Computer Organization and Assembly Language

Lecture 2 – x86 Processor Architecture

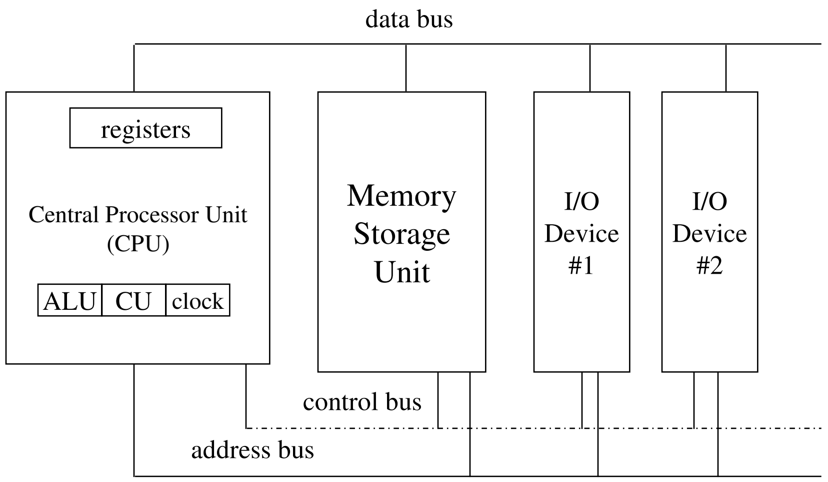
What is a processor?

- CPU (***C***entral ***P***rocessing ***U***nit) or Processor - is the brain of the computer.
- In the PC, the Processor is in the Intel 80x86 or Pentium family.

What does the processor contain?

- ***Busses*** - Carries data, control signals and addresses between the processor components and other devices within the computer.
- ***Registers*** - High-speed memory units within the CPU.
- ***Clock*** - synchronizes all the steps in fetching, decoding and executing instructions.

Basic Microprocessor Design



Steps of the Instruction Execution Cycle

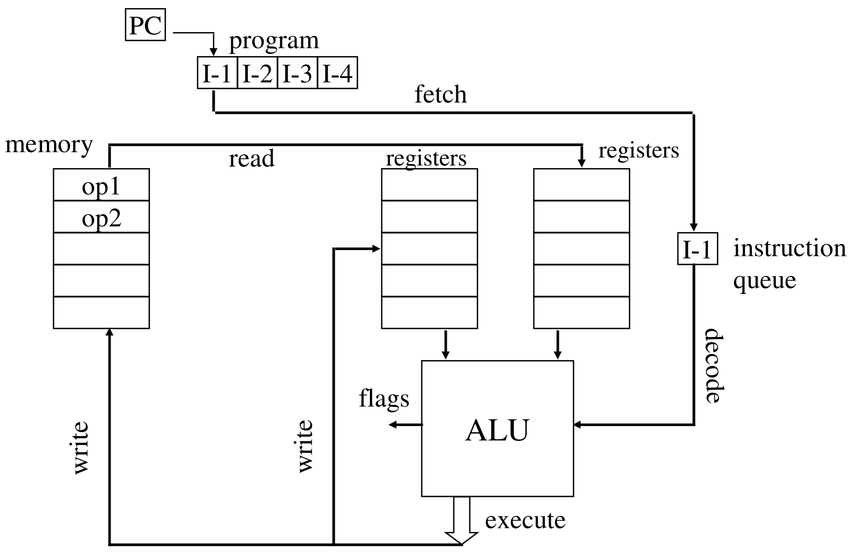
As many as five different operations may be required to execute machine-language instructions:

- **Fetch** – The control unit fetches the instruction, copying it from memory into the CPU, incrementing the program counter (PC).
- **Decode** – The control determines the type of instruction, passing along the necessary control signals to the ALU indicating the operations to be performed.
- **Fetch operands** – If it is a memory-reference instruction, operand(s) have to be fetched from memory
- **Execute** – the operation is performed
- **Store output operand** – the result is saved in memory

Pseudocode For the Instruction Execution Cycle

```
loop
   fetch next instruction
   advance the instruction pointer (ip)
   decode the instruction
   if memory operand is needed, read value
     from memory
   execute the instruction
   if result is memory operand, write
     result to memory
continue loop
```

Instruction Execution Cycle



Intel 32-bit Architecture

The 80386 was Intel’s first 32-bit processor and the first to include parallel stages of execution. The parts that carry these out are:

BIU (***B***us ***I***nterface ***U***nit) - accesses memory and provides input-output.

