater than -3 Volts is reckoned to be an indeterminate level. In reality levels down to +/- 1 Volts can be used in electrically "quiet" areas. The high voltages give RS232C an immunity to external electrical "noise". The quality of the cable used for the connection also plays a part in determining the performance. With lower capacitance cable allowing higher Baud rates over a greater distance.

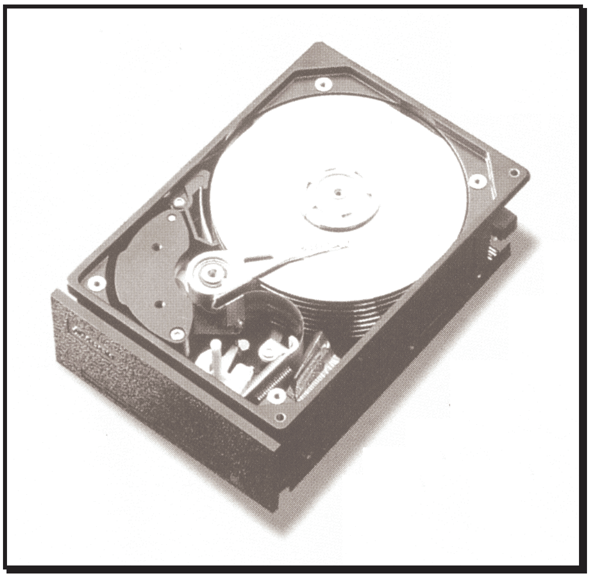
Centronics Interface

Parallel printer ports use the "Centronics" interface, which is based upon Transistor Transistor Logic (TTL) voltage levels. A binary "1" must be a voltage level higher than 2.4 Volts and a binary "0" is a level below 0.4 Volts. TTL chips sink or pull current to switch levels at high speeds.

SCSI

In the early eighties, two standards were introduced to control the flow of data between the cpu and its peripherals. In 1981 Adaptec was formed from pioneers who developed SCSI, the Small Computer Systems Interface. It was designed to provide high performance and was meant for fileserver and CAD workstations. A SCSI installation requires a controller adapter card inserted into the PC’s expansion bus. The controller can support up to seven other ‘controllers’ on a wired bus. Hard disk drives, CD Roms, scanners and tape streamers can all share this data bus. A SCSI hard disk on the AT bus gives performance of around 5Mega Bytes per Second. With Local Bus or EISA this can rise to 10.5MB per Second with SCSI-2 drives and as high as 40MB per Second with SCSI-FAST.

A typical SCSI hard disk



IDE - Integrated Drive Electronics

As an alternative interface IDE came late onto the scene. IDE or Integrated Drive Electronics is estimated to be in use on 90% of all pc's. IDE is "cost effective" and provides moderate performance dependent upon the bus interface. With the AT bus at 8MHz this can be 3 to 4 MB per Second. With Local Bus this can be as high as 9MB per second and could be increased further by the use of late model drives which support the newer modes. With newer drives it is possible to achieve up to 16MB/Sec using Ultra DMA.

CD ROM drives

Compact Disk Read Only Memory drives have become a very useful addition to the information hungry computer users. They provide high capacity storage of data with low cost of manufacture. Typically 680 Megabytes of data can be stored on a CD ROM. The capacity may change in the near future as a new standard is being worked on. DVD - Digital Video Disk is coming soon!

Fitting a CD Rom drive is usually straight forward as either a SCSI type interface card or an IDE type interface may be provided as part of the installation hardware. Data rates are not as high as hard disks, typically 180 to 200 k Bytes per second for double speed. Quad speed drives give 600-700k Bytes per second. Eight speed to Twelve speed and are now the "NORM". These should be SCSI connected so as to reduce the load on the CPU. Tests carried out have shown that an IDE 12 speed CD ROM can reduce overall performance to that of a quad speed! Also the practice of putting the CD drive in parallel with the hard disk is not acceptable. The reason for this is that the 32 bit access in Windows 3.x will revert to 8bit access using the BIOS, with a lowering of performance. This fortunately is not a problem in Windows 95 or NT4, as all the drivers are protected/enhanced mode 32 bit drivers.

IDE types can be fitted to a Sound Blaster card or a standard IDE interface. Most drives will play audio CD's as well.

Tape Streamers

For backup of data at low to medium cost nothing comes close to tape backup. Tape storage has been used in the industry since the early days. Very large amounts of data can be stored on tape. Tape has a limited useful life in storage (1 to 2 years) and is susceptible to magnetic fields and temperatures.

Tape capacities vary, for example, a data cartridge such as the Teac 60, which is similar to the popular Philips-type audiocassette, holds only 60 megabytes; the 20-year-old quarter-inch 3M data cartridge can hold up to 250 megabytes; and its downsized brother the DC2080 holds 80 megabytes. Though with data compression techniques these figures can be doubled or tripled.

A Quarter Inch Cartridge Tape (QIC) +/- 250MB

Helical scan recording

Rather than locating a set of fixed heads in a tape path, helical scan recording uses a set of three heads mounted on a drum that rotates at 1800 rpm. The tape is then wrapped around the rotating drum in the shape of a distorted 'C' (C wrap) at an angle of about 5 degrees to the face of the drum. The recorded data is no longer a longitudinal path across the tape, but a track that extends at an angle from the bottom of the tape to the top of the tape. The recording of helical tracks, or stripes, offers several distinct technical advantages that otherwise must be overcome in conventional longitudinal recording.

The actual recording pattern on tape utilises a series of very narrow stripes (0.000098 inch) written acutely across the tape. Each stripe (3 inches approx.) represents a single rotation of the head. With the head travelling at 1800 rpm, data density increases dramatically. Each track can hold about 130 k bits, and there are 820 tracks per linear inch of tape. A 346 foot tape can store over 2 Gigabytes of data.

A DAT Tape

Magneto-Optical Storage

These new drives designed by Fujitsu have the capability of storing 230MB or 640MB on a 3.5" disk. The disk is impervious to magnetic fields and needs the laser to heat the area on the disk to 200 degrees Celsius, so that the write head can store data. MO technology has certain advantages which overcome problems that other storage types. It has a non-contact read/write mechanism, the write head makes no contact with the media. So unlike tape, it has no wear and tear on the mechanism itself or on the head assembly. The storage life of the disk is estimated to be in excess of 40 years.

Scanners - Hand, Flat Bed

Hand scanners are low cost ways of generating graphics for information and programs. They generally require a steady hand and will scan approximately 4 inch wide strips, which can be joined together to give larger areas. Most late model hand scanners can provide up to 256 shades of grey as well as dithering options for photographic scanning. Colour hand scanners are available but cost more.

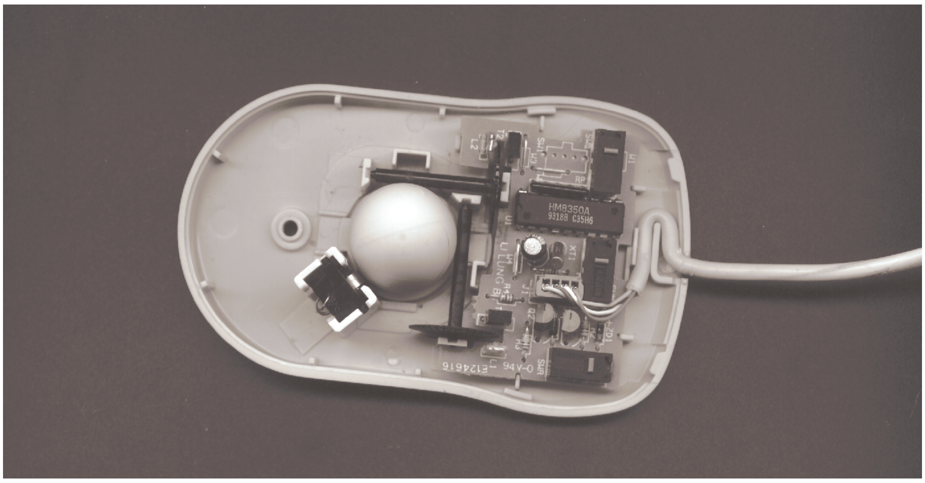
A4 flat bed scanners provide at least 300 dots per inch resolution and can scan colour images with true photographic rendering. The quality scanners usually come with Optical Character Recognition software or Photo retouching software.

Plotters, printers and other output devices

The industry standard plotter has for many years been the Hewlett Packard 7475 and most plotter manufacturers emulate this plotter. Connection is either using a parallel port or a standard serial port and drawings of up to A0 size can be produced in colour with little effort.

Mice, Digitisers and pads

The “desktop rodent” has become a part of our lives with the popularity of Windows, but it has other cousins which can be used to drive the system. Digitisers are used by artists and draftspersons to draw pictures and artwork. Care and feeding of the mouse includes regular cleaning sessions especially with users who smoke. To clean without attacking plastics, use surgical spirits on the ball and rollers which are mechanically coupled to the optical sensors.



The System Bus Structure

All the major components of a microcomputer system are connected to each other by means of physical links. These links are known as Busses. Three major busses are used, these are Data Bus, Address Bus and Control Bus. These three busses and several other lines form the System Board Bus which are; interrupt request lines, timing lines and DMA control lines.

Bus. Abbreviated form of "bus-bar" derived from "omnibus". A group of conductors carrying words in parallel (one bit per conductor) in either direction; usually common to several devices and identified by function, e.g. address bus.

Control Bus

The control bus is used to transport control signals from the microprocessor to devices that contain data. It includes signals to select memory or Input/Output, to read or write and so on. The size and complexity of the control bus depends on the microprocessor used.

Address Bus

This transport medium, is used to convey the addresses at which required data or instructions can be found. The address bus is normally used, by the microprocessor, to inform memory where the data it needs is. The number of lines that make up the address bus also depends on the microprocessor used.

Data Bus

Data is moved from device to device on this bus. The size of the data bus is 8 bits, or lines, wide for 8088 based microcomputers. The data bus will be 16 bits for the 80286 and 80386SX based systems. Normally the bus will be 32 bits wide for the 80386DX, 80486SX and 80486 based microcomputers. Some manufacturers of microcomputers restricted the 32 bit microprocessors, by using a 16 bit data bus (ISA bus). This had the effect of slowing down the effective speed of the microcomputer system.

Bus Architectures - Expansion buses

The five standards of bus structure are:

The XT 8 bit bus, which is now obsolete.
ISA (Industry Standard Architecture) also known as the AT Bus which is a 16 bit bus. Which is now also obsolete.
EISA (Extended Industry Standard Architecture) or 32 bit bus. No longer used.
Microchannel is the IBM 32 bit bus originally designed for the PS/2. No longer used.
The VESA Local Bus (VL-Bus). No longer used.
The PCI bus developed by INTEL as an alternative to VESA local bus.

NOTE: AGP is not a bus as such as it is only used for high performance graphics adapters.

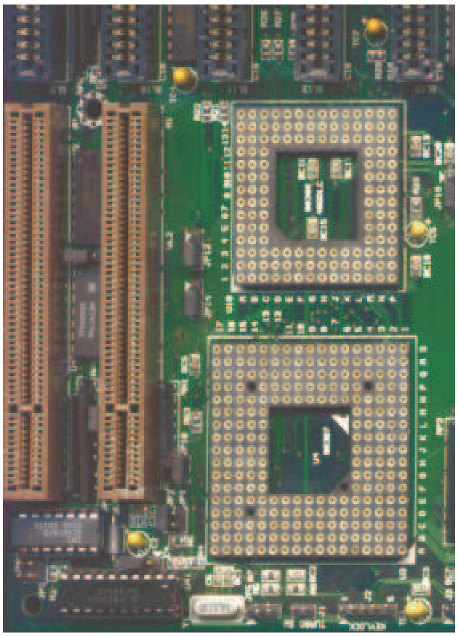
The Industry Standard Architecture (**ISA**) bus originally ran at the clock speed of the processor. Later motherboards attempted to keep this the same. With processors reaching clock speeds of 12MHz, the peripheral adapter cards, notably the hard disk controller card, started to behave erratically. The bus was then set to 8MHz for all expansion cards. This was to become a problem (bottleneck) later when Windows the graphical operating system, finally had general acceptance.

The **EISA** 32 bit bus structure does allow the use of 8 and 16 bit boards and allows devices to share interrupt request lines. Unfortunately the use of an 8 bit adapter will tend to degrade the system performance. EISA can perform 32 bit burst transfers at up to 33 Megabytes per second.

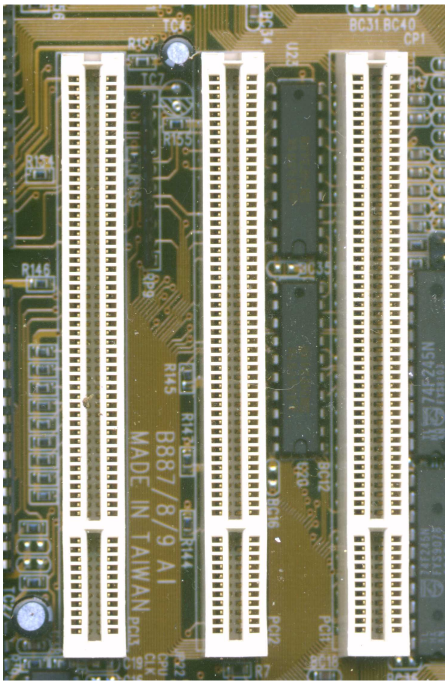
The IBM **Microchannel** bus structure does not allow the use of 8 or 16 bit boards. This keeps the system speed of the Microchannel microcomputers constant. Microchannel adapters can share interrupt request lines.

Towards the end of the 80386 era, we saw the arrival of a new bus architecture, called the **VESA** (VL-Bus) Local Bus. This was originally designed to speed up the VGA adapter interface and thereby speed up Windows operation. It is a 32 bit bus providing four to five times improvement in display speed. Unfortunately it is no longer in production as the industry as a whole went with 80486 or Pentium I,II,III and IV with the PCI bus.

PCI stands for Peripheral Connect Interface. This was originally unlikely to be used on general purpose machines as it was more expensive to implement, but since the adapter cards can be made lower in cost, it gained acceptance and has been used on virtually all motherboards since the 80486.



Vesa Local bus slots (brown) below the ISA slots on a 386/486 motherboard.



PCI Slots on a late model 80486 motherboard.

Interrupts

There are three general types of interrupts that can occur in a PC: hardware interrupts, software interrupts, and processor exceptions.

If the processor in your computer had to continually poll the various I/0 devices, it would not be very efficient at doing the real work you ask of it. So to maintain efficient use of the processor's time, computers use interrupts to handle asynchronous hardware events.

An interrupt signals the processor to perform some special operation. On the IBM PC, the interrupt vector table governs the choice of operation. This table is located in the first 1024 bytes of memory. Each entry in the table is an address that's 4 bytes long consisting of a segment and offset.

Here are the 8088/86 hardware interrupt levels:

Interrupt level	Usage
NMI	Motherboard RAM parity and I/O channel check, maths coprocessor.
IRQ0	System timer output channel 0.
IRQ1	Keyboard scan code interrupt.
IRQ2	Available for extra equipment.
IRQ3	Asynchronous communications (COM2) and SDLC (Synchronous Data Link Control) communications.
IRQ4	Asynchronous communication (COM 1) and SDLC communications.
IRQ5	Hard disk drive.
IRQ6	Floppy diskette drive.
IRQ7	Parallel printer.

Most of these interrupt levels were retained in the 16 and 32 bit microcomputers, but a further 8 are added and some are changed. These are the 16 and 32 bit interrupt levels:

Interrupt level	Usage
NMI	Motherboard RAM parity and I/O channel check. Not available
IRQ0	System timer output channel 0. Not available
IRQ1	Keyboard scan code interrupt. Not available
IRQ2	Cascade to IRQ9. Not available.
IRQ3	Asynchronous communication (COM2) and SDLC communications.
IRQ4	Asynchronous communication (COM1) and SDLC communications.
IRQ5	Parallel printer (LPT2). Especially ECP/EPP.
IRQ6	Floppy diskette drive controller. Not available.
IRQ7	Parallel printer (LPT1). Especially ECP/EPP.
IRQ8	Realtime clock. Not available.
IRQ9	Redirected IRQ2 and usually available.
IRQ10	Available for use.
IRQ11	Available for use might be used by SCSI controller.
IRQ12	Available for use though probably used by bus-mouse.
IRQ13	Maths Coprocessor. Most processors have FPU so - Not available.
IRQ14	Hard disk drive controller. Now IDE - Not available.
IRQ15	Secondary hard disk controller. Now EIDE - Not available.

When an interrupt occurs, the processor has to:

Save where it is in a program. It does this by ‘pushing’ its code segment and instruction pointer registers onto the ‘stack’. It also saves the Flags or Machine Status Word register as well. The stack is a small area in memory which is set aside for the processor to use as a scratch pad. The Flags registers was called that with the 8088/86 and was called the MSW with the 80286 up.

