Principles and Applications of Microcomputers

1. Introduction to Microprocessor and Computer

國立臺北科技大學 電子工程系
焦名楷

TAIPEI TECH

Introduction

- Overview of Intel microprocessors.
- Discussion of history of computers.
- Function of the microprocessor.
- Terms and jargon (**computerese).**

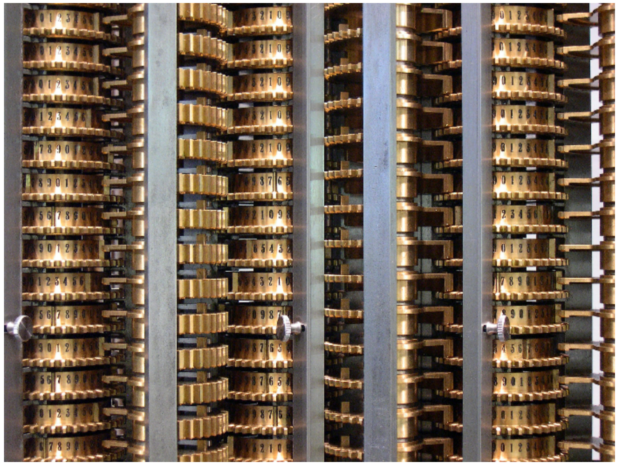
1–1 A HISTORICAL BACKGROUND

- Events leading to development of the microprocessor.
- 80X86, Pentium, Pentium Pro, Pentium III, Pentium 4, and Core2 microprocessors.
- While not essential to understand the microprocessor, furnishes:
  - interesting reading
  - historical perspective of fast-paced evolution

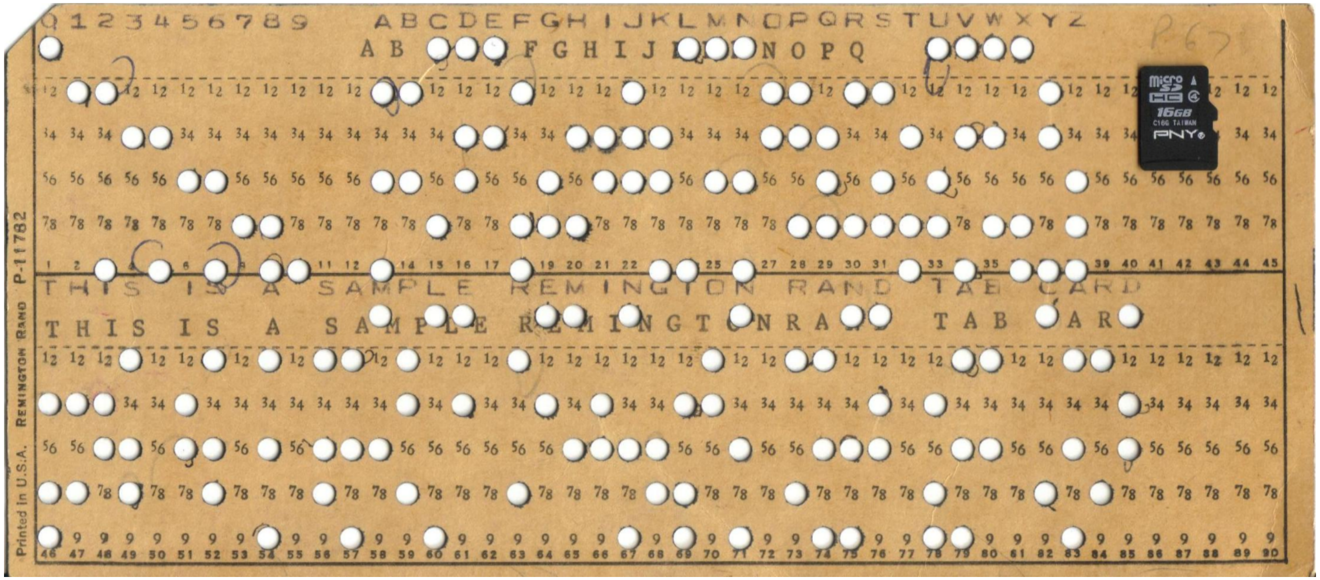
The Mechanical Age

- Idea of computing system not new.
- Calculating with a machine dates to 500 BC (Before Christ).
- Babylonians invented the **abacus**.
  - first mechanical calculator
  - strings of beads perform calculations
- Used by ancient priests to keep track of storehouses of grain.
  - still in use today