- **Code Prefetch Unit** - receive machine instructions from BIU & inserts them into the prefetch queue.
- **Instruction Decode Unit** - decodes machine instructions and converts them into microcode instructions.
- **Execution Unit** - executions microcode instructions.
- **Segment Unit** - translates logical addresses into linear addresses (and performs protection checks).
- **Paging Unit** - translates linear addresses into physical addresses, performs page protections checks and keeps track of recently accessed pages.

Six-Stage Non-Pipeline Execution

Stages

Cycles	S-1	S-2	S-3	S-4	S-5	S-6
1	I-1					
2		I-1				
3			I-1			
4				I-1		
5					I-1	
6						I-1
7	I-2					
8		I-2				
9			I-2			
10				I-2		
11					I-2	
12						I-2

Without pipelining, CPU resources are wasted

For **k** *stages,* **n** *instructions require* **n × k** *clock cycles*

Six-Stage Pipeline Instruction

Stages

Cycles	S-1	S-2	S-3	S-4	S-5	S-6
1	I-1					
2	I-2	I-1				
3		I-2	I-1			
4			I-2	I-1		
5				I-2	I-1	
6					I-2	I-1
7						I-2

With pipelining, a second instruction can begin execution almost immediately and finish sooner

n *instructions require* **k + (n-1)** *clock cycles*

Pipeline Execution Using A Single Pipeline

Stages

exe

Cycles

	S-1	S-2	S-3	S-4	S-5	S-6
1	I-1					
2	I-2	I-1				
3	I-3	I-2	I-1			
4		I-3	I-2	I-1		
5			I-3	I-1		
6				I-2	I-1	
7				I-2	▨	I-1
8				I-3	I-2	
9				I-3	▨	I-2
10					I-3	
11						I-3

If an instruction needs 2 clock cycles to perform an instruction, the one pipeline leads to wasted clock cycles

n *instructions require* **k + 2n - 1** *clock cycles*

Superscalar 6-Stage Pipelined Processor

Stages

Cycles

				S4			
	S-1	S-2	S-3	u	v	S5	S-6
1	I-1						
2	I-2	I-1					
3	I-3	I-2	I-1				
4	I-4	I-3	I-2	I-1			
5		I-4	I-3	I-1	I-2		
6			I-4	I-3	I-2	I-1	
7				I-3	I-4	I-2	I-1
8					I-4	I-3	I-2
9						I-4	I-3
10							I-4

If there are 2 pipelines,odd instructions use the odd pipeline and even instructions use the even pipeline

n *instructions require* **k + n** *clock cycles*

Reading From Memory

- Memory access is an important factor in understanding program execution speed because memory access via the system bus is much slower than the CPU clock.
- The clock cycles that are wasted while waiting for operands to be fetched are called ***wait states***.

Cache Memory

- Cache memory saves data received fetched from or written to memory. Since it is much faster than conventional memory, there are fewer wait states.
- Level-1 cache is built into the processor.
- Level-2 cache is located on separate chips near the processor.

Load and Execute Process

When you “tell” the computer to run a program, certain things happen:

- The user issues a command to run the program
- The operating system (OS) finds the program’s filename in the system directory, if necessary searching through the path for the name.
- The OS retrieves the basic file information, including size and disk location.
- The OS determines a memory location for the file and reads it in and creates a process table entry for it.

Load and Execute Process (continued)

- The OS executes a branching instruction, beginning program execution, creating a new process (the user’s program).
- The process runs by itself, with the OS keeping track of its use of system resources.
- When the program is finished, its table entry and memory are made available for reuse.

Multitasking

- An operating system that can run more than one process (or task) at once is called ***multitasking***.
- Since most computers have only one processor, they multitask by giving each process a small portion of processor time called a ***time slice***.
- The computer must be able to switch processes quickly, which means that they can store the process's state before switch.
- Round-robin scheduling is a typical scheduling algorithm where there is a strict rotation between the active processes.