It must then acknowledge the interrupt. It does this by setting the INTACK line connecting the cpu to the 8259 master interrupt controller.

It then receives two bytes from the interrupt controller. The first is an INT instruction, the second has the number assigned to the hardware interrupt. This number does not relate to the interrupt request number (IRQ). For example IRQ0 translates to INT 8.

Having got the index into the table, the cpu picks up the values in the interrupt table and places these in the code segment and instruction pointer (CS:IP). On the next clock cycle it jumps to that address.

Hopefully at this point the cpu finds an ‘Interrupt Handler’, a piece of program that does what is necessary for the hardware device. If not - we have the cpu very confused, executing random data and program - usually hung.

At the end of the interrupt handler, the cpu ‘returns’ and in doing so, ‘pops’ the saved code segment and instruction pointer into the CS:IP registers. It continues with what it was doing before it was ‘interrupted’.

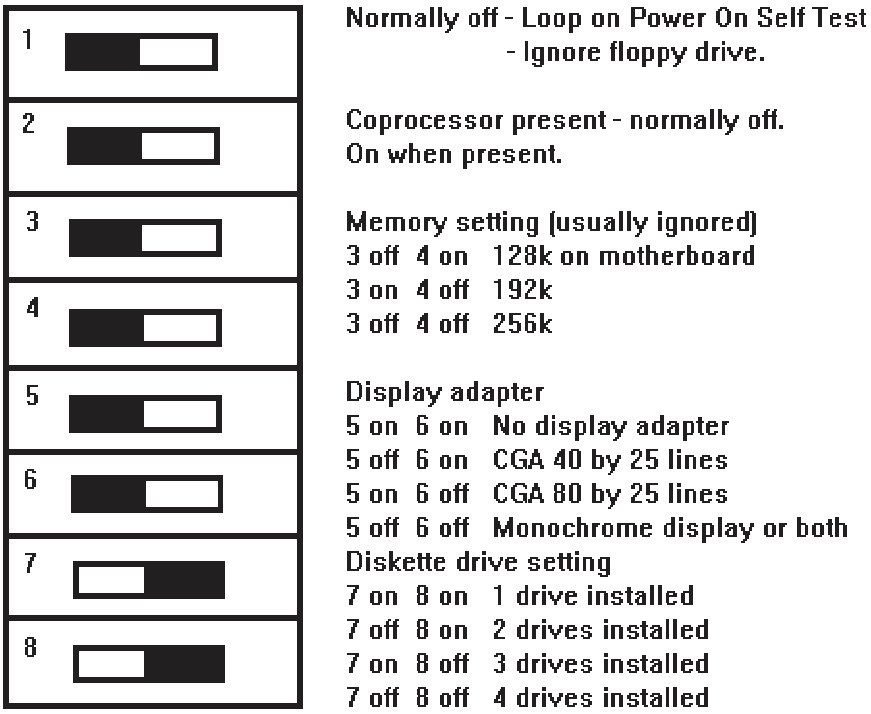
Software interrupt vectors & memory structure

Address (Hex)	Interrupt Number	Name	BIOS Entry
0	0	Divide by Zero	
4-7	1	Single Step	
8-0B	2	Nonmaskable	NMI INT
0C-0F	3	Breakpoint	
10-13	4	Overflow	
14-17	5	Print Screen	
18-1B	6	Reserved	
1D-1F	7	Reserved	
20-23	8	Time of Day	TIMER INT
24-27	9	Keyboard	KB INT
28-2B	A	Reserved	
2C-2F	B	Communications	
30-33	C	Communications	
34-37	D	Disk	
38-33	E	Diskette	DISK INT
3C-3F	F	Printer	
40-43	10	Video	VIDEO I/O
44-47	11	Equipment Check	
48-4B	12	Memory	MEMORY SIZE-DETERMINE
4C-4F	13	Diskette/Disk I/O	
50-53	14	Communications	RS232 IO
54-57	15	Cassette I/O	
58-5B	16	Keyboard	KEYBOARD IO
5C-5F	17	Printer	PRINTER IO
60-63	18	Resident	BASIC F600:0000
64-67	19	Bootstrap	BOOT STRAP
68-6B	1A	Time of Day	TIME OF DAY
6C-6F	1B	Keyboard Break	DUMMY RETURN
70-73	1C	Timer Tick	DUMMY RETURN
74-77	1D	Video Initialisation	
78-7B	1E	Diskette Parameters	
7C-7F	1F	Video Graphics Characters	
20-3F		Reserved for DOS	
41		First Fixed Drive table	
46		Second Fixed Drive, AT PC and up	
60-67		Reserved for User Software Interrupts	
80-85		Reserved by BASIC	
86-F0		Used by Basic Interpreter	

System Configuration

The XT dip switch

To "tell" the 8088 microprocessor what equipment the PC or XT had installed, a small configuration dip switch was installed on the motherboard. This has settings for the number of diskette drives, the type of display adapter, the amount of memory on the motherboard and whether the coprocessor is installed.



Normally off - Loop on Power On Self Test
- Ignore floppy drive.

Coprocessor present - normally off.
On when present.

Memory setting (usually ignored)
3 off 4 on 128k on motherboard
3 on 4 off 192k
3 off 4 off 256k

Display adapter
5 on 6 on No display adapter
5 off 6 on CGA 40 by 25 lines
5 on 6 off CGA 80 by 25 lines
5 off 6 off Monochrome display or both
Diskette drive setting
7 on 8 on 1 drive installed
7 off 8 on 2 drives installed
7 on 8 off 3 drives installed
7 off 8 off 4 drives installed

NOTE: Having more than two diskette drives in an XT can confuse the machine if it has a hard disk installed as drive C:.

Please note that the above is only here for the sake of completeness as XT’s have been obsolete for some time.

The equipment settings were later to be saved in battery backed up memory in the design of the AT machine.

The Equipment Information Settings and Setup

In the AT the system configuration settings are saved in battery backed up CMOS ram contained as a series of registers inside a real-time clock chip called the MC146818. This chip was originally developed by Motorola and has been integrated into to some VLSI AT designs. With newer ISA motherboard designs an enhanced chip having 128 registers has been used, allowing for more settings to be stored.

Registers		Description	Function
00	00	Real-time clock Information	Seconds
01	01		Alarm Seconds
02	02		Minutes
03	03		Alarm Minutes
04	04		Hours
05	05		Alarm Hours
06	06		Day of week
07	07		Day of month
08	08		Month
09	09		Year
10	0A	Status register	A
11	0B	Status register	B
12	0C	Status register	C
13	0D	Status register	D
14	0E	Diagnostic status byte	
15	0F	Shutdown status byte	
16	10	Diskette drive types A: and B:	stored in 4 bits
17	11	Reserved	
18	12	Hard disk drive type 1 to 14	
19	13	Reserved	
20	14	Equipment byte	
21	15	Base memory low byte	
22	16	Base memory high byte	
23	17	Extended memory byte (Low)	
24	18	Extended memory byte (High)	
25	19	Hard disk (C:) type 16 to 49	
26	1A	Hard disk (D:) type 16 to 49	
27 to 45	1B to 2D	Reserved	
46	2E	Cmos checksum	
47	2F	Cmos checksum	
48	30	Expansion memory byte (Low)	
49	31	Expansion memory byte (High)	
50	32	Century	
51	33	Information flags	
52 to 63	34 to 63	Reserved	

The Equipment Information Settings and Setup (contd.)

register 10h Floppy disk drive types
0 in 4 bits means no disk drive
1 means a 360k disk drive
2 means a 1.2 MB disk drive
3 means a 720k 3.5" disk drive
4 means a 1.44 MB 3.5" disk drive
6 means a 2.88 MB 3.5” disk drive.

For a very complete discussion of the CMOS, see the September 13th 1994 issue of PC Magazine on page 339.

The hard disk type numbers that were stored in the CMOS referred to a set of tables, hard coded in the BIOS. These largely have been superseded by ‘user defined’ hard disk entries. Recently the disk tables have disappeared from the BIOS as the user defined cmos memory stores the parameters.

Hard Disk table entries

To use as much of the capacity of a drive as possible, the controller must “know” about the characteristics of that hard drive. When booting up, the BIOS in ROM on the motherboard looks at the expansion bus for additional ROMs to those on the motherboard. One of these ROMs may be on a hard disk controller. If IBM compatibility is adhered to, it has a hex address of C8000 in the XT but can differ in AT machines. This also applies to SCSI and ESDI controller cards with a BIOS attachment generally at segment C800h. The ROM on the hard disk controller in the XT contains the hard disk table, however, in the AT it is in the motherboard ROM as a series of tables. The tables can usually be found in the BIOS at **F000:E401**. Interrupt vectors 41h and 46h are used to direct disk routines to the tables for drives C: and D:.

The hard disk table is a listing of a number of different drive types according to characteristics i.e. how many heads, cylinders, etc.(table 1), as well as a type number for each drive type listed. This type number will differ from machine - if an ST225 is a type 2 under one brand it may be a type 7 on another brand of machine.

When an MFM type hard disk is purchased subsequent to buying a PC it usually needed a low level, or hard, format prior to being formatted with DOS and/or partitioned under the operating system. Occasionally such a format was also useful if severe fragmentation of disk space keeps occurring, or if a different operating system is to be used. Note not different versions or levels but totally different systems such as UNIX or PICK.

Low Level Formatting of Hard Disk

All information stored on a hard disk is erased during a low level format, so this procedure should not be undertaken lightly. The XT PC has the formatting procedure installed as part of the bios resident on the controller card. The AT PC's usually don't have such support for formatting and require the use of a low level formatting program. The program can be supplied by the hard disk manufacturer or it can be part of the machines diagnostic software. Disk Manager by Ontrack has been supplied by SEAGATE with most of its MFM drives to allow low level formatting. The IBM AT diagnostics has as part of its program a low level formatting program.

NOTE: More modern AT BUS or IDE disks do not require low level formatting and can be damaged as a result of such a format.

MFM (Modified Frequency Modulation) drives are tested in manufacture for unreliable areas on the platters and a printed table is generally provided with the drive. XT PC's usually store the disk parameters on the disk, so that entering the number of heads, cylinders etc in the low level procedure stores the values.

With AT PC's this information is stored in the ROM as a fixed table corresponding to the type number of the drive. The correct setting of the type number is part of the setup procedure and must be done prior to the low level format. See the section on the CMOS setup to determine the drive type number.

Some of the valuable information given on the printout sheet with the drive includes:

Number of Heads.
Number of Cylinders.
Whether Write Precompensation is Used.
The Starting Write Precompensation Cylinder.
The Cylinder Number of the Park Cylinder/Landing Zone.
A Bad Track Listing. (only with MFM drives)

The documentation that comes with the drive controller will usually detail the procedure for doing a low level format on a suitable drive in an XT. The instructions normally require the use of debug but are documented step-by-step so no previous debug experience is needed. (Debug normally comes with DOS). When formatting make sure that there is a DOS path to this file. In the AT (80286) the low level format is done from within the Setup Options or using a "Disk Manager" utility.

NOTE: Interrupt vector 41 points to the hard disk table for the first hard drive, and interrupt 46 points to that of the second hard drive.

The BIOS holds tables for hard disk types 1 to 14, 16 through 47 possibly up to 49. The type number is stored in the CMOS and refers to a table in the BIOS rom. These tables are stored in the BIOS at segment F000 and offset E401 hexadecimal. Using Debug we can see the tables :

```
C:\DOS>debug
-df000:e401
F000:E400 32 01 04 00 00 80 00-00 00 00 00 00 31 01 11 2...........1..
F000:E410 00 67 02 04 00 00 2C 01-00 00 00 00 00 67 02 11 .g....,......g..
F000:E420 00 67 02 06 00 00 2C 01-00 00 00 00 00 67 02 11 .g....,......g..
F000:E430 00 AC 03 08 00 00 00 02-00 00 00 00 00 AC 03 11 ................
F000:E440 00 AC 03 06 00 00 00 02-00 00 00 00 00 AC 03 11 ................
F000:E450 00 67 02 04 00 00 FF FF-00 00 00 00 00 67 02 11 .g...........g..
F000:E460 00 CE 01 08 00 00 00 01-00 00 00 00 00 FF 01 11 ................
F000:E470 00 DD 02 05 00 00 FF FF-00 00 00 00 00 DD 02 11 ................
F000:E480 00 84 03 0F 00 00 FF FF-00 08 00 00 00 85 03 11 ................
F000:E490 00 34 03 03 00 00 FF FF-00 00 00 00 00 34 03 11 .4...........4..
F000:E4A0 00 57 03 05 00 00 FF FF-00 00 00 00 00 57 03 11 .W...........W..
F000:E4B0 00 57 03 07 00 00 FF FF-00 00 00 00 00 57 03 11 .W...........W..
F000:E4C0 00 32 01 08 00 00 80 00-00 00 00 00 00 3F 01 11 .2...........?..
F000:E4D0 00 DD 02 07 00 00 FF FF-00 00 00 00 00 DD 02 11 ................
F000:E4E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
F000:E4F0 00 64 02 04 00 00 00 00-00 00 00 00 00 97 02 11 .d..............
```

The first entry is a 16 bit word containing the number of cylinders on the drive. The second entry is a byte containing the number of heads. The last entry is a byte telling the system the number of sectors in a track. In this case, 17 sectors per track. Most MFM style drives only supported 17 or 26 sectors per track. IDE drives can have up to 63 sectors per track.

The 528MB Barrier

Once again the estimations of the designers were out of line. The limitation is a combination of both the DOS limit of 1024 cylinders and the BIOS limit of 63 sectors and 16 heads. Thankfully most new BIOS designs have a way round this, namely the LBA mode. LBA is either Logical Block Addressing or Linear Block Addressing. Either acronym, the BIOS translates the number of actual cylinders to proportionately less to let DOS use the full capacity of the disk. For those who don’t have this in their BIOS a DOS device driver can be loaded to translate the cylinders. This is not really desirable as it will slow down the access and take up memory.

See the later section on the file allocation table for reasons why you don’t want your 2GB drive as one drive.

CPU Speed Enhancement

The PC has been on steroids for some time now, using all the tricks it could to boost performance. If some of these had been around for the 386SX, we might have stayed longer with the 386 as the processor of choice.

Interleaved memory - reduces wait states by dividing the total memory into banks (2 or 4) that are accessed alternately. While one is being refreshed, the other can be accessed with no wait states. This can cut wait states by 50-75%.

Paged memory - special chips designed to allow back-to-back accesses without wait states within blocks of memory called pages. This can cut wait states by 80%.

Processor ram caching - 32k to 256k of fast static ram is attached to the processor by using a cache controller. The i82385 cache controller transfers data from slow main memory to this cache. The controller is programmed to 'spoonfeed' the cpu. It can be correct sometimes, sometimes not adding further wait states.

NOTE: The amount of Static Ram needed for performance improvement is dependent upon the application software and need not be more than 128k. Adding more will be like adding a “go fast stripe” - nice to look at, nothing more.

With the Pentium the situation is different as the Pentium can make use of a larger cache. This should be proportional to the amount of main memory. Quite a few Pentium machines recently have memory of 64MB to 128MB and should have 512k to 1MB of cache.

Shadow ram - the process of copying the BIOS ROM into main memory on start up. This allows the cpu to read the data and instructions faster. 10 to 15% improvement. Unfortunately not usable with the newer operating systems as the BIOS plays no part in the running of the system.

Bus Speed - The 8MHz ISA bus was brought about by peripheral boards not being reliable at higher speeds. Windows 3.x brought about a need for higher video updating speeds, thus the VESA local bus was born. This runs the peripheral cards, typically the VGA card at the same speed as the cpu. This removes the ISA bottleneck and improves performance by 4 to 5 times. PCI was designed as an alternative to local bus by Intel for the Pentium. It has been back installed for 486 CPUs on most modern motherboards and is the current industry standard.

Communicating with the outside world

Terminal emulators

The PC community would have you believe that mainframes and mini-computers no longer exist. It just isn't so! There are vast amounts of data stored in mainframe-based databases, which is available for terminal based users to access. To get your PC to use and access that data you require a terminal emulator program. Most terminal emulator programs simulate the operation of a standard terminal type such as VT100 a DEC terminal or 3270 an IBM terminal.

File transfer is sometimes easy to and from the "HOST" computer and sometimes non-existent. Beltel is a terminal based service operated by the South African Post Office using mainframe computers in the the main cities to hold very large databases. File transfer is possible with Beltel and the Post Offices' software called PcBel.

Beltel comes in two flavours, an ASCII terminal based service using VT100 based emulation and a PRESTEL based service. The former has not been too successful, but carries on operating with some information providers. The PRESTEL emulation is a block graphic orientated display giving colour and chunky graphic pictures.

Perhaps the one single outstanding service feature added to Beltel has been the phone directory. This service is far quicker to use and is more up to date than the phone book. The French post office literally gave away Prestel terminals to telephone subscribers instead of printing phone directories, giving rise to a large user base for their service.