- When moved one complete revolution, a second gear advances one place.
- In 1642 mathematician Blaise Pascal invented a calculator constructed of gears and wheels.
  - PASCAL programming language is named in honor of Blaise Pascal.
- Humans dreamed of mechanical machines that could compute with a program
- Commissioned in 1823 by Royal Astronomical Society to build programmable calculating machine.



- Variable program could modify function of the machine to perform various calculating tasks.
- Input through punched cards, much as computers in the 1950s and 1960s used punched cards
- *Jacquard's loom* used punched cards to select intricate weaving patterns in cloth it produced.
  - punched cards programmed the loom



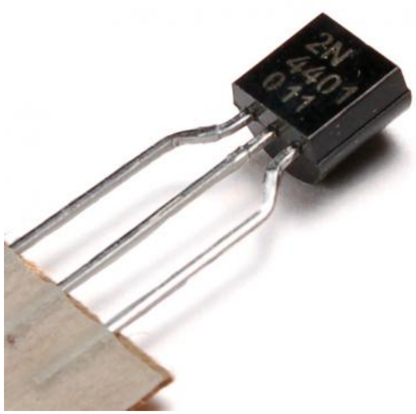
The Electrical Age

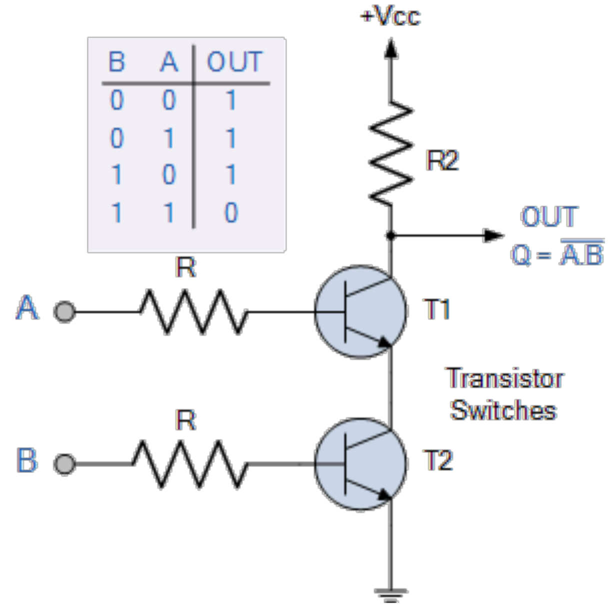
- 1800s saw advent of the electric motor.
- Also a multitude of electrically motor-driven adding machines based on the Pascal mechanical calculator.
  - common office equipment until 1970s
- Introduced by Bomar Corporation the Bomar Brain, was a handheld electronic calculator.
  - first appeared in early 1970s