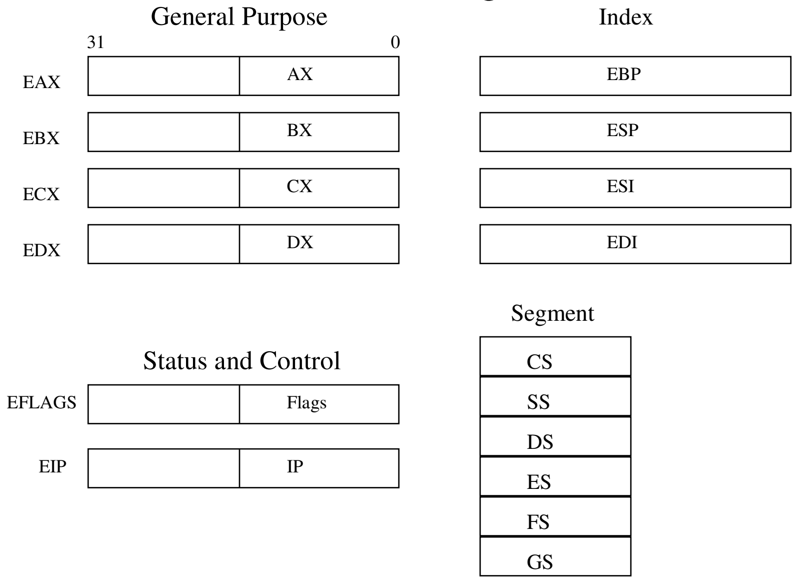
IA-32 Processor Modes of Operations

- There are three basic modes of operation on IA-32 processors:
  - ***Protected Mode*** – The native processor state, where all instructions and features are available. Each process is given its own memory segment and the processor catches any process attempting to go outside its own segment
  - ***Real-address Mode*** – The processor acts as if it were an Intel 8086 processor with its more limited environment
  - ***System Management Mode*** – provides a mechanism for implementation power management and system security

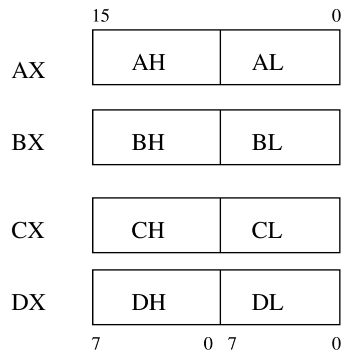
IA-32 Processor Address Space

- In protected mode IA-32 processors can access up to 4 Gigabytes of storage, with memory addresses from 0 to 2^{32}-1.
- In real mode, a maximum of 1 megabyte of memory can be accessed with memory addresses from 0 to 2^{10}-1.
- The IA-32 processors provide a Virtual 8086 where multiple MS-DOS programs can run safely within an Windows environment.

32-bit Register



16-bit Processor Architecture General Purpose Registers



AX (Accumulator) - favored for arithmetic opertions

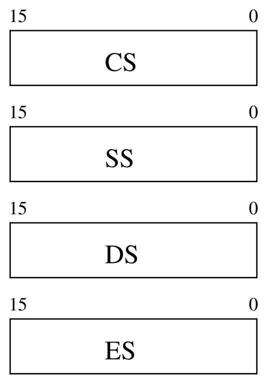
BX (Base) - Holds base address for procedures and variables

CX (Counter) - Used as a counter for looping operations

DX (Data) - Used in mulitplication and division operations.

Segment Registers

Segment registers are used to hold base addresses for program code, data and the stack.



CS (Code Segment) - holds the base address for all executable instructions in the program

SS (Stack Segment) - holds the base address for the stack

DS (Data Segment) - holds the base address for variables

ES (Extra Segment) - an additional base address value for variable.

Index Registers

Index Registers contain the offsets for data and instructions.

Offset - distance (in bytes) from the base address of the segment.

BP	BP (Base Pointer) - contains an assumed offset from the SS register; used to locate variables passed between procedures.
SP	SP (Stack Pointer) - contains the offset for the top of the stack.
SI	SI (Source Index) - Points to the source string in string move instructions.
DI	DI (Destination Index) - Points to the source destination in string move instructions.

Status and Control Registers

IP	IP (Instruction Pointer) - contains the offset of the next instruction to be executed within the current code segment.

x	x	x	x	O	D	I	T	S	Z	x	A	x	P	x	C

Flags register contain individual bits which indicate CPU status or arithmetic results. They are usually set by specific instructions.