Other services do exist, such as "Bulletin Boards" which though proscribed recently from storing mail for other users, allow upload and download of Public Domain and Shareware programs. These services allow the use of various terminal emulators and of course file transfer.

In similar vein the large databases for specialist use have grown over the past few years. These are the private or subscriber databases such as the Pharmacists database or the tender bulletin database ERNII.

NOTE: A simple terminal emulator program (TERMINAL.bas) is provided.

Terminals and Terminal emulators respond to “Escape sequences” that allow remote control of terminals. An example of this is the ANSI escape sequence to clear the screen.

Esc [2 J this will clear the screen on the terminal or the pc if the terminal software emulates the ANSI terminal. ANSI stands for American National Standards Institute. You can try this by loading Ansi.sys in the Config.sys file and placing the characters in an Echo statement in a batch file.

Modems

Possibly the most common type of modem available today is the "Hayes" compatible modem. This comes from an "intelligent" modem designed by Hayes of the USA called the Hayes Smartmodem. This modem made use of a microprocessor to provide a command orientated operation from the host machine. It can dial a number, answer a call, negotiate the speed of the link and go online or offline. Its operational commands are preceded by the letters "AT" for **AT**tention, and terminated by a carriage return.

AT	OK	check modem in command mode.
ATS0=1	OK	Puts modem into auto-answer mode.
AT&C1	OK	Makes DCD/RLSD follow carrier detect.
ATDP7644681	OK	Dials 7644681 in pulse mode.
ATDT7644681	OK	Dials 7644681 in tone mode.
ATI0	(varies)	Identifies the modem software.

Later models incorporated error correction into the design (Microcom Network Protocol MNP) and on the fly data compression to further increase throughput of data. Data rates on normal telephone lines can now achieve in excess of 33.6kbits per Second. Since most data flow is in a single direction, the modems simulate full duplex operation, turning the line around when necessary.

Modem Standards

Modem standards are the definitions of electrical and telecommunications characteristics which enable modems from dissimilar manufacturers to speak to each other.

Some common standards:

Bell 103	US standard for 300 bps
CCITT V.21	International standard for 300 bps
Bell 212A	US standard for 1200 bps
CCITT V.22	International standard for 1200 bps
CCITT V.22 bis	US and international standard for 2400 bps
CCITT V.23	International videotex standard (1200/75 bps or 75/1200 bps).

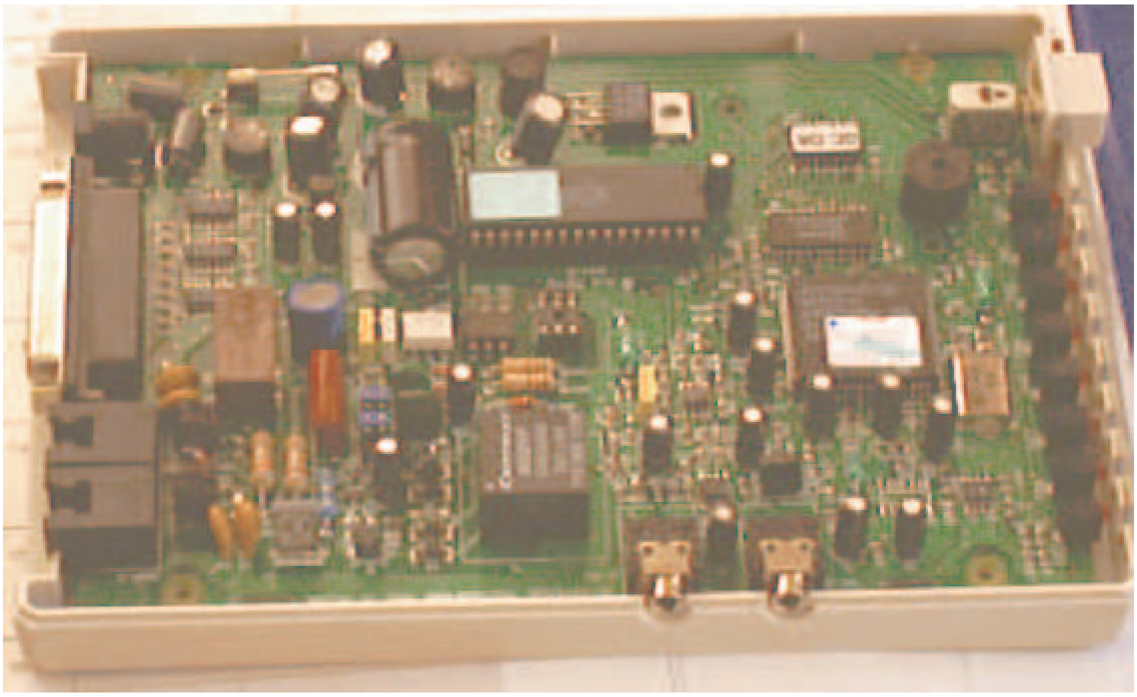
Quite a few suppliers recently have taken to importing foreign modems, designed for US operation. These modems have several drawbacks:

They use BELL tones by default for communication. This can interfere with Telkom's call metering, making your phone bill go negative! If this happens and they detect it, it could mean equipment confiscation and a severe fine. Check on this and use ATB0 or 1 to set it correctly.

The lightning protection is only good for puny overseas lightning. A good Highveldt storm and you can wave it goodbye.

It may not connect reliably with a local modem for some time and this could have your software disconnect.

Talk to your Modem



A typical modem with top removed

How to talk to a modem [politely]

[Refer to the indicator lights. CD, RD, SD, TR, OH, MR, AA on the front panel]

CD - carrier detect. After the modem negotiates a connection with another modem, this light will light, indicating that a carrier is being received from the other modem. Any blinking of this light usually indicates a fault condition.

RD - Receive data. On receiving any data this light will flash. A permanently off light will mean that no data is being received. An on light means a fault condition at the other modem on the RS232 side.

SD - Send data. On sending any data this light will flash. A permanently off light will mean that no data is being sent. An on light means a fault condition at the local modem on the RS232 side.

TR - really DTR or Terminal Ready. This indicates the state of the DTR line from the terminal or pc. If it is off the terminal is “not ready” and either software must be loaded or the terminal made ready.

OH - Off Hook. This light goes on to show that the telephone line is “off hook” as if you had lifted the handset.

MR - Modem Ready or Data Set Ready. This indicates that the modem is ready to receive commands or data.

AA - Auto Answer. This shows that the modem is “auto answer” enabled and will answer the phone if it rings. This can be a problem if it is connected to your normal phone line and not to a dedicated line.

HS - High speed. This is almost obsolete and used to show the user the modem was communicating at 1200 baud, when the modem could also talk at 300 baud. It is generally an indication of the highest speed of the modem being used.

HAYES AT commands

The simple Hayes Smartmodem used commands prefixed with AT and terminated by a carriage return. So most commands are AT[something] <carriage return>. The exception to this rule is ATA, which needs no CR and immediately answers the call.

Command string	Function	Explanation
ATB0	Set CCITT tones	In SA these are the correct tones to use, as we are “technically” European.
ATB1	Set BELL tones	In the USA these are the tones used. So if you are connecting with a USA modem, use ATB1 to send and receive the correct tones.
ATD[nnnnnn]	Dials the number nnnnnn	If no Tone or Pulse specified, it will dial using what was last used.
ATDP[nnnnnn]	Dials the number using pulse mode	
ATDT[nnnnnn]	Dials using tone dial	DTMF is much quicker than pulse and has become quite common in SA.
ATDR	Originates the call in answer mode	Not normally used.
ATD,	The comma makes the modem wait 2 seconds before continuing to dial.	Useful for dialling out through a switchboard. eg ATD0,123456 will use 0 to dial out, wait 2 seconds and then dial 123456.
ATDW	Waits for second dial tone before dialling.	An alternative to using the comma.

Example command strings

To set the modem into auto-reliable mode with verbose messages, bidirectional hardware flow control, maximum size of 256, BPS rate adjust off and MNP5 enabled :-

```
AT X4 \V1 \N3 \Q3 \A3 \J0 %C1
```

```
AT X4 V1 E1 Q0 S0=0
```

This string will set extended codes, verbose, echo on, quiet is off and will not auto-answer.

```
AT S0=1
```

This string sets the modem into auto-answer.

Extended HAYES AT commands

Nineteen commands have been added to the standard HAYES AT command set to control the MNP error corrector and compressor. Here is a summary of those commands :-

AT\An	Set the maximum MNP block size
AT%An	Set the auto-reliable fallback character
AT\Bn	Send break signal
AT\Cn	Set the type of MNP auto-reliable buffering
AT%Cn	Set to class 5
AT\En	Set data echo
AT\Gn	Set modem (DCE) flow control
AT\Jn	Set bits per second adjust from the terminal
AT\Kn	Set break type
AT\Ln	Set MNP service type (Block or Stream)
AT\Nn	Set operating mode
AT\O	Initiate an MNP link during a normal link
AT\Qn	Set Terminal (DTE) flow control
AT\U	Accept an MNP link during a normal link
AT\Vn	Set extended result codes
AT\Xn	Set XON/XOFF pass through
AT\Y	Initiate or accept an MNP link during a link
AT\Z	Terminate an MNP link, return to normal link

Command descriptions

AT\An The maximum transmitter MNP block size

Use this command to set the maximum block size for MNP stream link connections.

n=0 Sets the maximum block size to 64 characters
n=1 Sets the maximum block size to 128 characters
n=2 Sets the maximum block size to 192 characters
n=3 Sets the maximum block size to 256 characters

AT\Bn Send break signal

This will send a break signal to the remote system from the command state of length n * 100 milliseconds (n = 1 to 9)

n = 0 gives a default of 300 ms

Break length is always 300ms in Reliable mode.

To send a break and re-enter the connect state, at the OK prompt :-
type AT\B4O <carriage return>

AT\Cn The type of MNP auto-reliable buffering

n=0 Does not buffer data

n=1 Buffers data for 4 seconds, until 200 characters have been buffered, or the modem detects a packet by the SYN character and switches to reliable mode. If the buffer fills, the modem switches to normal mode and passes the data through to the serial port.

n=2 Does not buffer data. Switches to normal mode upon receipt of a fallback character defined by the AT%A command and passes that character to the serial port.

AT%Cn Set class 5

n=0 Disable class 5 operation
n=1 Enable class 5 operation

AT\En Echo control

n=0 Echo turned off while on line.
n=1 Echoes data sent by the local terminal while on line.

Echo is disabled in the reliable link mode.

AT\Gn Modem flow control

n=0 Disables modem port flow control.
n=1 Sets modem port flow control to XON/XOFF.

AT\In Show software ID

n=0 Displays 248
n=1 Displays the checksum of the error corrector firmware.
n=3 Displays the revision of the SC11018 firmware code.
n=4 Displays the modem type and serial or parallel mode & hybrid status.
n=5 Displays the error corrector revision.

AT\Jn Bits Per Second adjust

n=0 BPS rate adjust off
n=1 BPS rate adjust on

When a connection is established the modem automatically adjusts the DTE port speed to match the connection rate. If the modem connects at a lower rate than requested, and the DTE remains at the original speed, it must be manually set to the new connection speed. When n=0 the DTE port rate is independent of the connection rate. The data interface to the DTE is buffered.

AT\Kn Control of break signal type

This command sets the type of break the modem sends when the DTE sends a break to the modem. The modem can buffer up to 4 breaks on both the modem port and the DTE port.

n=0,2,4 Break signal from DTE is ignored. Enter command state. To send break, issue the AT\B command. This sends a break of the appropriate type as outlined below.

n=1 Expedite destructive. Buffer is emptied and break signal is immediately sent to the remote system.
n=2 Expedited. Break signal from DTE is ignored. Enter command state.
n=3 Immediately send break signal to remote modem.
n=4 Break signal from DTE is ignored. Enter the command state.
n=5 Queued. Send a break signal to the remote modem after transmitting any buffered data.

Break sent to the modem while in command state during a reliable or normal connection.

n=0,1 Expedite destructive. Empty data buffers and immediately send a break to the remote system.
n=2,3 Expedite. Send break immediately.
n=4,5 Queued. Send a break signal to the remote modem after transmitting any buffered data.

AT\Ln MNP service type : Block or Stream

n=0 Initiates stream link
n=1 Initiates block link

Note: This should only be used with programs that use the Microcom Interface Protocol.

AT\Nn Request link type

n=0 Sets the error corrector to Normal mode
n=1 Sets the error corrector to Direct mode
n=2 Sets the error corrector to Reliable mode
n=3 Sets the error corrector to Auto-reliable mode

AT\O Initiate an MNP link during a normal link

If the current link is a normal link, this command will originate an MNP link. This command should be used in conjunction with the AT\U command.

AT\Qn Set Terminal (DTE) flow control

n=0 Disables flow control
n=1 Sets flow control to XON/XOFF
n=2 Enables CTS flow control
n=3 Enables CTS/RTS flow control

AT\S Show type of link

Displays one of the following messages:-
Direct link
MNP link
Normal (buffered) Link

AT\U Accept an MNP link during a normal link

If the current link is a normal link, this command will accept an MNP link. This command should be be used in conjunction with the AT\O command, thus :- AT\UO

AT\Vn Extended result codes

n=0 Disables extended result codes
n=1 Enables extended result codes

Long form codes Short form codes

OK	0
CONNECT	1
RING	2
NO CARRIER	3
ERROR	4
CONNECT 1200	5
NO DIALTONE	6
BUSY	7
NO ANSWER	8
CONNECT 2400	10
CONNECT 0300/REL	20
CONNECT 1200/REL	21
CONNECT 2400/REL	23

AT\Xn XON/XOFF pass through

n=0	Process XON/XOFF characters and don't pass them to the DTE.
n=1	Process XON/XOFF characters and do pass them to the DTE.

AT\Y Make an MNP link during a normal link

If the current link is a normal link, this command will accept or originate an MNP link depending on whether the modem originated or answered the call.

AT\Z Terminate an MNP link

If the current link is an MNP link, this command will cause the link to be terminated and the error corrector will switch to normal mode.

AT&Z Store phone number

BIOS - The Basic Input / Output System

Its a Windows World

Over the last few years, Windows 3.x and Windows 95/98 have taken over the vast majority of personal computers. Windows makes very little use of the BIOS in versions 3.0 to 3.11 and none in Windows 95/98 or Windows NT. The only purpose of the BIOS now is to boot the operating system from floppy or more usually the hard disk. The original BIOS was designed to provide a consistent interface between programs or the operating system and the hardware. MS-DOS used the BIOS for both keyboard functions and the display fiunctions as well as the disk functions.

‘MS-DOS Compatible’ and ‘ROM BIOS Compatible’

MS-DOS compatible applications used only the documented MS-DOS function calls and do not call the ROM BIOS or access the hardware directly. They use ANSI escape sequences for screen control, and their input and output is redirectable. An MS-DOS compatible application will run on any machine that supports MS-DOS, regardless of the machine configuration. Because of the relatively poor performance of MS-DOS's built-in display and serial port drivers, few popular programs other than compilers, assemblers, and linkers fall into this category.

ROM BIOS compatible applications use the documented MS-DOS and ROM BIOS function calls but do not access the hardware directly. As recently as five years ago, this strategy might have significantly limited a program's potential market. Later on, the availability of high-quality IBM compatible ROM BIOS’ from companies such as Phoenix ensured the dominance of the IBM ROM BIOS standard; virtually no machines are being sold in which a program cannot rely as much on the ROM BIOS interface as it might on the MS-DOS interface. However, the ROM BIOS display and serial drivers are still not adequate to the needs of high-performance interactive applications, so the popular programs that fall into this category are few.

Hardware-compatible applications generally use MS-DOS functions for mass storage, memory management, and the like, and use a mix of MS-DOS and ROM BIOS function calls and direct hardware access for their user interfaces. The amount of hardware dependence in such programs varies widely. For example, some programs only write characters and attributes into the video controller's regen buffer and use the ROM BIOS to switch modes and position the cursor; others bypass the ROM BIOS video driver altogether and take complete control of the video adapter.

A Bios function example.

The Basic Input / Output System or BIOS is the program that interfaces the disk based programs to the hardware installed in the machine. It provides this interface as a series of INT calls with function codes usually passed in the AX register. Pointers and other data are usually placed into the other registers prior to the call.

eg:

```
        Mov   Ah,0E         ; function to print character at current cursor location
        Mov   Al,41         ; the character "A" in ASCII
        Int   10            ; the video BIOS interface for the display adapter.
```

This will print a character "A" at the next character position and move the cursor position to the next character position.

The BIOS interfaces most of the standard hardware - display, printers, keyboard, disks, memory, asynchronous communications adapters. It mainly uses the interrupts vectors between 0 and and 1Fh.

BIOS Shadowing

Some machines provide the ROM BIOS shadowing option. While this feature provides an advantage with Microsoft MS-DOS, it is not an advantage with Microsoft Windows NT or Windows 95/98.