Bowmar *901B*

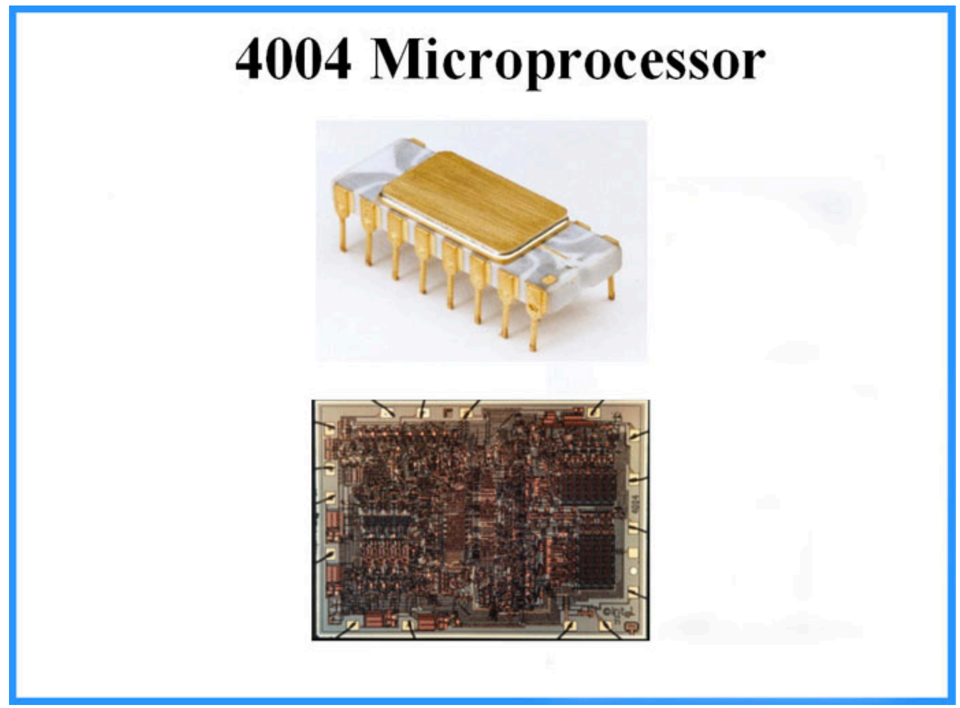
- First electronic computer placed in operation to break secret German military codes.
- **Electronic Numerical Integrator and Calculator (ENIAC)**, a huge machine.
  - over 17,000 vacuum tubes; 500 miles of wires
  - weighed over 30 tons
  - about 100,000 operations per second
- Programmed by rewiring its circuits.
  - workers changed electrical connections on plug-boards like early telephone switchboards
- Required frequent maintenance.
  - vacuum tube service life a problem

- December 23, 1947, John Bardeen, William Shockley, and Walter Brattain develop the transistor at Bell Labs.
- Followed by 1958 invention of the integrated circuit (IC) by Jack Kilby of Texas Instruments.
- IC led to development of digital integrated circuits in the 1960s.
  – RTL, or resistor-to-transistor logic
- First microprocessor developed at Intel Corporation in 1971.





- Intel engineers Federico Faggin, Ted Hoff, and Stan Mazor developed the 4004 microprocessor.
- U.S. Patent 3,821,715.
- Device started the microprocessor revolution continued today at an ever-accelerating pace.



Programming Advancements

- Once programmable machines developed, programs and programming languages began to appear.
- As early practice of rewiring circuits (X) proved too cumbersome, computer languages began to appear in order to control the computer.
- The first, **machine language**, was constructed of ones and zeros using binary codes.
  - stored in the computer memory system as groups of instructions called a program