O = Overflow	S = Sign
D = Direction	Z = Zero
I = Interrupt	A = Auxiliary Carry
T = Trap	P = Parity
x = undefined	C = Carry

Flags

There are two types of flags: control flags (which determine how instructions are carried out) and status flags (which report on the results of operations.

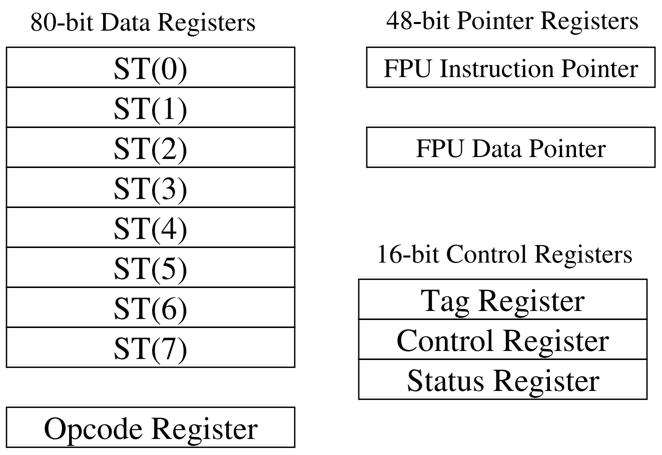
Control flags include:

– ***Direction*** Flag (DF) - affects the direction of block data transfers (like long character string). 1 = up; 0 - down.
– ***Interrupt*** Flag (IF) - determines whether interrupts can occur (whether hardware devices like the keyboard, disk drives, and system clock can get the CPU's attention to get their needs attended to.
– ***Trap*** Flag (TF) - determines whether the CPU is halted after every instruction. Used for debugging purposes.

Status Flags

- Status Flags include:
  – ***Carry*** Flag (CF) - set when the result of **unsigned** arithmetic is too large to fit in the destination. 1 = carry; 0 = no carry.
  – ***Overflow*** Flag (OF) - set when the result of **signed** arithmetic is too large to fit in the destination. 1 = overflow; 0 = no overflow.
  – ***Sign*** Flag (SF) - set when an arithmetic or logical operation generates a negative result. 1 = negative; 0 = positive.
  – ***Zero*** Flag (ZF) - set when an arithmetic or logical operation generates a result of zero. Used primarily in jump and loop operations. 1 =zero; 0 = not zero.
  – ***Auxiliary Carry*** Flag - set when an operation causes a carry from bit 3 to 4 or borrow (frombit 4 to 3). 1 = carry, 0 = no carry.
  – ***Parity*** - used to verify memory integrity. Even # of 1s = Even parity; Odd # of 1s = Odd Parity

Floating-Point Unit



The Intel Microprocessor Family

- The Intel family owes its origins to the ***8080***, an 8-bit processor which could only access 64 kilobytes of memory.
- The ***8086*** (1978) had 16-bit registers, a 16-bit data bus, 20-bit memory using segmented memory. The IBM PC used the ***8088***, which was identical except it used an 8-bit data bus.
- ***8087*** - a math co-processor that worked together with the 8086/8088. Without it, floating point arithmetic require complex software routines.
- ***80286*** - ran in real mode (like the 8086/8088) or in protected mode could access up tp 16MB using 24-bit addressing with a clock spped between 12 and 25 MHz. Its math co-processor was the 80287.

The Intel Microprocessor Family (continued)

- 80386 or ***i386*** (1985) - used 32-bit registers and a 32-bit data bus. It could operate in real, protected or virtual mode. In virtual mode, multiple real-mode programs could be run.
- ***i486*** - The instruction set was implemented with up to 5 instructions fetched and decoded at once. SX version had its FPU disabled.
- The Pentium processor had an original clock speed of 90 MHz and cold decode and executed two instructions at the same time, using ***dual pipelining***.

P6 Processor Family

- The P6 family of processors was introduced in 1995.
- It includes the Pentium Pro, Pentium II, Pentium III and and Pentium 4.
- The Pentium II introduces MMX technology for multimedia applications.
- The Pentium III introduced SIMD with 128-bit registers to move larger amounts of data.
- The Pentium 4 uses NetBurst micro-architecture to allow the processors to operate at higher speeds.