ROM BIOS shadowing is the process of copying the BIOS from ROM into RAM and using either hardware or 386 enhanced mode to remap the RAM into the normal address space of the BIOS. Because reading RAM is much faster than reading ROM, BIOS-intensive operations are substantially faster. For example, MS-DOS uses the BIOS to write to the screen; therefore, with ROM BIOS shadowing, directory listings run more quickly.

NOTE: Windows NT does not use the BIOS (except during startup); therefore, no performance is gained by shadowing. If ROM BIOS shadowing is not used, more RAM is available. With Windows NT, there is an advantage to disabling the ROM BIOS shadowing option.

The ROM BIOS Date

Getting the ROM-BIOS date for a computer can be an important clue in diagnosing a variety of potential hardware and software compatibility problems. QBASIC program to display the ROM-BIOS date in month-day-year format for an IBM PC:

```
DEF SEG = &HF000
FOR A = 0 TO 7
        PRINT CHR$(PEEK( &HFFF5 + A ));
NEXT A
```

Programs that change the port addresses in the ROM BIOS Data area

Port entries in the BIOS data segment can be changed by network redirector software or by the MS-DOS mode command. The mode command does not change any existing nonzero entries. It places a 1 in port entries that didn't previously exist but that have been created by mode redirection. For example, if you have a physical LPT1, but not an LPT3, and you execute the mode LPT3=LPT1 command, then output sent to LPT3 will be redirected to LPT1. Because LPT3 does not exist in hardware, the BIOS data segment area contains zeroes in the LPT3 section. When mode creates a virtual LPT3 through its redirection, it places a 1 in the LPT3 section of the data segment.

MS-Net and LAN Manager redirector-based networks operate in a similar manner. The redirector does not modify any nonzero entries in the data segment when it performs redirection. When you redirect an existing port, the port entry is not changed. This is done so that when the redirection is removed, you can again print to the physical port. However, when you redirect a nonexistent port, the zero entry in its section is changed to a 2. Many applications check the BIOS data segment to determine whether a port exists. Both mode and the MS-Net redirector place nonzero entries in this field as necessary, so the application will function correctly.

Novell networks place a 3BCh entry in the data segment locations for LPT2 and LPT3 when the NetWare shell is loaded. If you do not have a physical LPT1 in the machine, you will have a zero in the port entry for LPT1.

DOS Startup

What happens after you switch-on?

1 Power supply (switched mode) switch on, dangerous high voltages etc. Usual failure occurs at this time.

2 Power Good signal coupled to the Reset circuit releases the cpu which ‘jumps’ to F000:FFF0

3 The reset vector is in the motherboard BIOS rom and instructs the cpu to jump to the Power On Self Test section. The motherboard components are tested including the cpu and the memory for basic read/write operation. It will check for printer port hardware and serial ports as well as initialising the floppy and hard disk controllers.

4 The cpu will then ‘look’ for a bios attachment flag of 55,AA hexeadecimal on a page (256 byte) boundary. If found the cpu will jump to the initialisation code that is in that attachment rom. For example the VGA bios will then initialise the VGA adapter and may print a copyright message on the screen. Other BIOS attachments can be the SCSI bios to initialise and attache the SCSI hard disk into the system or a network boot prom on a network adapter that boots the pc from a file server.

5 The cpu will attempt to ‘boot’ from the first device usually the A: drive though these days this may be the hard disk C: set by the bios setup. If the floppy drive times out the cpu will be instructed to try the hard disk. This usually works as the disk drive has been initialised with a partition table and a master boot record (sector). If it does not work for some reason the cpu will be instructed to try to boot the old standby, ROM Basic. This will fail as no pc’s have been supplied with ROM Basic for some years. Some bios’ have been rewritten to allow for this and will produce a message asking for a system (bootable) disk and for the user to insert it somewhere appropriate.

6 The bios will attempt to load and execute the boot sector from the floppy disk. If it is successful the boot sector program will load the two hidden and system files IO.SYS and MSDOS.SYS. If they are not there the boot program will issue the error message “Non system disk or disk error” - “Replace and press a key when ready”.

7 Alternatively if the pc boots from the hard disk the partition table is loaded and executed to get the active boot sector on the disk. This method allows for four possible operating systems to be installed on the same disk drive. Usually only one is installed, typically DOS, Windows 95 or Windows NT. At this point various error messages can appear. “Invalid partition table” - indicating damage to the disk. “Missing operating system” - indicating faulty or damaged boot sector. The latter usually appears when the BIOS setup has been lost through the CMOS battery going flat.

8 The system files are loaded into memory typically at around 30000 bytes and the cpu is jumped to the start address of IO.sys. This is when the message “Starting MS-DOS ....” is printed on the screen.

The "BOOT" Sector

The PC is instructed by its BIOS and startup program to try the floppy disk drive first and to load the boot sector into memory. This sector is present on all DOS structured disks both floppies and hard disks. If the program fails to read a floppy at this point, it will try the hard disk if it is present. The PC will first load the partition table from the hard disk into memory. It will then execute the code and find which of the four tables is ‘active’. The PC will attempt to load the ‘active’ boot sector from the hard disk. Should it fail again the PC may enter ROM BASIC or emit an error message asking for a disk to be inserted into drive A:.

The boot sector contains a small "bootstrap" program and a disk structure table. The table is there to tell DOS how the disk is structured. The program looks for IO.SYS and MSDOS.SYS and loads them into low memory. If these two files are not found, the bootstrap loader program will declare the disk a non-system disk.

The Boot sector structure

```
bsJump            db    3 dup (?)          ; E9 XX XX or EB XX 90
bsOemName         db    '????????'         ; OEM name and version
                                           ; start of bios parameter block
bsBytesPerSec     dw    ?                  ; bytes per sector
bsSecPerCluster   db    ?                  ; sectors per cluster
bsResSectors      dw    ?                  ; number of reserved sectors.
bsFATs            db    ?                  ; number of file allocation tables.
bsRootDirEntries  dw    ?                  ; number of root directory entries.
bsSectors         dw    ?                  ; total number of sectors.
bsMedia           db    ?                  ; media descriptor byte.
bsFATsecs         dw    ?                  ; number of sectors per FAT
bsSecPerTrack     dw    ?                  ; sectors per track
bsHeads           dw    ?                  ; number of heads.
bsHiddenSecs      dd    ?                  ; number of hidden sectors.
bsHugeSectors     dd    ?                  ; number of sectors if bsSectors = 0
                                           ; end of BIOS parameter block
bsDriveNumber     db    ?                  ; drive number (80h)
bsReserved1       db    ?                  ; reserved
bsBootSignature   db    ?                  ; extended boot signature (29h)
bsVolumeID        dd    ?                  ; Volume serial number
bsVolumeLabel     db    11 dup (?)         ; volume label
bsFileSysType     db    8 dup (?)          ; file system type. "FAT12" or "FAT16"
```

NOTE: To restore a garbled or Stoned boot sector make use of an undocumented parameter of FDISK, the /MBR parameter which creates a brand new boot sector. This overwrites any infected or garbled boot sector.

The 32 Megabyte barrier

The 32MB barrier came about from the storing of the number of controllable sectors in a 16 bit word. A sector is 512 bytes. If you multiply 65535 by 512 you get 33,553,920 bytes or 32MB.

The Partition table

The partition table is a structure that specifies the size, starting and ending sectors of a partition on a hard disk. It is the first sector read by the hard disk boot routine. The pc can then determine which of the four partitions is active and should be ‘booted’.

Partition structure

```
peBootable          db    ?    ; 80h if bootable, 00 if not bootable
peBeginHead         db    ?    ; beginning head
peBeginSector       db    ?    ; beginning sector
peBeginCylinder     db    ?    ; beginning cylinder
peFileSystem        db    ?    ; name of file system
peEndHead           db    ?    ; ending head
peEndSector         db    ?    ; ending sector
peEndCylinder       db    ?    ; ending cylinder
peStartSector       dd    ?    ; starting sector relative to beginning of disk.
peSectors           dd    ?    ; number of sectors in partition
```

peFileSystem byte can be :-

00	An unused partition or unknown type
01	DOS with a 12 bit FAT; partition smaller than 10MB.
02	Xenix
03	Xenix
04	DOS with a 16 bit FAT; partition smaller than 32MB.
05	Extended DOS partition
06	Compaq DOS 3.31, 4.0, 4.1 and 5.0 > 32MB partition
07	OS/2 HPFS partition
75	PC/IX
DB	CP/M
FF	Xenix bad block table

NOTE: There is only room for four 16 byte partition records in the table. Thus only four different types of operating system can be used/booted this way.

The DOS System files

IO.SYS & MSDOS.SYS (IBMBIO.COM IBMDOS.COM)

Continuing the boot-up sequence of events. The initialisation module of IO.SYS is invoked and then control is passed to MSDOS.SYS which initialises a disk buffer and file control block. It also determines the hardware status and performs any necessary hardware initialisation. Control is then passed back to IO.SYS which looks for and interprets config.sys if present and installs the device drivers found, into memory.

Next IO.SYS calls DOS function 4Bh which invokes the DOS program loader (called EXEC). It loads the shell program into memory and passes control to that program. Usually EXEC loads COMMAND.COM but any command processor can be used. Examples of alternative command processors are Hewlett Packard's "Pamcode" or 4DOS' command.com.

The Command Interpreter

COMMAND.COM comprises three parts: an initialisation portion, a resident portion and a transient portion. The resident portion is loaded by EXEC and is responsible for loading the transient portion. Amongst others the resident portion contains the routines that handle int 22h (terminate address), input and output errors, int 23h (control break), and int 24h (critical error).

The initialisation portion of COMMAND.COM is loaded immediately after the resident portion. This portion of COMMAND.COM also processes autoexec.bat. If autoexec.bat is not found, it will prompt for date and time. The initialisation portion is then discarded.

The transient portion of COMMAND.COM is loaded into high end memory. It displays the system prompt, contains internal system files, and loads external commands and executable files. It may be overwritten during the execution of a program. At the end of the execution of a program, the resident portion of COMMAND.COM will determine if this is so and if necessary reloads the transient portion. The familiar system prompt is now displayed.

```
C:\>
```

If Command.com cannot be found, a message “Bad or missing Command interpreter” will be displayed. This has a variety of causes, usually the user has overwritten command.com with a previous version, by copying a floppy with *.*. Protect against this by using the shell directive in config.sys or by marking command.com as read only by using the attrib command.

If the command interpreter cannot be found, MS-DOS 6.x will display a nice message asking to be told where Command.com can be found.

DOS’ Disk structures

Clusters and allocation units

A cluster or allocation unit is used by DOS for allocating blocks of storage for files. Generally for a floppy disk (360k) this is two sectors or 1024 bytes and for a hard disk this is 8 sectors or 4096 bytes. Clusters for actual storage start at cluster 2 immediately after the directory on the disk. The directory occupies 7 sectors usually starting at sector 6 of the first cylinder or track. The term cylinder refers to two tracks immediately opposite on both surfaces of the disk. DOS uses this as one cylinder to store contiguous data and is more efficient than storing data on adjacent tracks on one side.

The Directory entry

Byte position	Function
00 to 07	Filename (8 characters)
08 to 10	Extension (3 characters)
11	Attribute byte
12 to 21	not used / reserved
22 to 23	Time last modified hours * 2048, Minutes * 32, Seconds / 2
24 to 25	Date last modified (Yr. -1980) * 512, Month * 32, Day
26 to 27	Number of first cluster occupied (pointer to start on disk)
28 to 31	File size in bytes (LONG)

There are a limited number of directory entries as each entry takes up 32 bytes in 7 sectors of 512 bytes. This limits the number of files normally found in the ROOT of a disk.

The attribute byte is used to tell DOS about the file, whether to display it when a DIR is executed, whether it is read only or a system file and whether the file has been backed up. It is also used by Novell to indicate a shareable file.

DOS doesn't just use the directory entry to find and control its files. It uses a File Allocation Table as a kind of map to locate where on disk parts of a file are residing. This "map" is a series of 12 bits stored as a pattern on disk. Later disks and larger disks have to use 16 bits to map out locations on hard disks. DOS now supports up to 2 GigaBytes storage on disk where previously it could only accommodate 32 MB.

The File Allocation Table (FAT)

There happen to be two FAT's on each disk, it is thought that this was to provide DOS with the means of recovering a damaged FAT, but it is only used by third party disk utilities, such as the Norton Utilities.

Some FAT entries

Entry (cluster)	Value	Meaning
0	FFF	Double-sided 8 sectors per track
	FFE	Single-sided 8 sectors per track
	FFD	Double-sided 9 sectors per track
	FFC	Single-sided 9 sectors per track
1	FFF	Filler - does nothing
2	003	This is a pointer to the next cluster ie: 3
3	004	This is a pointer to the next cluster ie: 4
4	005	This is a pointer to the next cluster ie: 5
5	FFF	This is the last cluster in this file
6	000	Unoccupied cluster
7	000	Unoccupied cluster
8	FF7	This cluster has a bad sector in it. Marked unusable.

MS-DOS Features

Programs use MS-DOS system functions to allocate memory, load programs, read from and write to files and devices, connect to a network, and so on.

Programs that use MS-DOS system functions have access to the following features of MS-DOS:

- **File system**

  The MS-DOS file system consists of the files, directories, and supporting data structures on the disks of the computer. Although MS-DOS controls the file system, programs can create, read from, write to, and delete files and directories. The primary supporting data structure for the file system is the file allocation table (FAT). Programs do not access the FAT directly. Instead, MS-DOS manages all the details of the operations on files, including updating the FAT as files are created and modified.

- **Character devices**

  Character devices process data one byte (one character) at a time. Examples of character devices are the computer's keyboard, screen, and serial and parallel ports. Programs can open, read from, and write to character devices by using the same functions as they use for accessing files. Devices have logical names, such as CON and PRN, that programs use to open them. Programs can set operating modes for character devices by using input-and-output-control (IOCTL) functions.

- **Program execution**

  Although MS-DOS is a single-tasking operating system - that is, it runs only one program at a time - programs can load and run other programs. While one program runs, the program that started it is temporarily suspended. MS-DOS ensures that adequate memory and other resources are available to each program.

- **Memory management**

  When it starts a program, MS-DOS allocates memory for program code and data and copies the program file from the storage medium into memory. Programs can free unneeded memory or allocate additional memory while they run. MS-DOS organizes memory in blocks of one or more paragraphs (a paragraph is 16 bytes).

- **Networks**

  A network enables programs running on one computer to use the drives and devices of other computers. Programs can make connections to network drives and devices and then access files and character devices to open, read from, and write to the network drives and devices.

- **National language support**

  National language support permits programs to adapt themselves for operation in a variety of national markets. Programs use country information to prepare the characters and formats for date, time, currency, and other information they display; they use code pages to display and print characters that are language-specific or country-specific.

- **Interrupt handling**

  Programs can install custom interrupt handlers to carry out special processing while they run. For example, a program can install a CTRL+C handler that replaces the default action when the user presses the CTRL+C key combination.

- **Task-switcher notifications**

  Programs can add themselves to the notification chain of the MS-DOS task switcher. Programs that are sensitive to task switches, such as communication programs that must respond immediately to asynchronous input, add themselves to the chain to control when and under what conditions task switching occurs.

DOS Utility Programs

Mirror

Mirror is the same utility that Central Point provided with PC Tools and provides the means of saving partition tables and File allocation tables. Mirror /Partn will save the hard disk's partition table onto a floppy disk for later restoration. Running Mirror in the autoexec.bat file allows for sophisticated file undeletion and unformatting a disk.

NOTE: Mirror has been dropped from MS-DOS 6 along with the Qbasic example programs. Copy them across from your OLD_DOS.1 directory into DOS to carry on using them.

Share

Share is a terminate and stay resident program that provides file locking and protection against overwriting data.

Typically, you use SHARE in a network or multitasking environment in which programs share files. SHARE loads the code that supports file-sharing and locking in these environments. Once you install Share, MS-DOS uses the code loaded by Share to validate all read and write requests from programs. For example, if two people are accessing the same data file, SHARE manages the file so both people do not write to the file at the same time.

When deciding how many bytes to allocate for file sharing, note that each open file requires enough space for the length of the full path and filename. The average length of a filename and its path is 20 bytes.

Undelete - How to protect against accidental/deliberate file deletion.

By using the DOS utility Undelete in sentry mode, you can protect against files being lost forever. Normally running Undelete will attempt to recover files that have been deleted. Undelete has several other options that loads Undelete into memory, so that when files are deleted, they are moved to a sentry directory. In Windows 95 it has become the ‘recycle bin’. We use Undelete /SC to guard files on drive C:. Remember the /? option will provide all the command line options for that program. eg: Undelete /?

DOS Performance enhancers

When installing Windows 3.x or using communications software, ensure that DOS has enough hardware stack allocation by using the STACKS= command in config.sys. A typical setting is STACKS=9,256.

NOTE: With DOS and Windows using the FAT file system, fragmentation of files can occur regularly with a degradation of the PC’s disk performance. Use the DOS utlity DEFRAG to defragment the files on your disk on say a weekly basis to keep the performance up to standard.