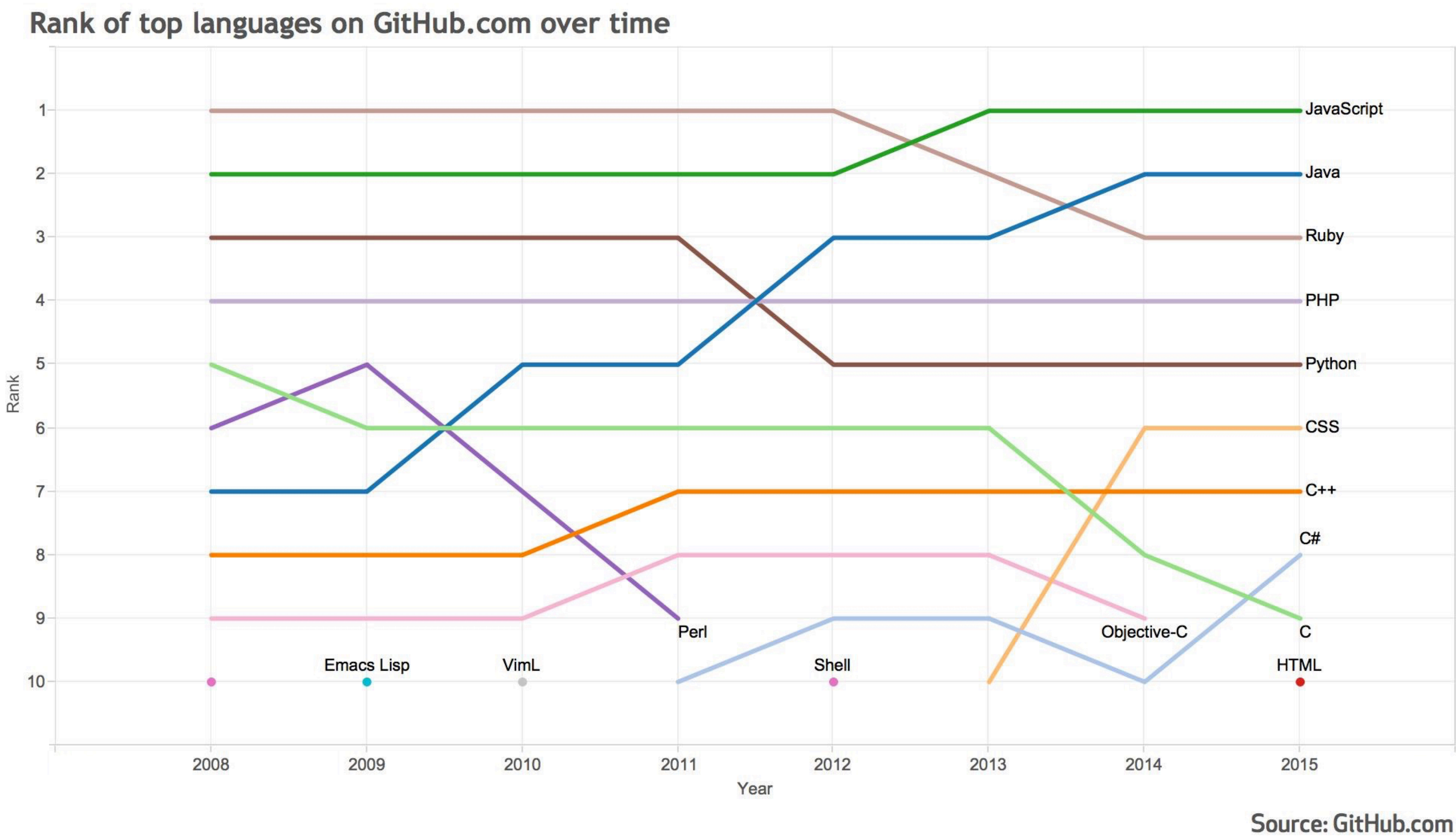
- More efficient than rewiring a machine to program it.
  - still time-consuming to develop a program due to sheer number of program codes required (i.e., machine code)
- Mathematician John von Neumann first modern person to develop a system to accept instructions and store them in memory.
  - Computers are often called **von Neumann machines** in his honor.

- Once systems such as UNIVAC became available in early 1950s, **assembly language** was used to simplify entering binary code.
- Assembler allows programmer to use mnemonic (助記) codes…
  – such as ADD for addition
- In place of a binary number.
  – such as 0100 0111
- Assembly language an aid to programming.

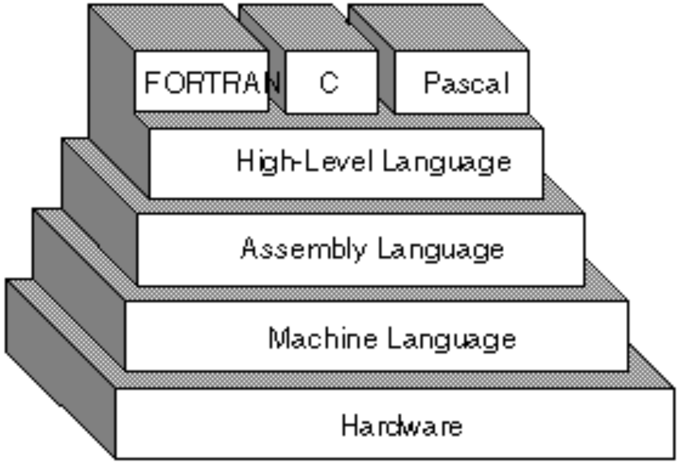
- 1957 Grace Hopper developed first high-level programming language called **FLOWMATIC.**
  - computers became easier to program
- In same year, IBM developed FORTRAN **FORmula TRANslator**) for its systems.
  - Allowed programmers to develop programs that used formulas to solve mathematical problems.
- FORTRAN is still used by some scientists for computer programming.
  - Similar language, ALGOL (**ALGOrithmic Language**) introduced about a year later