Intel Core Processor Family

- Intel introduce the Core family of processors in 2006, which are more powerful than the Pentium processors that preceded them.
- So far they include:
  - Core 2 Duo – 2 processors codes, 1.8-3.3 GHz, 64 bit, 6 MByte L2 cache.
  - Core 2 Quad - 4 processors codes, up to 12 MByte L2 cache, 1333 MHz front side bus.

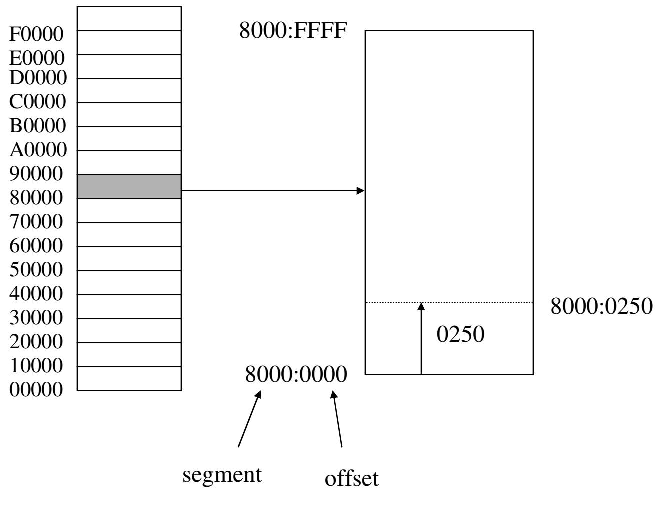
CISC Architecture

- The Intel processors have been based on the CISC (***C***omplex ***I***nstruction ***S***et ***C***omputer) approach to processor design.
- CISC processors have large , powerful instruction sets that can include many high-level operations. But the size of the instruction set makes the control unit relatively slow.

RISC Architecture

- RISC computers use smaller, streamlined instruction sets that allow their control units to be quicker.
- Intel processors are backwards-compatible and are basically CISC but use RISC features such as pipelining and superscalar.

Segmented Memory Map, Real-Address Mode



Calculating Absolute Addresses

- Every byte of PC memory has its own address, with addresses running from 0 up through highest memory location.
- Logical addresses (used in instructions) and physical addresses (where data and instructions are stored) are not the same; there is a translation process.
- There are two hexadecimal formats used used by Intel processors:
  - 32-bit segment-offset address (e.g.,08F1:0100)
  - 20-bit absolute address (e.g., 09010)

```
Segment value:          0 8 F 1 (0)
Offset:                   0 1 0 0
Absolute address        0 9 0 1 0
```

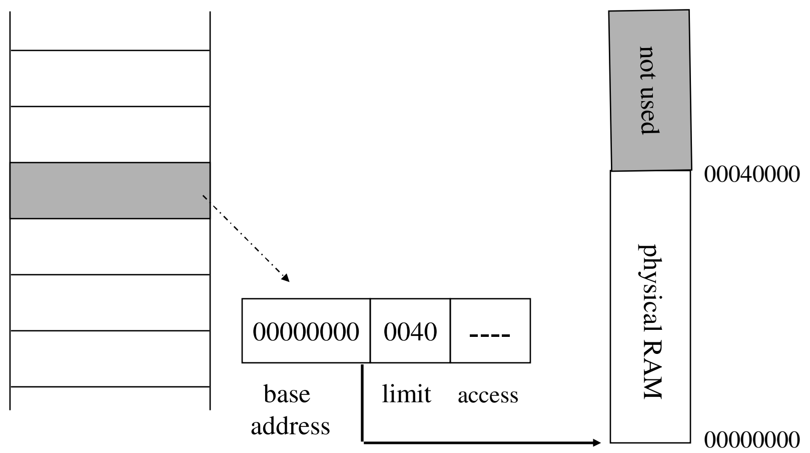
Relocatable addressing

- This is an example of relocatable addressing, which allows programs on mulitasking systems to be moved from one area of memory to another without rearranging every address referenced.
- Addresses can be rearranged simply by changing the value of the appropriate segment register.