The use of SMARTDRV is to be recommended, but should be tested to see just how much memory you need in the cache. Smartdrive can cache floppies and CD-ROMs. We run a test program called HDT.bas to show just how much Smartdrive can improve DOS.

DOS’ Limitations

What limitations ? Microsoft and Digital Research keep moving the goalposts! We have seen with MS-DOS 5.0 and DR-DOS 6.0 the enhancements of the use of extended memory, the hard disk compression of data and the use of Upper Memory Blocks. Several of the most popular utilities bought as add-on items, have been brought into the fold as DOS utilities. Mirror, Undelete, safe Format and a graphically oriented DosShell menu program have been added to the list of DOS programs. DOS 5.0 also sports a decent on-screen editor and QBasic a replacement for GWBasic.

DOS 6 from Microsoft was reported to be the last DOS that will run on an 80286. It was supplied with hard disk data compression, anti-virus support built-in and so on. DOS 7 was supposed to follow soon after and would only run on 80386 based machines or better.

DOS 6.2 is a free upgrade to DOS 6.0 and was brought about by the extensive use of DoubleSpace on old and unreliable disk drives. It has several safety features including ScanDisk, which will not allow you to install DoubleSpace until you have performed a surface scan of your hard disk using ScanDisk.

DOS 6.21 & DOS 6.22 came about as a result of the lawsuit between Stac and Microsoft. The result is that you can upgrade from 6.x to 6.22 for free. You can download DOS 6.22 step-up files from Microsoft and upgrade an existing DOS 6.x installation.

DOS 7 was going to provide most of the functionality of OS/2 and Windows NT for dos type programs without the graphical user interface. However it has been used as part of Windows 95. Using MSD or my program Equip, you can see the version of DOS reported as version 7.

Novell’s DR DOS has resurfaced recently and may become public domain. Check out the Caldera web page on the internet.

Memory Configuration

The High memory area

The designers of the 286 and 386 chips intended to support 1MB of addressable memory in real mode. However, the actual method used by these chips to address memory results in an additional 64KB of memory above 1MB. The address FFFF:FFFF translates to a physical address of 10FFEFH. On an 8086 this wraps to 0FFEFH. On a 286 or 386, 10FFEFH is an actual address past 1MB. The addresses FFFF:0000 to FFFF:FFFF allow an additional FFEFH (64KB -16) bytes of memory to be addressed.

If this High Memory Area (HMA) is available, MS-DOS 5 has the option of loading itself into it. While about 28KB of MS-DOS 5 (its data and some of its code), must be kept low for technical and compatibility reasons, most of it (about 45KB), can go into the high memory area, freeing up space in the low 640KB. The resident part of COMMAND.COM is also placed high. Since a 286 or 386 with at least 1MB of memory usually has physical memory at the HMA location, virtually all 286 and 386 users benefit from this feature.

The DOS=HIGH instruction in Config.sys makes DOS use the HMA.

Extended Memory Management (Himem.sys)

HIMEM.SYS is an XMS (Extended Memory Specification) driver that allows programs to use memory above 640K in a cooperative manner. Most of the time, the simple statement

```
DEVICE=C:\DOS\HIMEM.SYS
```

is enough to load it and get it working (assuming, of course, that it's stored in C:\DOS).

If DOS=HIGH appears in your CONFIG.SYS file, DOS is loaded into the first 64K of extended memory, the High Memory Area (HMA), rather than conventional memory.

If you receive an error message when HIMEM is installed or have problems running DOS in the HMA, you may need to install HIMEM.SYS with a /MACHINE switch that matches an A20 handler to the type of PC you're running it on. (An A20 handler is the routine HIMEM uses to manipulate the A20 address line to gain access to the HMA.) Your DOS manual details the /MACHINE codes that HIMEM.SYS supports.

On some EISA (Extended Industry Standard Architecture) computers, there may be more extended memory available than the system's Int 15h/88h BIOS call can detect. On these computers, HIMEM.SYS uses only the amount of extended memory detected. This can result in a large amount of unused memory.

You can have HIMEM.SYS use all available extended memory by using the /EISA option in the CONFIG.SYS file. For example, if HIMEM.SYS is located in your WINDOWS directory, you would use the following command line in the CONFIG.SYS file to take advantage of all available extended memory:

```
device=c:\windows\himem.sys /EISA
```

If you use the /EISA option, do not load any device drivers that use the Int 15h/88h BIOS call to allocate extended memory before loading HIMEM.SYS. If you do, your system may not work properly.

NOTE: If you must reserve some extended memory for device drivers or applications that use the Int 15h/88h BIOS call, you can specify the amount of memory in kilobytes that you want to reserve by using the /INT15= option. For example, if you want to reserve 128 kilobytes of extended memory, you would use the following command line in the CONFIG.SYS file:

```
                    device=c:\windows\himem.sys [/EISA] /INT15=128
```

NOTE: This has been removed from the MS-DOS 6.22 version of Himem.sys

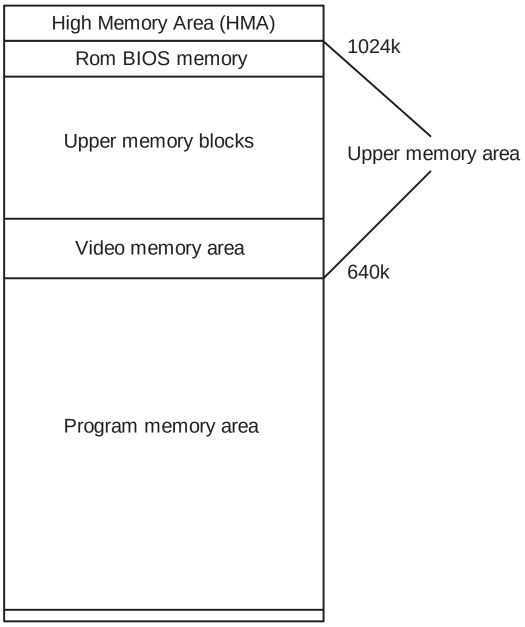
Upper Memory Blocks UMB's

In the area between 640KB and 1MB is an area called the Upper Memory Area (UMA). If free memory is available there, DOS 5 adds it to its memory arena. QEMM has been doing this for some time now using special loaders. MS-DOS 5 now supports the UMBs and can load both device drivers and programs high. This removes these device drivers and TSRs from the low 640KB freeing up more memory for programs. This is generally only available on machines using 80386 processors or higher as the processor is switched into protected mode and the memory area back-filled from extended memory.

Certain 80286 motherboards allow shadowing of the 384k byte area into ram, thus speeding up the performance of the processor. This allows such shareware utilities as "UMBDRV.SYS" to provide upper memory blocks as actual memory in this area.

The LOADHIGH and DEVICEHIGH commands which appeared with DOS 5, allows for loading terminate and stay resident programs, into this high memory area, freeing up more low memory for use by normal programs.

EMM386.EXE and Upper Memory



Upper memory is the region of your 80386 or 80486 computer's memory that is used by the machine system. The IBM PC designers must have said "we'll give the user ten times what he is used to, and use the rest for adapters and video". As a result the area above 640k is not normally available for use by DOS and is allocated to video and other adapters.

The parts of upper memory that are not used are called "upper memory blocks" (UMB). With MS-DOS 5 and up, you can use UMBs for running device drivers and other memory-resident programs. This makes more conventional memory available for running programs. The MS-DOS upper memory manager is EMM386.EXE. You need to use MS-DOS 4.x or higher for EMM386.EXE to access upper memory.

NOTE: EMM386 is not DOS version dependent and the latest (DOS 6.2) can be run on a machine with DOS 5.0.

For more complete information on using EMM386, see the EMM386 reference at the end of this section.

NOTE: Some programs cannot be moved to upper memory. These include HIMEM.SYS, EMM386.EXE, and MS-DOS system data.

Programs such as DOSKEY, SHARE, FASTOPEN, RAMDRIVE.SYS, console and other device drivers are good choices for loading into upper memory. You can also load your TSRs into high memory if your MS-DOS version supports it. This includes network programs such as IPX, NETX, LSL, IPXODI and hardware driver programs.

NOTE: The only way to find out if a program can run in upper memory is to try it. Some programs do not run properly in upper memory. If the program does not execute correctly, or if the system locks up, run it in conventional memory.

To run device driver programs in upper memory, you must include the following commands in your CONFIG.SYS file for loading HIMEM.SYS and EMM386.EXE:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
```

The DEVICE command for EMM386.EXE installs EMM386.EXE as an upper memory manager. The NOEMS option tells MS-DOS to run EMM386.EXE to manage upper memory only. Since NOEMS prevents EMM386.EXE from emulating expanded memory, use NOEMS only if your programs do not require expanded memory.

The NOEMS option for EMM386.EXE is the most efficient setting for the Windows operating system.

If you want to use EMM386.EXE both for the upper memory area manager and to emulate expanded memory, use : DEVICE=C:\DOS\EMM386.EXE RAM

NOTE: The Microsoft Windows operating system will be unable to allocate expanded memory to programs that need it if you specify the NOEMS option when installing EMM386.EXE. If you use such programs, use the RAM option (or no options) instead.

Ensure that HIMEM.SYS is loaded first by putting the DEVICE command for HIMEM.SYS before the DEVICE command for EMM386.EXE. EMM386 cannot run without the high memory manager in place. Also the DEVICE commands for HIMEM.SYS and EMM386.EXE must appear before any other DEVICE commands.

If MS-DOS runs in upper memory, your CONFIG.SYS file will have DOS=HIGH,UMB instead of DOS=UMB.

DOS 5 TIP: To get more upper memory area for use with network driver programs, use the include option to tell EMM386 to use the expanded memory frame area of E000 to EFFF. eg: DEVICE=C:\DOS\EMM386.EXE NOEMS **I=E000-EFFF**. This will add another 64k to the upper memory pool. MS DOS 6.x users can ignore this as their EMM386 will use this memory if it is available.
NOTE: This does not work on all machines.

Using MEM to check upper memory

To load programs into upper memory, check your memory layout by executing the MEM /c command. At the end of the output from the MEM /c command, note the size in the line Largest available upper memory block. Then look in the “Conventional Memory” section of the output and find the largest device driver or program that will fit into that upper memory block (UMB). Change the command in the CONFIG.SYS file for that device driver from DEVICE to DEVICEHIGH. For memory-resident programs, change the command in the AUTOEXEC.BAT file LOADHIGH (LH for brevity). Do this for one program at a time. You must restart your system each time.

Example MEM output using the /c option:

```
Modules using memory below 1 MB:

 Name Total = Conventional + Upper Memory
 -------- ---------------- ---------------- ----------------
 MSDOS 17,293 (17K) 17,293 (17K) 0 (0K)
 SETVER 800 (1K) 800 (1K) 0 (0K)
 HIMEM 1,168 (1K) 1,168 (1K) 0 (0K)
 EMM386 3,120 (3K) 3,120 (3K) 0 (0K)
 SMARTDRV 31,520 (31K) 2,496 (2K) 29,024 (28K)
 PROTMAN 128 (0K) 128 (0K) 0 (0K)
 COMMAND 4,720 (5K) 4,720 (5K) 0 (0K)
 win386 40,016 (39K) 9,744 (10K) 30,272 (30K)
 WIN 1,568 (2K) 1,568 (2K) 0 (0K)
 COMMAND 6,944 (7K) 6,944 (7K) 0 (0K)
 USPI14 8,704 (9K) 0 (0K) 8,704 (9K)
 MOUSE 17,056 (17K) 0 (0K) 17,056 (17K)
 DBLSPACE 39,472 (39K) 0 (0K) 39,472 (39K)
 NE2000 9,184 (9K) 0 (0K) 9,184 (9K)
 WORKGRP 4,400 (4K) 0 (0K) 4,400 (4K)
 DOSKEY 4,144 (4K) 0 (0K) 4,144 (4K)
 Free 607,264 (593K) 607,264 (593K) 0 (0K)

Memory Summary:

 Type of Memory Total = Used + Free
 ---------------- ---------- ---------- ----------
 Conventional 655,360 48,096 607,264
 Upper 142,256 142,256 0
 Reserved 393,216 393,216 0
 Extended (XMS) 15,586,384 14,537,808 1,048,576
 ---------------- ---------- ---------- ----------
 Total memory 16,777,216 15,121,376 1,655,840

 Total under 1 MB 797,616 190,352 607,264

 Largest executable program size 607,248 (593K)
 Largest free upper memory block 0 (0K)
 MS-DOS is resident in the high memory area.
```

[DOS 5.0]

If you get an error with one of the programs you have loaded into upper memory, or the program or device driver is still running in conventional memory after you restart your system, it may be that the largest UMB is not large enough. Some programs require more memory when they are loaded than when they are running. Try using the SIZE= option with the DEVICEHIGH command. Modify the DEVICEHIGH command in your CONFIG.SYS file to specify the hexadecimal size of the driver from the Size in Hex column of the MEM output, and restart your computer. For example, if the information in the Size in Hex column from the MEM command output for MOUSE.SYS is 39E0, you would put this statement in your CONFIG.SYS:

DEVICEHIGH **SIZE=39E0** C:\WIN3\MOUSE.SYS

The SIZE= option takes effect only if needed. If using the SIZE= option doesn't allow your program to run, or if your system locks up during startup or when running the program, it is likely that the program cannot run in upper memory. Change the DEVICEHIGH command to DEVICE and remove LOADHIGH commands one at a time until the program works correctly.

Some hardware programs might attempt to use upper memory after EMM386.EXE has determined this memory is available for running device drivers and programs. To avoid this, you can use the x= option when you load EMM386.EXE. This option prevents EMM386.EXE from allocating a specified range of upper memory for its use. For example, to prevent EMM386.EXE from using the addresses D000h through D7FFh (An ARCnet adapter card's memory buffer or a PCMCIA adapter on a notebook), you can include the following command in your CONFIG.SYS file:

```
DEVICE=C:\DOS\EMM386.EXE NOEMS X=D000-D7FF
```

[DOS 6.x]

With DOS 6 a new program called MEMMAKER arrived promising to automate the procedure of setting up UMB's and memory management.

Memory Configuration Fault Finding

If you think your computer is set up correctly to run device drivers and programs in upper memory, but nothing appears there when you use the MEM /c command, check the following:

1. [DOS 5.0] Make sure you are not running the Windows operating system version 3.x in 386-Enhanced mode when you execute the MEM command. The MEM command does not report the contents of upper memory when you are running the Windows operating system.

2. Your CONFIG.SYS file must contain the DOS=UMB or DOS=HIGH,UMB command.

3. The DEVICE command for EMM386.EXE in your CONFIG.SYS file must contain the NOEMS or RAM option. RAM is the default.

4. Your CONFIG.SYS file must contain a DEVICEHIGH command, or your AUTOEXEC.BAT file must contain the LOADHIGH command for each program you want to run in upper memory.

5. The DEVICE command for HIMEM.SYS must appear before the DEVICE command for EMM386.EXE; the DEVICE command for EMM386.EXE must appear before any DEVICEHIGH command in your CONFIG.SYS file.

Once programs are working successfully in upper memory, you can experiment to find the most efficient way to use available memory.

In general, load device drivers and programs in order of size, from largest to smallest. Do this because MS-DOS uses the largest remaining UMB, even if that program would fit into a smaller UMB. The optimal loading order depends on the sizes of programs you are loading and the sizes of available UMB.

Mem like all DOS programs responds to the useful "**/?**", but does have an undocumented option. This is "**/A**" which tells you how much of the high memory area is available.

Other operating systems

Windows 3.x and Windows for Workgroups

Windows 3.0 and later 3.10, now 3.11, has proved so incredibly popular that there were over 60 million users in December 1994. It is based on a 80286 style cooperative multitasking system, where programs release the cpu when waiting for user input or I/O to finish. The greatest change came about when version 3.0 included the “enhanced mode” operation specific to 80386 cpu’s. This also made use of the 386’s capability for “virtual 8086” operation and allows DOS programs to run concurrently with Windows programs. The system is not entirely crash proof and with buggy applications can do great damage to files and data, not to mention user confidence!

Windows 95/98

File Cabinet desktop interface. Recurring menus.
Runs with MS-DOS 7.0 integrated into the system. Boots directly into Win95.
Long file name support instead of the CP/M style filename.ext format.
Will run 32 bit NT programs. Now referred to as Win32 programs.
Network integrated into the system, as used in Windows for Workgroups.
Pre-emptive multitasking for 32 bit applications, not for 16 bit ones though.
Auto installs hardware into the system. Has Plug-n-Play support.

OS/2 2.1 - 3.0 “Warp Connect”

A fully pre-emptive multitasking 32 bit operating system.
Will run DOS, Windows 3.1 and OS/2 programs concurrently.
Graphical operating system.
Integrated networking has been added in latest version.

Windows NT (Now NT4 Server & Workstation)

A fully pre-emptive multitasking 32 bit operating system.
Will run DOS, Windows 3.1, Posix compatible UNIX programs and OS/2 programs concurrently.
Graphical operating system similar to Windows 95.
Integrated networking with enhanced security above WFW. 10 user licence for workstation.
Is available for other processors.
Can be used on multiprocessing platforms. This last has major implications for network servers as performance can be enhanced considerably by the addition of more processors.

There are of course lots of other operating systems. Unix, Linux, Pick etc. DOS used to have competition with DR DOS and IBM’s PC DOS. Unfortunately we cannot go into them all on this course.

Multitasking Systems

What is multitasking?

People always seem to think that multitasking is where the processor does several things at the same time. It isn't.

Multitasking is the ability to schedule multiple tasks for execution. It uses time slices to accomplish this faster than the human eye can perceive it happening.

A process is an executing program that owns resources. These resources include open files, shared memory, message queues, semaphores, pipe handles, and even static memory. A single processor machine uses a sequential multitasking operating system, and only one task actually executes at a time. A multiprocessor machine uses a parallel multitasking operating system, and multiple tasks can execute concurrently.

The operating system has a "scheduler" that controls what is executing and what has yet to execute. With OS/2 the smallest unit of execution that you can schedule to run is a "thread", which is simply a path of execution through a process. When a program is ready to execute, the operating system creates a thread to represent the process and schedules that thread for execution. This becomes thread1, or the "primary" thread. In a single-threaded application, the thread begins execution at the function main() and continues until the process terminates. This is a sequential algorithmic application. A process can have up to 512 threads associated with it under OS/2 version 1.3.

Windows vs OS/2

Windows does not run a "real" multitasking operating system, it runs as a "co-operative" multitasking system. When one process has idle time, eg: waiting for keyboard input, it releases the processor back to the operating system. If the program doesn't release the processor the other programs will "freeze". In a "pre-emptive" multitasking system the operating system can decide when to disconnect the processor from that task and to assign it to another.

Windows NT vs OS/2

Yes there is a difference between Windows 3.x and Windows NT. Windows 3.x is as I said above is a "co-operative multitasking" system, whereas Windows NT is not only a "pre-emptive" multitasking system, it is a multiprocessor operating system. OS/2 has not yet been made into a multiprocessor operating system, so NT may well be the ***correct future***. OS/2 is only available for Intel based processors and NT is available for RISC processors and will be marketed on multiprocessor (n * 486) type machines soon.

Section 3 Problem Areas

Hardware handling - STATIC

In winter here in the reef area, as much as 25,000 Volts can be generated by someone walking across a tile floor. It usually gives us quite a fright when we discharge ourselves into a door frame or even someone else. If you feel the discharge, it is over 3,500 Volts. Electronic equipment need only have a few hundred volts of static discharged into it for it to fail or be affected. The items usually hit first in this manner are the items we handle, such as mice, keyboards or keyboard interface chips on the motherboard. Also static doesn't necessarily destroy or stop the item working right away, it can fail an hour or two later.

To guard against damage always keep chips and cards as well as motherboards etc in an anti-static bag. In summertime the moisture content in the air is high allowing quick discharge of static. Even so keep the electronics in anti-static protection until assembly and earth yourself before handling, by touching a known earth point such as the chassis of a computer or the plate of a mains socket.

Radio Frequency Interference (RFI)

[Why they won't let you fly and work on your notebook at the same time.]

Aircraft still use VHF AM radio for communications and navigation. Frequencies used are between 116 and 132 MHz which has a high susceptibility to interference from computer equipment. Computers generate frequencies from a few kilohertz to over a 1000 MegaHertz. Aircraft use low power (10 Watts typically), as line of sight is guaranteed from 30000 feet.

So if you find yourself in PE instead of Cape Town on the SAA flight, don't say I didn't warn you.

The local authority for RFI is the SABS and most locally available equipment has been tested by them for compliance with their specifications governing RFI. Power supplies especially have come under close scrutiny as they are generally switch mode designs, running at 25 to 50 kiloHertz, with high power levels.

What happens when it goes wrong. Fault finding

Its Monday morning, and your favourite user has just switched on his PC. He was surprised by no display and whole cacophony of beeps and of course he phones you for an explanation/help/support.

The Power On Self Test internal to the BIOS ROM has tested the motherboard and most of the associated peripherals and has found a fault with one of them. Unfortunately BIOS designers follow completely different standards for error beeps and messages. So the next step is to find out what type of BIOS your user has in his PC.

The AMI BIOS is fairly common and to help you determine what has failed here is a list of their beeps and what causes them.

BEEPS	ERROR	What has happened
1	Refresh failure	The memory refresh circuitry on the motherboard has failed
2	Parity error	A parity error occurred in the first 64k block of main memory
3	Base 64kb memory failure	memory failed during power on self test in the first 64k
4	Timer not operational	Timer no 1 in the 8254 has failed.
5	Processor error	The CPU has failed its test
6	8042 A20 Gate error	The keyboard controller (8042) also controls the A20 line which controls access to memory above 1MB. The BIOS was not able to switch this line.
7	Processor exception interrupt error	The CPU has generated an exception interrupt internally during self test.
8	Display adapter failure	The display adapter either has faulty memory or a faulty controller chip. (6845)
9	Rom checksum error	The BIOS eprom has a fault which was detected by the POST routine.
10	CMOS shutdown register Read/Write error	The Shutdown register in the 146818 CMOS real time clock, has failed.

Lightning

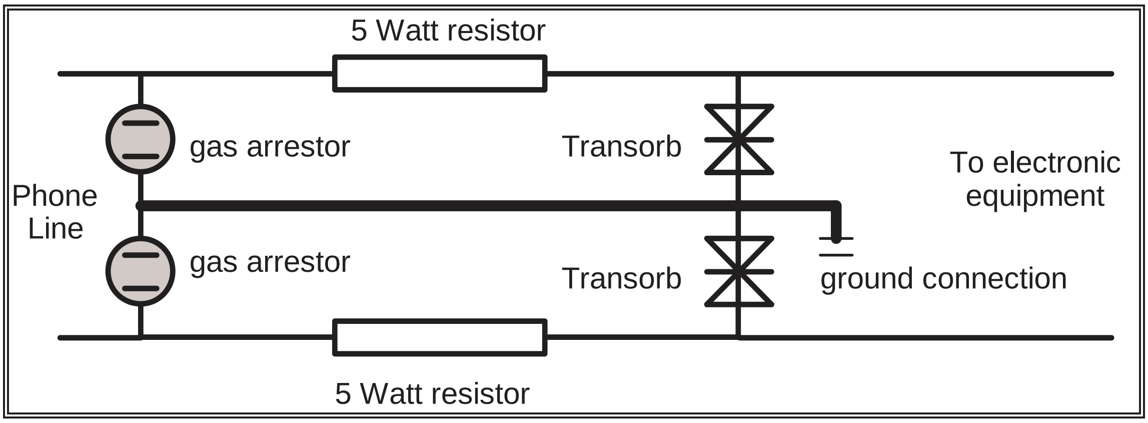
Unlike other places in the world, South Africa has a very high incidence of lightning with three to five times the average currents of elsewhere. There are not many places similarly afflicted with lightning like ours. Denver Colorado and Mexico City have similar lightning problems. So it is understandable that equipment designers overseas do not incorporate sufficient protection into their products.

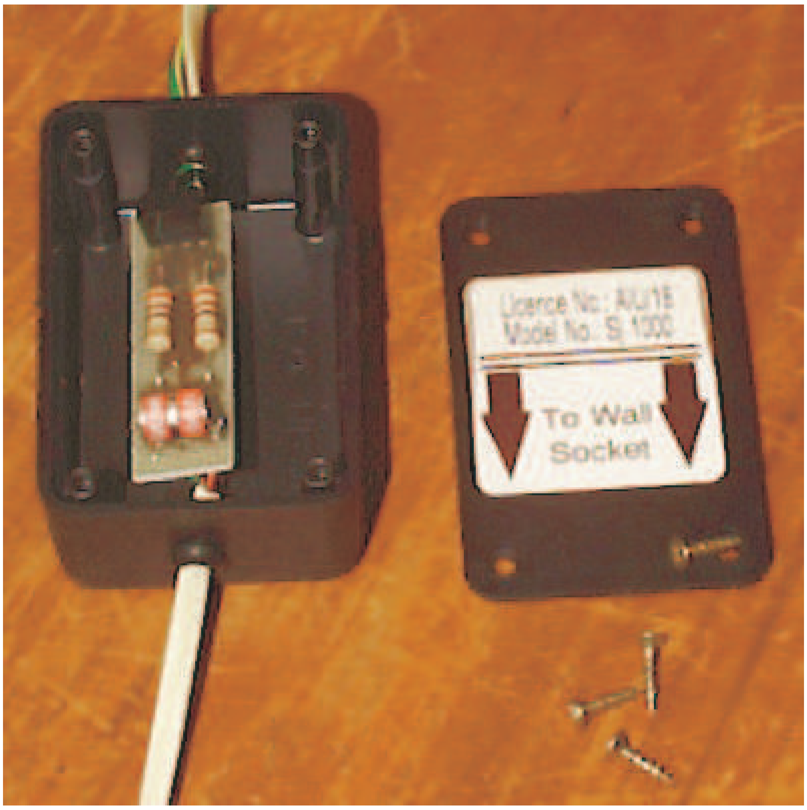
The average reef lightning strike has a current of approximately 150,000 Amperes flowing in it. It also is a multiple strike, with a series of flashes occurring in a fraction of a second. The electrical effects of the strike can be observed tens of kilometres away and heard on radio hundreds of kilometres away. Closer to the strike, say 500 metres, the magnetic field generated induces high voltages in any conducting material.

At 500 metres distance a length of wire, 10 metres in length will have about 33,000 Volts induced into it! This can have a serious effect on any electronic component connected to that wire. Modems and Fax machines are nearly always left connected to one of the longest pieces of wire known, the telephone network.

The Post Office has a recommended lightning scheme called a five point scheme for equipment connected to the telephone network. It consists of two gas arrestors, two 5 Watt resistors (wire wound as other types tend to explode) and two Transorb devices. The Transorbs are semiconductor devices and respond in microseconds to the overvoltage, conducting high current to ground. The voltage rises across the two resistors sufficiently to ionise the gas inside the gas arrestors, which then "strike", conducting much higher currents to ground as well.

The Post Office "Five Point Scheme"





A lightning protection unit

Backup

(Backup should be simple so that users have no excuse not to back up their data.)

A simple batch file to backup all DOC files on C:

```
Cls
Echo off
:LOOP
Rem Could be B: or a Network drive.
Echo Put blank disk in drive A:
Rem this puts “Press a key when ready...” on screen.
Pause
Rem XCOPY supports errorlevels and can search subdirectories.
XCOPY C:*.DOC A: /E /S /M
Rem ErrorLevel 4 is disk full
If ErrorLevel == 4 goto LOOP
Echo All done!
```

PkZip options for making a full and incremental backup

Full backup

Use the -a+ option to create a full backup of files on your hard disk or network disk.

For example:

```
C:\> PkZip a:fullback -&s -a+
```

This will backup the entire hard disk to a multi-disk Zip file called Fullback.zip on a set of disks in the a: drive

Incremental backup

-a+ Turn off archive attribute of added files.

By using this option you can create a complete backup of your disk, while clearing the archive attributes to make the way for incremental archiving.

Incremental archiving makes use of the archive attribute to take only the files which have been modified since the last backup. In order for this process to work smoothly you must first have a complete backup and a clearing of the archive attribute for all files.