- First successful, widespread programming language for business applications was COBOL (**COmputer Business Oriented Language**).
- COBOL usage diminished in recent years.
  - still a player in some large business and government systems
- Another once-popular business language is RPG (**Report Program Generator**).
  - allows programming by specifying form of the input, output, and calculations

- Some common modern programming languages
  - Such as C/C++, C#, Java, JavaScript, Ruby, Python, Swift, Obj-C, and so on.



- These languages allow programmer almost complete control over the programming environment and computer system.
  – especially C/C++
- C/C++ replacing some low-level machine control software or drivers normally reserved for assembly language.
- Assembly language still plays important role.
  – many video games written almost exclusively in assembly language
- Assembly also interspersed with C/C++ to perform machine control functions efficiently.



The Microprocessor Age

- World's first microprocessor the Intel 4004.
- A 4-bit microprocessor-programmable controller on a chip.
- Addressed 4096, 4-bit-wide memory locations.
  - a **bit** is a binary digit with a value of one or zero
  - 4-bit-wide memory location often called a **nibble (btw, 8-bit: a byte)**
- The 4004 instruction set contained 45 instructions.

[Those instructions preceded by an asterisk (*) are 2 word instructions that occupy 2 successive locations in ROM]

MACHINE INSTRUCTIONS (Logic 1 = Low Voltage = Negative Voltage; Logic 0 = High Voltage = Ground)

MNEMONIC	DESCRIPTION OF OPERATION	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$
NOP	No operation.	0 0 0 0	0 0 0 0
*JCN	Jump to ROM address $A_2 A_2 A_2 A_2$, $A_1 A_1 A_1 A_1$ (within the same ROM that contains this JCN instruction) if condition $C_1 C_2 C_3 C_4$ (1) is true, otherwise skip (go to the next instruction in sequence).	0 0 0 1 $A_2 A_2 A_2 A_2$	$C_1 C_2 C_3 C_4$ $A_1 A_1 A_1 A_1$
*FIM	Fetch immediate (direct) from ROM Data D_2, D_1 to index register pair location RRR. (2)	0 0 1 0 $D_2 D_2 D_2 D_2$	R R R 0 $D_1 D_1 D_1 D_1$
SRC	Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X_2 and X_3 time in the Instruction Cycle.	0 0 1 0	R R R 1
FIN	Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.	0 0 1 1	R R R 0
JIN	Jump indirect. Send contents of register pair RRR out as an address at A_1 and A_2 time in the Instruction Cycle.	0 0 1 1	R R R 1
*JUN	Jump unconditional to ROM address A_3, A_2, A_1.	0 1 0 0 $A_2 A_2 A_2 A_2$	$A_3 A_3 A_3 A_3$ $A_1 A_1 A_1 A_1$
*JMS	Jump to subroutine ROM address A_3, A_2, A_1, save old address. (Up 1 level in stack.)	0 1 0 1 $A_2 A_2 A_2 A_2$	$A_3 A_3 A_3 A_3$ $A_1 A_1 A_1 A_1$
INC	Increment contents of register RRRR. (3)	0 1 1 0	R R R R
*ISZ	Increment contents of register RRRR. Go to ROM address A_2, A_1 (within the same ROM that contains this ISZ instruction) if result $\neq 0$, otherwise skip (go to the next instruction in sequence).	0 1 1 1 $A_2 A_2 A_2 A_2$	R R R R $A_1 A_1 A_1 A_1$
ADD	Add contents of register RRRR to accumulator with carry.	1 0 0 0	R R R R
SUB	Subtract contents of register RRRR to accumulator with borrow.	1 0 0 1	R R R R
LD	Load contents of register RRRR to accumulator.	1 0 1 0	R R R R
XCH	Exchange contents of index register RRRR and accumulator.	1 0 1 1	R R R R
BBL	Branch back (down 1 level in stack) and load data DDDD to accumulator.	1 1 0 0	D D D D
LDM	Load data DDDD to accumulator.	1 1 0 1	D D D D