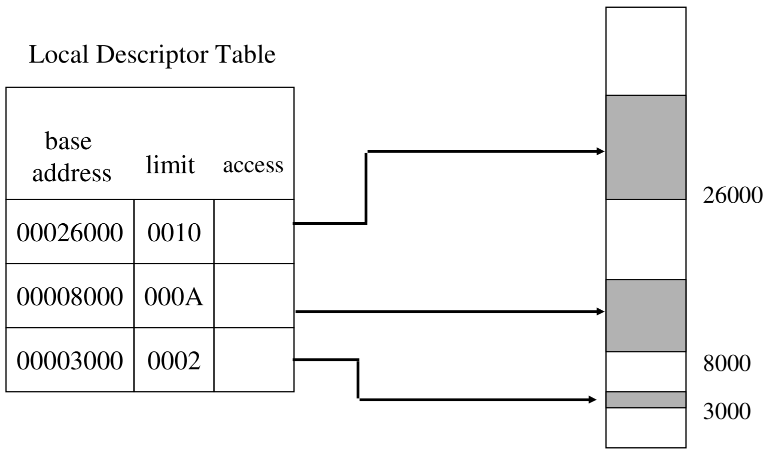
Protected Mode Memory Management

- When the processor runs in protected mode, a program can access up to 4 gigabytes of memory.
- Although the programmer's view of memory is a flat image of 4 GB, the operating system works in the background to create and maintain this image.
- The segment registers point to segment descriptor tables, which define locations of the program segments:
  - CS refers to the code segment's descriptor table
  - DS refers to the data segment's descriptor table
  - SS refers to the stack segment's descriptor table

Flat Segmentation Memory Model



Multi-Segment Memory Model



Local Descriptor Table

base address	limit	access
00026000	0010	
00008000	000A	
00003000	0002	

Paging

- IA-32 architecture also allows memory segments to be divided into 4K units called ***pages***.
- Many of these pages of memory are saved on disk in a swap file and are loaded into memory (and rewritten in the swap file) when the CPU needs a page that is not present in physical memory. This situation is called a ***page fault***.
- The use of paging and swap files allows the memory used to be several times larger than physical memory; it is known as ***virtual memory***.

The Motherboard

- The motherboard has connections to all or most of the following:
  - CPU
  - External cache memory
  - Main memory SIMMs or DIMMs
  - ROM BIOS
  - IDE cables (for hard disks and CD-ROM drives)
  - Sound synthesizers
  - Parallel, serial, USB, video, keyboard, joystick, and mouse connections
  - Network adapters
  - PCI Bus Connectors for sound cards, graphics cards, data acquisition boards and other I/O devices.

Other PC Components

- The PCI (Peripheral Component Interconnect) bus was develop by Intel to connect the Pentium processor with other devices within the computer.
- The Motherboard chipset includes other controllers and processors that work as a set to manage all the components of the computer, including the DMA (Direct Memory Access) Controller, Interrupt Controller, Timer Counter, Keyboard and Mouse Controller, etc.

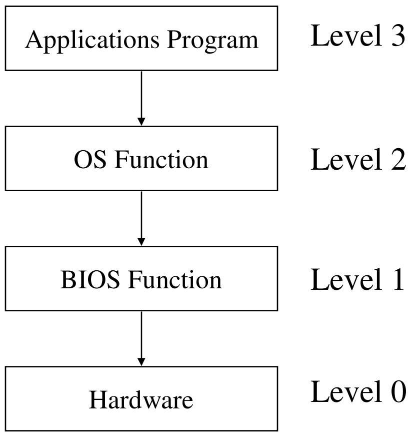
Video Output

- The video adapter control the display of both text and graphics.
- The video adapter consists of:
  - the video controller, which is a special-purpose microprocessor which controls what appears where on the screen.
  - video display memory, which stores what is displayed where on the screen.
- All text and graphics is stored in video RAM and sent to the monitor via the video controller.

Memory

- There are several types of memory that is used by PCs:
  - ROM (***R***ead-***O***nly ***M***emory) - memory permanently burnt into chips and cannot be changed.
  - EPROM (***E***rasable ***P***rogrammable ***R***ead-***O***nly ***M***emory) – can be erased using UV light and reprogrammed.
  - DRAM (***D***ynamic ***R***andom ***A***ccess ***M***emory) –where most programs and data are stored. IF it is not refreshed regularly, its contents is lost.
  - SRAM (***S***tatic ***R***andom ***A***ccess ***M***emory) – it doesn't need refreshing, making it ideal for cache's high-speed use.
  - CMOS RAM – used on the motherboard to store system setup data.

Access Levels for I/O Operations


Applications Program
Level 3
OS Function
Level 2
BIOS Function
Level 1
Hardware
Level 0