```
 C:> pkzip f:backup.zip -a+ -rp
```

Windows troubleshooting

General Protection Fault

The Intel 80286, 80386, and 80486 CPUs can detect when a program does something wrong. The most common problems are stack faults, invalid instructions, divide errors (divide by zero), and general protection faults. These generally indicate non-standard code in an application.

The following faults can occur in a Windows application, in Windows itself, or in a Windows device driver (for example, a video display driver).

General Protection Fault (Interrupt 13)

All protection violations that do not cause another exception cause a general protection exception. This includes, but is not limited to:

Exceeding the segment limit when using the CS, DS, ES, FS, GS segments. This is a very common bug in programs, usually caused by miscalculating how much memory is required in an allocation.

Transferring execution to a segment that is not executable (for example, jumping to a location that contains garbage).

Writing to a read-only or a code segment.

Loading a bad value into a segment register.

Using a NULL pointer. A value of 0 is defined as a null pointer. In protected mode, it is always invalid to use a segment register that contains 0.

Stack Fault (Interrupt 12)

Reasons for a stack fault are:

An instruction tries to access memory beyond the limits of the Stack segment (POP, PUSH, ENTER, LEAVE, or a stack relative access: MOV AX, [BP+6]).

Loading the SS register with a selector marked not present, but otherwise valid (shouldn't happen under Windows).

Stack faults are always fatal to the current application in Windows.

Invalid Instruction (Interrupt 6)

The CPU detects most invalid instructions, and generates an Interrupt 6. This is always fatal to the application. This should never happen, and is usually caused by executing data instead of code.

Divide Error (Interrupt 0)

This is caused when the destination register cannot hold the result of a divide operation. It could be divide by zero, or divide overflow.

Segment Load Failure

Windows loads programs into memory in segments (64K blocks of memory) marked fixed, movable, or discardable. Windows can unload discardable segments when it needs more memory for other applications.

When an application needs to access a segment of code that is not in memory, Windows loads it from the hard disk. If Windows cannot load the segment for some reason, it generates a "Segment Load Failure" error.

"Segment Load Failure" errors are generated if a discarded code segment has been damaged or moved, or if Windows lacks the system resources to load a file, as in the case when there are not enough MS-DOS file handles.

"Segment Load Failure" errors may also be caused by corrupt binary files. Reinstalling Windows or Windows for Workgroups into a new directory or over itself may alleviate the error.

The Win /B option and others

Launch Windows with switches /D:[F][S][V][X]

Syntax: WIN /D:[F][S][V][X]]

/D Used for troubleshooting when Windows does not start correctly.
:F Turns off 32-bit disk access. Equivalent to SYSTEM.INI [386enh] setting: 32BitDiskAccess=FALSE.
:S Specifies that Windows should not use ROM address space between F000:0000 and 1 MB for a break point. Equivalent to SYSTEM.INI [386enh] setting: SystemROMBreakPoint=FALSE.
:V Specifies that the ROM routine handles interrupts from the hard drive controller. Equivalent to SYSTEM.INI [386enh] setting: VirtualHDIRQ=FALSE.
:X Excludes all of the adapter area from the range of memory that Windows scans to find unused space. Equivalent to SYSTEM.INI

Section 4 Comedy in computing - keeping your sense of humour.

Some users would try the patience of Job. Even well adjusted technical personnel can be driven to the point of violence by apparently ignorant users. It comes down to two things. 1) Educate the users and provide them with first line assistance. On-line manuals/help data files etc. 2) Try not to lose your sense of humour.

Users don't normally go out of their way to ruin your day! Nor do they make up stories to tell you. The PC can get even the most computer literate person frantic or mystified.

South Africans have a unique phrase describing this effect. “Sense of humour failure”. Which I think is very apt.

Section 5 Information, we want information.

Looking after your information - support, diagnostics, tools.

Do you keep copies of the driver disks ? Do you have them stored and available for installation from the network?

Do you document your test procedures? You really don’t have an excuse as you probably have your own PC with at the very least a simple word processor like Windows Write.

Did you know that you can “record” a series of actions in Windows and replay them continuously for test purposes? You can use the Windows accessory Recorder to record your keystrokes. Then you can be “testing” the PC whilst getting on with something else.

Documentation and aids to documentation

MSD. MSD can also be made to operate automatically and to print its report to a text file. Check MSD /?

I have provided a Windows Help file as part of the software. This file provides all the connection details for most of the PC connections as well as some hard disk details.

External sources - the Internet

If anything has captured the imagination in networking recently it must be the Internet. Even accountants know about it! What started as a defence project in the Cold War Years, has become a global network of interconnected computer systems. More information that you can ever use can be obtained about any subject.

Its growth over the last few years has been phenomenal as the ease of use of the browser software has improved. It has the potential to replace the postal services, the data transport services and the telephone service. This will upset the present monopolistic situation for quite a few post services.

To gain access to the network you need a modem, preferably a 14,400 or 28,800 bps, a service provider with dial-up facilities and preferably Windows based “browsing” software.

for example:	www.conner.com	for Conner drive information
	www.seagate.com	for Seagate drive information
	www.microsoft.com	for Microsoft information
	www.hp.com	for Hewlett Packard
	www.sdd.com	“ “
	www.ibm.com	
	www.microsoft.com	for software and information
	www.intel.com	for info about cpu’s, net cards etc.

Packetisation of data - X25

The CCITT X.25 protocol describes how data communications devices like bridges and routers package data into specially addressed and configured packets. It also details how the value-added networks (VAN) computers, acting as switches, route these packets through the network toward their destinations.

X.25 begins with the packet, an electronic envelope that is typically 128 bytes long. The protocol allows for a variety of packet types to be transmitted, of which only some may contain some of the actual data being sent.

A packet-switching node (PSN), which may be a PC on a network, runs software with accompanying hardware called a PAD (packet assembler/disassembler). The PAD creates and transmits packets; it also receives incoming data from the public data network and breaks down the packets into usable data. Although the PAD generally comes in the form of an adapter board with its own processing capability and software, Hayes Microcomputer Products incorporates the required PAD circuitry into its Smartmodem V-Series models. The packets to be relayed through the PDN are interleaved on a circuit with packets transmitted by other PDN customers to make the best use of whatever bandwidth is available hence the word "public" in PDNs.

In order for a PAD to establish an X.25 circuit and pass information, it must follow a specific order of packet exchange. First, the PAD sends a call request packet to the requested host (in this case, the PDN), which issues a call accepted packet to grant permission for further exchange. Once the call is set up, data is exchanged in the form of packets that include addressing information as well as the actual data comprising the purpose of the transmission.

Throughout the process, every packet sent has a clearly specified structure, with every field defined. A call clear packet, sent by the PAD, initiates the termination of a call; the exchange ends when the PAD receives acknowledgement in the form of a clear confirmation packet. Other packet types that add to X.25's strength include special reset and restart packets, along with other X.25-specified packets such as the diagnostic packet, which checks the status and efficiency of the network. Overall, they help make the X.25 protocol both versatile and easy to use.

At the heart of the value-added network itself is a group of networked computers (often depicted on diagrams as a cloud) with a variety of connections leading in and out. The image is accurate, especially for PDNs, since the minicomputers operating within are complex both in their connections and in the resulting types of services that this setup makes available to customers.

PDN access services can be readily lumped into two categories: leased access and dial-up access. Leased access lines, often synchronous for greater throughput, can carry data at signalling speeds of 19.2, 56, or 64 kilobits per second (or even 1.544 Mbps) using full X.25 protocols from the customer's network to the cloud. This particularly expensive connection is best reserved for the busy headquarters segment of the LAN that serves as the focal point for traffic with other LAN segments typically in branch offices. The other type of connection, a dial-up telephone line, is convenient and relatively inexpensive for users. This often asynchronous (synchronous modems are not inexpensive) connection does not provide full X.25 data handling between the computer and the X.25 network. Error correction between the customer and the PDN, in this case, is provided only by the asynchronous modem transmitting the data to the PDN. Typically, dial-up services range from 300 to 9,600 bps, although some services allow speeds up to 19.2 Kbps and even 56 Kbps.

Section 6 The Standard PC Specification

Processor of choice (best bang for the buck)

Intel Pentium 233MHz MMX. Certainly adequate for the "Office" suite and Windows for Workgroups with memory from 16MB up. This will change soon to a Pentium II 266MHz as the stocks of ‘classic’ Pentiums dries up.

Cpu manufacturer AMD or Intel or Cyrix/TI

Performance difference is not really measurable so either manufacturer will be ok. Cost will then play a part in your decision. Cyrix pushed Intel into releasing the Pentium 200MHz as their 686 processor running at 150MHz out-performed the P166. Compatibility is not known to be a problem, however some game players reckon the Pentium is best as the FPU gives better graphics performance.

Memory

16MB should be considered a suitable starting point as most software will operate adequately in 16MB. However (later) facilities will need at least 32MB for such as OLE or database and spreadsheet data transfer. With memory at an all time low price, it makes sense to invest in at least 32MB.

Hard Disk

Disk capacities have been increasing rapidly the last 12 months and 1.2GB appears to be the average. However many would be "happy" with less. Every CD based software package seems to want a nibble at your hard disk for index storage or main program. So again a cost based decision here. Bear in mind a second, third and fourth large drive can be added at a later stage.

Disk performance

Various drives are available from different makers with various specifications. Conner, Seagate, Quantum and Maxtor all make good drives with 2 to 3 years guarantee. Evaluation of performance can be carried out using standard "benchmarks" such as coretest.

We generally supply Fujitsu as these have given extremely high reliability coupled with above average performance. The usual disk drive interface is IDE or AT Bus now Enhanced IDE with an interface to the Local Bus.

Video Display Adapter

Only PCI and Local Bus (VL) provide the performance needed for Windows or OS/2 graphical operating systems. Local Bus adapters also have accelerator capabilities. The cards we have tried and tested as adequate are Western Digital and Chips and Technologies.

PCI can provide similar performance to VL/Local Bus at slightly lower cost. PCI motherboards cost slightly more as they are more complex than local bus, which compensates for the lower complexity of PCI adapters.

Display Monitor

14 inch monitors (measured diagonally) are ok for general viewing of 80 by 25 lines of text or 640 by 480 pixels. For 'Super' VGA a 14 inch monitor needs to be non-interlaced (NI). The dot pitch should also be 0.28mm.

Serial ports (Com1 to 4)

Use of the "OLD" style 8250/16450 UARTS should be discouraged. The correct type should be the 16550A UART with its 16 character transmit and receive buffers. It is possible to have four serial ports with the use of the 'spare' IRQ5 and 7 interrupt request lines. However these may not be available on a new machine with the new printer ports installed. Windows will support four interrupt driven serial ports.

Parallel Ports (Lpt1 to 3)

Again the use of a standard printer port may not be the best option. Most new multi I/O cards have both super serial ports and extended printer ports. The printer ports are now bi-directional as well, allowing the HP printers more functionality with their software.

Sound Card (Microphone, line input, line output, speaker output)

The sound card usually used these days is the Sound Blaster, having stereo microphone inputs as well as stereo sound outputs. The Sound Blaster has become the industry standard. It also can be plugged into the Hi Fi using the Line inputs and outputs. The ‘clones’ are quite compatible and can save cost.

Power Supply

Reliability is the main concern here. The name "Tiger" springs to mind here as they are used by big name pc manufacturers. Old power supplies can be replaced cheaply and quickly as only plugs and sockets are used. Drive sockets fit only one way round. The motherboard pair of connectors can be connected wrongly, but if you remember "black to black", then you won't get it wrong.

Section 7 New developments in computing

New developments in PC technology

up to 20GB IDE drives promised soon.

Electronic camera that stores pictures and downloads to your pc. Expensive at present (R2000 +/-) but have come down in price.

Teletext adapters for your pc. Teletext has enjoyed good support in Europe, but not in SA. Teletext/Ceefax services include news, weather, share prices and programme details. Information can be distributed rapidly to the entire tv broadcast area. Talk to the SABC or MNET special technology departments for further details.

Virtual reality for training drivers. Kawasaki have bought 1000 driving simulators a deal worth 120 Million Pounds.

ISDN to finally arrive. ISDN offers a much faster communication speed than standard telephone lines because it replaces the analogue telephone system with a digital system. For example, a standard telephone line can typically transfer data with a modem at a speed of 9600 bits per second (bps); ISDN communicates without a modem at speeds of 64 or 128 kilobits per second.

Look for ASDL modems from Alcatel. If Telkom allow their use, we may see 4Mbits or more on ordinary telephone lines. These modems only require additional equipment at the exchange.

The UK government has identified the science and technology priorities it wants industry to pursue as :

- Software engineering for complex tasks.
- Communicating with machines.
- Environmentally sustainable technology.
- Genetics and biomolecular engineering.
- Bioformatics: Analysing biological information.
- Gene-based advice on health and life style.
- Optical technology using light pulse.
- Management and business process engineering.
- Sensors for measurement and analysis of data.
- Security and privacy technology.
- Remote working and multimedia.

[source Daily Mail]

APPENDIX A : Other Number Systems

A Number is comprised of digits each representing multiples of descending powers of the base of the number system.

eg: 618_{10} = (6 * 10^2) + (1 * 10^1) + (8 * 10^0)

- - -

digit base or radex exponent or index

Decimal system :

10^3	10^2	10^1	10^0
			0
			1
			:
			9
		1	0
		1	1
		:	:
		1	9
		2	0
		:	:
		2	9
		3	0

Hexadecimal system :

16^3	16^2	16^1	16^0
			0
			1
			2
			3
			4
			5
			6
			7
			8
			9
			A
			B
			C
			D
			E
			F
		1	0
		1	1
		1	2
		1	3
		:	:
		1	F
		2	0
		:	:
		F	F
	1	0	0
	1	0	1
	1	0	2

Binary system :

2^3	2^2	2^1	20
			0
			1
		1	0
		1	1
	1	0	0
	1	0	1
	1	1	0
	1	1	1
1	0	0	0
1	0	0	1

APPENDIX B : Screen modes reference

Here is a summary of the standard video modes supported by the BIOS or by the adapter's BIOS attachment. Other modes can be programmed by software and some interesting displays achieved as a result.

MDPA, CGA, Hercules, Olivetti, EGA, VGA, or MCGA Adapters

SCREEN 0: Text mode only
40 x 25, 40 x 43, 40 x 50, 80 x 25, 80 x 43, or 80 x 50 text format,
8 x 8 character box (8 x 14, 9 x 14, or 9 x 16 with EGA or VGA)
16 colors assigned to any of 16 attributes (with CGA or EGA)
64 colors assigned to any of 16 attributes (with EGA or VGA)
Depending on the text resolution and adapter, 8 video memory pages (07), 4 pages (03), 2 pages (01), or 1 page (0)

CGA, EGA, VGA, or MCGA Adapters

SCREEN 1: 320 x 200 graphics
40 x 25 text format, 8 x 8 character box
16 background colors and one of two sets of 3 foreground colors assigned using COLOR statement with CGA
16 colors assigned to 4 attributes with EGA or VGA
1 video memory page (0)
SCREEN 2: 640 x 200 graphics
80 x 25 text format, 8 x 8 character box
16 colors assigned to 2 attributes with EGA or VGA
1 video memory page (0)

Hercules, Olivetti, or AT&T Adapters

SCREEN 3: Hercules adapter required, monochrome monitor only
720 x 348 graphics
80 x 25 text format, 9 x 14 character box
Usually 2 video memory pages (01); 1 page (0) if a second color display adapter is installed
PALETTE statement not supported
Invoke the Hercules driver MSHERC.COM before using screen mode 3
SCREEN 4:
Supports Olivetti Personal Computers models M24, M240, M28, M280, M380, M380/C, and M380/T and AT&T Personal Computers 6300 series
640 x 400 graphics
80 x 25 text format, 8 x 16 character box
1 of 16 colors assigned as the foreground color (selected by the COLOR statement in BASIC); background is fixed at black
1 video memory page (0)
PALETTE statement not supported

EGA or VGA Adapters

SCREEN 7: 320 x 200 graphics
40 x 25 text format, 8 x 8 character box
Assignment of 16 colors to any of 16 attributes
If 64K EGA adapter memory, 2 video memory pages (01); otherwise, 8 pages (07)
SCREEN 8: 640 x 200 graphics
80 x 25 text format, 8 x 8 character box
Assignment of 16 colors to any of 16 attributes
If 64K EGA adapter memory, 1 video memory page (0); otherwise, 4 pages (03)
SCREEN 9: 640 x 350 graphics
80 x 25 or 80 x 43 text format, 8 x 14 or 8 x 8 character box
16 colors assigned to 4 attributes (64K adapter memory), or 64 colors assigned to 16 attributes (more than 64K adapter memory)
If 64K EGA adapter memory, 1 video memory page (0); otherwise, 2 pages (01)

EGA or VGA Adapters, Monochrome Monitor Only

SCREEN 10: 640 x 350 graphics, monochrome monitor only
80 x 25 or 80 x 43 text format, 8 x 14 or 8 x 8 character box
Up to 9 pseudocolors assigned to 4 attributes
2 video memory pages (01), 256K adapter memory required

VGA or MCGA Adapters

SCREEN 11 (VGA or MCGA)
640 x 480 graphics
80 x 30 or 80 x 60 text format, 8 x 16 or 8 x 8 character box
Assignment of up to 256K colors to 2 attributes
1 video memory page (0)
SCREEN 12 (VGA)
640 x 480 graphics
80 x 30 or 80 x 60 text format, 8 x 16 or 8 x 8 character box
Assignment of up to 256K colors to 16 attributes
1 video memory page (0)
SCREEN 13 (VGA or MCGA)
320 x 200 graphics
40 x 25 text format, 8 x 8 character box
Assignment of up to 256K colors to 256 attributes
1 video memory page (0)

NOTE: The 640*480 256 Colour mode, the 800*600 SVGA mode does not feature here as these were added later. No standard existed for software to interface to these modes and specific drivers had to be written for the cards. Later a VESA standard was produced which supports these and the other enhancements to the display.

Appendix C: EMM386.EXE Reference

Provides access to the upper memory area and uses extended memory to simulate expanded memory. This device driver must be loaded by a <DEVICE> command in your CONFIG.SYS file and can be used only on computers with an 80386 or higher processor.

EMM386 uses extended memory to simulate expanded memory for programs that can use expanded memory. EMM386 also makes it possible to load programs and device drivers into upper memory blocks (UMBs).

Syntax (Config.sys)