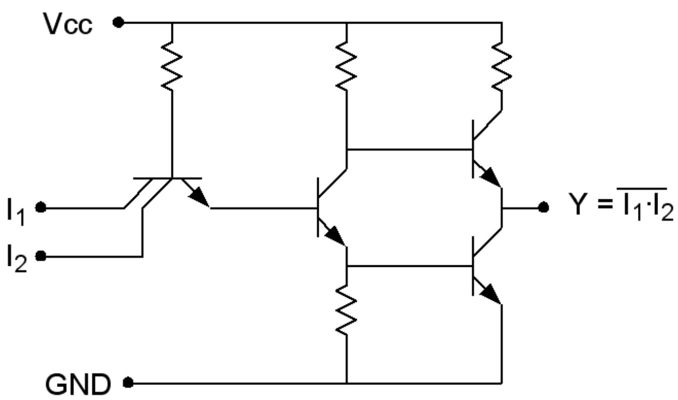
Barry B. Brey

- With the microprocessor a commercially viable product, Intel released 8008 in 1971.
  - extended 8-bit version of 4004 microprocessor
- Addressed expanded memory of 16K bytes.
  - A **byte** is generally an 8-bit-wide binary number and a **K** is 1024.
  - memory size often specified in K bytes
- Contained additional instructions, 48 total.
- Provided opportunity for application in more advanced systems.
  - engineers developed demanding uses for 8008

Prefix	Analog value	Digital value
p (pico)	10^{-12}	-
n (nano)	10^{-9}	-
μ (micro)	10^{-6}	-
m (milli)	10^{-3}	-
k (kilo)	10^{3} (1000)	2^{10} (1024)
M (mega)	10^{6} (1,000,000)	2^{20} (1,048,576)
G (Giga)	10^{9} (1,000,000,000)	2^{30} (1,073,741,824)
T (Tera)	10^{12} (1,000,000,000,000)	2^{40} (1,099,511,627,776)

What Was Special about the 8080?

- Intel introduced 8080 microprocessor in 1973.
  - Second of the 8-bit Intel microprocessors
- 8080 addressed four times more memory.
  - 64K bytes vs 16K bytes for 8008
- Executed additional instructions; 10x faster.
  - Addition (+ opeation) taking 20 μs on an 8008-based system required only 2.0 μs on an 8080-based system
- TTL (transistor-transistor logic) compatible.
  - the 8008 was not directly compatible



The 8085 Microprocessor

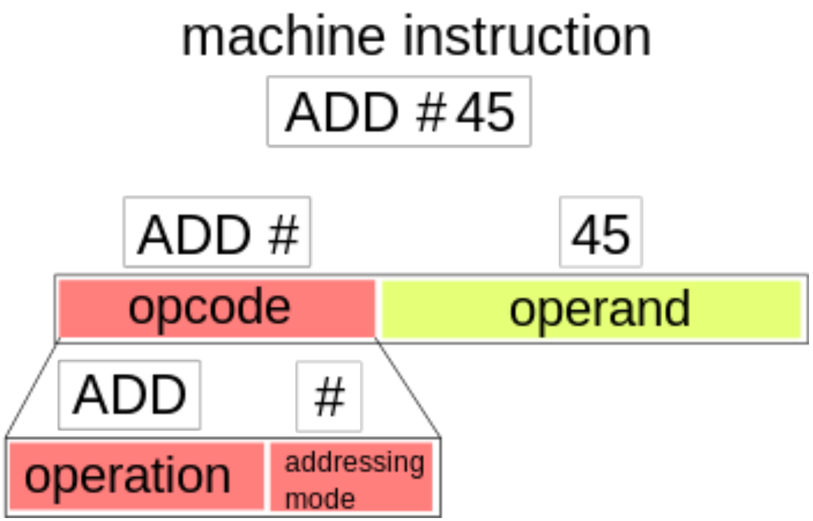
- In 1977 Intel Corporation introduced an updated version of the 8080—the 8085.
- Last 8-bit, general-purpose microprocessor developed by Intel.
- Slightly more advanced than 8080; executed software at an even higher speed.
  – 769,230 instructions per second vs 500,000 per second on the 8080).
- Main advantages of 8085 were its internal clock generator and system controller, and higher clock frequency.
  – higher level of component integration reduced the 8085's cost and increased its usefulness