```
 DEVICE=[drive:][path]EMM386.EXE [ON|OFF|AUTO] [memory] [MIN=size]
 [W=ON|W=OFF] [Mx|FRAME=address|/Pmmmm] [Pn=address] [X=mmmm-nnnn]
 [I=mmmm-nnnn] [B=address] [L=minXMS] [A=altregs] [H=handles] [D=nnn]
 [RAM=mmmm-nnnn] [NOEMS] [NOVCPI] [HIGHSCAN] [VERBOSE]
 [WIN=mmmm-nnnn] [NOHI] [ROM=mmmm-nnnn] [NOMOVEXBDA] [ALTBOOT]
```

Parameters

[drive:][path]
Specifies the location of the EMM386.EXE file.

[ON|OFF|AUTO]
Activates the EMM386 device driver (if set to ON), or suspends the EMM386 device driver (if set to OFF), or places the EMM386 device driver in auto mode (if set to AUTO). Auto mode enables expanded-memory support and upper memory block support only when a program calls for it. The default value is ON. Use the EMM386 command to change this value after EMM386 has started.

memory
Specifies the maximum amount of extended memory (in kilobytes) that you want EMM386 to provide as expanded/Virtual Control Program Interface (EMS/VCPI) memory. This amount is in addition to the memory used for UMBs and EMM386 itself. Values for memory are in the range 64 through the lesser of either 32768 or the amount of extended memory available when EMM386 is loaded. The default value is the amount of free extended memory. If you specify the NOEMS switch, the default value is 0. EMM386 rounds the value down to the nearest multiple of 16.

Switches

MIN=size
Specifies the minimum amount of EMS/VCPI memory (in kilobytes) that EMM386 will provide, if that amount of memory is available. EMM386 reserves this amount of extended memory for use as EMS/VCPI memory when EMM386 is loaded by the DEVICE=EMM386.EXE command in your CONFIG.SYS file. EMM386 may be able to provide additional EMS/VCPI memory (up to the amount specified by the MEMORY parameter) if sufficient XMS memory is available when a program requests EMS/VCPI memory. Values are in the range 0 through the value specified by the MEMORY parameter. The default value is 256. If you specify the NOEMS switch, the default value is 0.

If the value of MIN is greater than the value of MEMORY, EMM386 uses the value specified by MIN.

W=ON|W=OFF
Enables or disables support for the Weitek coprocessor. The default setting is W=OFF.

Mx
Specifies the address of the page frame. Valid values for x are in the range 1 through 14. The following list shows each value and its associated base address in hexadecimal format:

1 => C000h 8 => DC00h
2 => C400h 9 => E000h
3 => C800h 10 => 8000h
4 => CC00h 11 => 8400h
5 => D000h 12 => 8800h
6 => D400h 13 => 8C00h
7 => D800h 14 => 9000h

Values in the range 10 through 14 should be used only on computers that have 512K of memory.

FRAME=address
Specifies the page-frame segment base directly. To specify a specific segment-base address for the page frame, use the FRAME switch and specify the address you want. Valid values for address are in the ranges 8000h through 9000h and C000h through E000h, in increments of 400h. To provide expanded memory and disable the page frame, you can specify FRAME=NONE; however, this may cause some programs that require expanded memory to work improperly.

/Pmmmm
Specifies the address of the page frame. Valid values for mmmm are in the ranges 8000h through 9000h and C000h through E000h, in increments of 400h.

Pn=address
Specifies the segment address of a specific page, where n is the number of the page you are specifying and address is the segment address you want. Valid values for n are in the range 0 through 255. Valid values for address are in the ranges 8000h through 9C00h and C000h through EC00h, in increments of 400h. The addresses for pages 0 through 3 must be contiguous in order to maintain compatibility with version 3.2 of the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS). If you use the Mx switch, the FRAME switch, or the /Pmmmm switch, you cannot specify the addresses for pages 0 through 3 for the /Pmmmm switch.

X=mmmm-nnnn
Prevents EMM386 from using a particular range of segment addresses for an EMS page or for UMBs. Valid values for mmmm and nnnn are in the range A000h through FFFFh and are rounded down to the nearest 4-kilobyte boundary.

The X switch takes precedence over the I switch if the two ranges overlap.

I=mmmm-nnnn
Specifies a range of segment addresses to be used (included) for an EMS page or for UMBs. Valid values for mmmm and nnnn are in the range A000h through FFFFh and are rounded down to the nearest 4-kilobyte boundary.

The X switch takes precedence over the I switch if the two ranges overlap.

B=address
Specifies the lowest segment address available for EMS "banking" (swapping of 16-kilobyte pages). Valid values are in the range 1000h through 4000h. The default value is 4000h.

l=minXMS
Ensures that the specified amount (in kilobytes) of extended memory will still be available after EMM386 is loaded. The default value is 0.

A=altregs
Specifies how many fast alternate register sets (used for multitasking) you want to allocate to EMM386. Valid values are in the range 0 through 254. The default value is 7. Every alternate register set adds about 200 bytes to the size in memory of EMM386.

H=handles
Specifies how many handles EMM386 can use. Valid values are in the range 2 through 255. The default value is 64.

D=nnn
Specifies how many kilobytes of memory should be reserved for buffered direct memory access (DMA). Discounting floppy-disk DMA, this value should reflect the largest DMA transfer that will occur while EMM386 is active. Valid values for nnn are in the range 16 through 256. The default value is 16.

RAM=mmmm-nnnn
Specifies a range of segment addresses to be used for UMBs and also enables EMS support. If you do not specify a range, EMM386 uses all available adapter space to create UMBs and a page frame for EMS.

NOEMS
Provides access to the upper memory area but prevents access to expanded memory.

NOVCPI
Disables support for VCPI applications. This switch must be used with the NOEMS switch. If you specify the NOVCPI switch without specifying the NOEMS switch, EMM386 does not disable VCPI support. If you specify both switches, EMM386 disregards the MEMORY parameter and the MIN switch. Disabling support for VCPI applications reduces the amount of extended memory allocated.

HIGHSCAN
Specifies that EMM386 use an additional check to determine the availablity of upper memory for use as UMBs or EMS windows. On some computers, specifying this switch may have no effect or cause EMM386 to identify upper memory areas as available when they are not. As a result, your computer might stop responding.

VERBOSE
Directs EMM386 to display status and error messages while loading. By default, EMM386 displays messages only if it encounters an error condition. You can abbreviate VERBOSE as V. (To display status messages without adding the VERBOSE switch, press and hold the ALT key while EMM386 starts and loads.)

WIN=mmmm-nnnn
Reserves a specified range of segment addresses for Windows instead of for EMM386. Valid values for mmmm and nnnn are in the range A000h through FFFFh and are rounded down to the nearest 4-kilobyte boundary.

The X switch takes precedence over the WIN switch if the two ranges overlap. The WIN switch takes precedence over the RAM, ROM, and I switches if their ranges overlap.

[NOHI]
Prevents EMM386 from loading into the upper memory area. Normally, a portion of EMM386 is loaded into upper memory. Specifying this switch decreases available conventional memory and increases the upper memory area available for UMBs.

[ROM=mmmm-nnnn]
Specifies a range of segment addresses that EMM386 uses for shadow RAM random-access memory used for read-only memory (ROM). Valid values for mmmm and nnnn are in the range A000h through FFFFh and are rounded down to the nearest 4-kilobyte boundary. Specifying this switch may speed up your system if it does not already have shadow RAM.

[NOMOVEXBDA]
Prevents EMM386 from moving the extended BIOS data from conventional memory to upper memory.

[ALTBOOT]
Specifies that EMM386 use an alternate handler to restart your computer when you press CTRL+ALT+DEL. Use this switch only if your computer stops responding or exhibits other unusual behavior when EMM386 is loaded and you press CTRL+ALT+DEL.

Syntax: (Dos prompt)

EMM386 [ON|OFF|AUTO] [W=ON|W=OFF]

To display the current status of EMM386 expanded-memory support, simply type EMM386 at the DOS prompt.

Parameters

ON|OFF|AUTO
Activates the EMM386 device driver (if set to ON), or suspends the EMM386 device driver (if set to OFF), or places the EMM386 device driver in auto mode (if set to AUTO). Auto mode enables expanded-memory support only when a program calls for it. The default value is ON.

W=ON|W=OFF
Enables (if set to W=ON) or disables (if set to W=OFF) Weitek coprocessor support. The default value is W=OFF.

Appendix D: The 8250 register set

Receive data register - at base address + 0 Read

7	6	5	4	3	2	1	0
data	data	data	data	data	data	data	data

Transmit data register - at base address + 0 Write

7	6	5	4	3	2	1	0
data	data	data	data	data	data	data	data

Interrupt enable register - at base address + 1 Write

7	6	5	4	3	2	1	0
0	0	0	0	enable modem status int.	enable Line status int	enable transmit register empty int.	enable receive data ready int.

Interupt identifier register - at base address + 2 Read

7	6	5	4	3	2	1	0
0	0	0	0	x	y	z	Interrupt pending

x	y	z	
0	0	0	modem status changed causing interrupt
0	0	1	transmit data register empty causing interrupt
0	1	0	receive data register full causing interupt

Line control register - at base address + 3

7	6	5	4	3	2	1	0
DLAB	Set break	Stick parity	Even parity	Parity enable	number of stop bits	wa	wb

wa	wb	
0	0	5 bit data
0	1	6 bit data
1	0	7 bit data
1	1	8 bit data

Modem control register - at base address + 4 Write

7	6	5	4	3	2	1	0
0	0	0	internal Loopback	Set Out2 enables Interrupts	Out1	Set RTS	Set DTR

Line status register - at base address + 5 Read

7	6	5	4	3	2	1	0
	Tx shift register empty	Tx holding register empty	Break received	Framing error	Parity error	Overflow error	Receive data ready

Modem status register - at base address + 6Read

7	6	5	4	3	2	1	0
Carrier detect	Ring indicator	Data set ready	Clear to send	Carrier detect changed	Ring indicator changed	Data set ready changed	Clear to send changed

Appendix E: PC AND XT Bus signal lines

Lines	Description
A0-A19	Twenty system address lines all are used to address memory. During I/O cycles, only the lowest 16 I/0 lines are actually used.
D0-D7	Eight bidirectional data lines.
ALE	Address latch enable. This signal goes high to indicate that a valid address is present on A0-A19 during a memory access.
IRQ2-IRQ7	Six maskable interrupt request lines.
DRQl -DRQ2	DMA request and acknowledge lines. There's no DRQ0 on the bus
DACK1-DACK3	DMA channel 0 is used for DRAM refresh on the PC and the XT.
IO CHRDY	A signal used by a memory or peripheral board to generate wait states.
IORD, IOWR, SMEMR, SMEMW	I/O and memory read and write strobes.
OSCA	14.31818-MHz clock used by some video boards. It's not synchronised with respect to the rest of the bus.
CLK	The bus clock signal (4.77 MHz in the original PC and proportionately faster on later machines). This clock is synchronised with respect to the read and write strobes.
AEN ,TC	Address enable and terminal count. These control signals are used during DMA cycles.
IO CHK	Alerts the processor to parity and other errors via a nonmaskable interrupt.
RESET DRV	Indicates that the system is being reset.
+5VDC, -5VDC, +12VDC, -12VDC, GND	Power-supply rails.

Because of its simple design, each interrupt line can be used by only one adapter card.

Additional Signal Lines for the 16-bit ISA bus

Lines	Descrlptlon
D8-D15	The eight new data lines.
SBHE	System bus high enable, which indicates when these data lines are being used.
IRQ10-12,	More interrupt lines. IRQ13 is absent because that interrupt is reserved for IRQ14-15 the maths coprocessor.
DRQ0, DACK0,	More DMA control lines for new DMA channels.
DRQ5-DRQ7, DACK5-DACK7	On the AT, DMA channel 0 is no longer used for refresh and is therefore available for other purposes.
MEMR, MEMW	Memory read and write strobes. These signals are active on all memory cycles, while SMEMR and SMEMW are active only on cycles that fall within the address space of the PC for compatibility reasons.
MASTER	A new signal that lets a board become a bus master on the AT bus. A bus handoff using this signal requires several cycles, and the master must relinquish the bus periodically to allow memory refresh (or do the refresh itself).
MEM CS16,	Signals used by a peripheral board to tell the motherboard that it's capable of handling a 16-bit data transfer.

New signal lines added in the AT's 16-bit bus. By duplicating some of the address lines, the AT bus maintains backward compatibitity while providing unlatched address lines for faster cards.

Signal lines added to lSA to create ElSA

Lines	Descrlption
BE0-BE3	Byte enables. These signals indicate which byte lanes of the 32-bit data bus are involved in the current bus cvcle. They're analogous to the BE0 through BE3 signals on the 80386 and 80486 microprocessors.
M-lO	Distinguishes between an ElSA memory cycle and an ElSA l/O cycle.
START	lndicates the start of an ElSA bus cycle.
CMD	Provides timing control within an ElSA bus cycle.
MSBURST	lndicates that a master is capable of performing burst cycles.
SLBURST	Indicates that a slave is capable of accepting burst cycles.
EX32 EX16	lndicate that a slave is an ElSA board and can support a 32- or 16-bit cycle respectively. lf neither of these signals is asserted at the beginning of a cycle the bus falls back to an lSA-compatible mode for that cycle.
EXRDY	lndicates that an ElSA slave is ready to terminate a cycle.
MREQn	Asserted by potential master number n to request the bus.
MAKn	lndicates to master n that it has been granted the bus.
D16-D31	The new data lines that, combined with the data lines on the lSA bus make the ElSA 32-bit data bus.
LA2-LA16, L A17-LA31	New address bus lines. Like L A17-L A23 these lines aren't latched on the motherboard and so provide a fast path to the peripheral boards. Note that there' s no need for an LA1 or LA0; the byte enable lines indicate which of the four byte lanes are used. Also note that there are now 32 address bits, supporting the full address range of the 80386 (rather than the 24-bit address range of the 80286). This lets system RAM grow above 16 megabytes.

Signal Lines added to ISA to create EISA. The addition of the new lines, including 32-bit data and address iines, almost doubles the total number of lines.

AT (80286) Hard disk table parameters

word	maximum number of cylinders
byte	maximum number of heads
word	not used
word	starting write precompensation cylinder
byte	not used
byte	control byte
byte	not used
byte	not used
byte	not used
word	park cylinder / landing zone
byte	no of sectors/track
byte	reserved

The Control Byte

bit Function
0 = 0
1 = 0
2 = 0
3 = 1 only if the drive has more than 8 heads
4 = 0
5 = 0
6 = 1 if retries are to be disabled OR
7 = 1 if retries are to be disabled (either 6 or 7 can be used)

Appendix D: The 8250 register set

Receive data register - at base address + 0 Read

7	6	5	4	3	2	1	0
Data	data	data	data	data	data	data	data

Transmit data register - at base address + 0 Write

7	6	5	4	3	2	1	0
data	data	data	data	data	data	data	data

Interrupt enable register - at base address + 1 Write

7	6	5	4	3	2	1	0
0	0	0	0	enable modem status int.	enable Line status int	enable transmit register empty int.	enable receive data ready int.

Interupt identifier register - at base address + 2 Read

7	6	5	4	3	2	1	0
0	0	0	0	x	y	z	Interrupt pending

x	y	z	
0	0	0	modem status changed causing interrupt
0	0	1	transmit data register empty causing interrupt
0	1	0	receive data register full causing interupt

Line control register - at base address + 3

7	6	5	4	3	2	1	0
DLAB	Set break	Stick parity	Even parity	Parity enable	number of stop bits	wa	wb

wa	wb	
0	0	5 bit data
0	1	6 bit data
1	0	7 bit data
1	1	8 bit data

Modem control register - at base address + 4 Write

7	6	5	4	3	2	1	0
0	0	0	internal Loopback	Set Out2 enables Interrupts	Out1	Set RTS	Set DTR

Line status register - at base address + 5 Read

7	6	5	4	3	2	1	0
	Tx shift register empty	Tx holding register empty	Break received	Framing error	Parity error	Overflow error	Receive data ready

Modem status register - at base address + 6Read

7	6	5	4	3	2	1	0
Carrier detect	Ring indicator	Data set ready	Clear to send	Carrier detect changed	Ring indicator changed	Data set ready changed	Clear to send changed

Appendix E: PC AND XT Bus signal lines

Lines	Description
A0-A19	Twenty system address lines all are used to address memory. During I/O cycles, only the lowest 16 I/0 lines are actually used.
D0-D7	Eight bidirectional data lines.
ALE	Address latch enable. This signal goes high to indicate that a valid address is present on A0-A19 during a memory access.
IRQ2-IRQ7	Six maskable interrupt request lines.
DRQI -DRQ2	DMA request and acknowledge lines. There's no DRQ0 on the bus
DACK1-DACK3	DMA channel 0 is used for DRAM refresh on the PC and the XT.
IO CHRDY	A signal used by a memory or peripheral board to generate wait states.
IORD, IOWR, SMEMR, SMEMW	I/O and memory read and write strobes.
OSCA	14.31818-MHz clock used by some video boards. It's not synchronised with respect to the rest of the bus.
CLK	The bus clock signal (4.77 MHz in the original PC and proportionately faster on later machines). This clock is synchronised with respect to the read and write strobes.
AEN ,TC	Address enable and terminal count. These control signals are used during DMA cycles.
IO CHK	Alerts the processor to parity and other errors via a nonmaskable interrupt.
RESET DRV	Indicates that the system is being reset.
+5VDC, -5VDC, +12VDC, -12VDC, GND	Power-supply rails.

Because of its simple design, each interrupt line can be used by only one adapter card.

Additional Signal Lines for the 16-bit ISA bus

Lines	Descrlptlon
D8-D15	The eight new data lines.
SBHE	System bus high enable, which indicates when these data lines are being used.
IRQ10-12,	More interrupt lines. IRQ13 is absent because that interrupt is reserved for IRQ14-15 the maths coprocessor.
DRQ0, DACK0,	More DMA control lines for new DMA channels.
DRQ5-DRQ7, DACK5-DACK7	On the AT, DMA channel 0 is no longer used for refresh and is therefore available for other purposes.
MEMR, MEMW	Memory read and write strobes. These signals are active on all memory cycles, while SMEMR and SMEMW are active only on cycles that fall within the address space of the PC for compatibility reasons.
MASTER	A new signal that lets a board become a bus master on the AT bus. A bus handoff using this signal requires several cycles, and the master must relinquish the bus periodically to allow memory refresh (or do the refresh itself).
MEM CS16,	Signals used by a peripheral board to tell the motherboard that it's capable of handling a 16-bit data transfer.

New signal lines added in the AT's 16-bit bus. By duplicating some of the address lines, the AT bus maintains backward compatibitity while providing unlatched address lines for faster cards.

Signal lines added to lSA to create ElSA

Lines	Descrlption
BE0-BE3	Byte enables. These signals indicate which byte lanes of the 32-bit data bus are involved in the current bus cvcle. They're analogous to the BE0 through BE3 signals on the 80386 and 80486 microprocessors.
M-lO	Distinguishes between an ElSA memory cycle and an ElSA l/O cycle.
START	lndicates the start of an ElSA bus cycle.
CMD	Provides timing control within an ElSA bus cycle.
MSBURST	lndicates that a master is capable of performing burst cycles.
SLBURST	lndicates that a slave is capable of accepting burst cycles.
EX32 EX16	lndicate that a slave is an ElSA board and can support a 32- or 16-bit cycle respectively. lf neither of these signals is asserted at the beginning of a cycle the bus falls back to an lSA-compatible mode for that cycle.
EXRDY	lndicates that an ElSA slave is ready to terminate a cycle.
MREQn	Asserted by potential master number n to request the bus.
MAKn	lndicates to master n that it has been granted the bus.
D16-D31	The new data lines that, combined with the data lines on the lSA bus make the ElSA 32-bit data bus.
LA2-LA16, L A17-LA31	New address bus lines. Like L A17-L A23 these lines aren't latched on the motherboard and so provide a fast path to the peripheral boards. Note that there' s no need for an LA1 or LA0; the byte enable lines indicate which of the four byte lanes are used. Also note that there are now 32 address bits, supporting the full address range of the 80386 (rather than the 24-bit address range of the 80286). This lets system RAM grow above 16 megabytes.

Signal Lines added to ISA to create EISA. The addition of the new lines, including 32-bit data and address iines, almost doubles the total number of lines.

Check the voltage selector is correctly set before switch on. Also ensure that the earth wire (green/yellow stripe) is screw connected to the chassis of the case.

BIBLIOGRAPHY

1 Peter Norton's Inside the IBM PC published by PRENTICE HALL
ISBN 0-89303-583-1 (Paperback)

2 Ray Duncan Advanced MS-DOS Microsoft Press
ISBN 0-914845-77-2

3 Technical Reference
Personal Computer XT IBM

4 Technical Reference
Personal Computer AT IBM

Keep yourself up to date with magazines such as :-

PC MAGAZINE (local and USA versions) - read the user to user and Utility articles.

Personal Computer World (UK) - A lot of good articles about the Internet.

Pc Pro (UK) - Excellent technical magazine with articles on Windows, Windows NT and Unix.