The Modern Microprocessor

- In 1978 Intel released the 8086; a year or so later, it released the 8088.
- Both devices of 8086 & 8088 are 16-bit microprocessors.
  - executed instructions in as little as 400 ns (2.5 **millions of instructions per second**) faster
  - major improvement over execution speed of 8085
- 8086 & 8088 addressed 1M byte of memory.
  - 16 times more memory than the 8085
  - **1M-byte memory** contains 1024K byte-sized memory locations or 1,048,576 bytes

- Internal architecture 16 bits
- External bus Width 8 bits data, 20 bits address

- Higher speed and larger memory size allowed 8086 & 8088 to replace smaller minicomputers in many applications.
- Another feature was a 4- or 6-byte instruction cache or queue that prefetched instructions (opcode) before they were executed.
  - queue sped operation
  - an opcode (abbr., operation code) is the portion of a machine language instruction that specifies the operation to be performed.
- These microprocessors are called CISC (**complex instruction set computers**).



	CISC	RISC
1	Complex instructions taking multiple cycles	Simple instructions taking 1 cycle
2	Any instruction may reference memory	Only LOADS/STORES reference memory
3	Not pipelined or less pipelined	Highly piplined
4	Instructions interpreted by the microprogram	Instructions executed by the hardware
5	Variable format instructions	Fixed format instructions
6	Many instructions and modes	Few instructions and modes
7	Complexity in the microprogram	Complexity is in the compiler
8	Single register set	Multiple register sets

Example for RISC vs. CISC

Consider the the program fragments:

CISC

```
mov ax, 10
mov bx, 5
mul bx, ax
```

RISC

```
      mov ax, 0
      mov bx, 10
      mov cx, 5
Begin add ax, bx
      loop Begin
```

The 80286 Microprocessor

- Even the 1M-byte memory system proved limiting for databases and other applications (more memory-consuming).
  - Intel introduced the 80286 in 1983
  - an updated 8086
- Almost identical to the 8086/8088 except memory system.
  - addressed 16M-byte memory system instead of a 1M-byte system
- Instruction set almost identical except for a few additional instructions.
- 80286 clock speed increased in 8.0 Mhz version.

The 32-Bit Microprocessor

- Applications demanded faster microprocessor speeds, more memory, and wider data paths.
- Led to the 80386 in 1986 by Intel.
  - major overhaul of 16-bit 8086–80286 architecture (from 16 to 32bits)
- Intel's first practical microprocessor to contain a 32-bit data bus and 32-bit memory address.
- Through 32-bit buses, 80386 addressed up to 4G bytes of memory.
  - **1G** memory = 1024M, or 1,073,741,824 locations
  - 1,000,000 typewritten, double-spaced pages of ASCII text data
- 80386SLC contained an internal cache to process data at even higher rates.

- Intel released 80386EX in 1995.
- Designed for an **embedded system.**
  - contains all components on a single integrated circuit
- On-chip peripherals: Programmable chip selection logic
  - Clock and power management
  - Timers/counters
  - Watchdog timer
  - Serial I/O units (sync and async) and parallel I/O
  - DRAM refresh controller

- Applications needing higher speeds and large memory systems include software systems that use a GUI, or **graphical user interface**
- Modern graphical displays contain large amount of picture elements (pixels).
- VGA (**variable graphics array**) resolution
  - resolution used to display computer boot screen
  - To display one screen of information, each picture element must be changed.
  - requires a high-speed microprocessor
- GUI packages require high microprocessor speeds and accelerated video adapters for quick and efficient manipulation of video text and graphical data.
  - the most striking system is Microsoft Windows

- 32-bit microprocessor needed due to size of its data bus.
  - transfers real (single-precision floating-point) numbers that require 32-bit-wide memory
- To process 32-bit real numbers, the microprocessor must efficiently pass them between itself and memory.
  - with 8-bit data bus, takes four read or write cycles
  - only one read or write cycle is required for 32 bit
- Significantly increases speed of any program that manipulates real numbers.

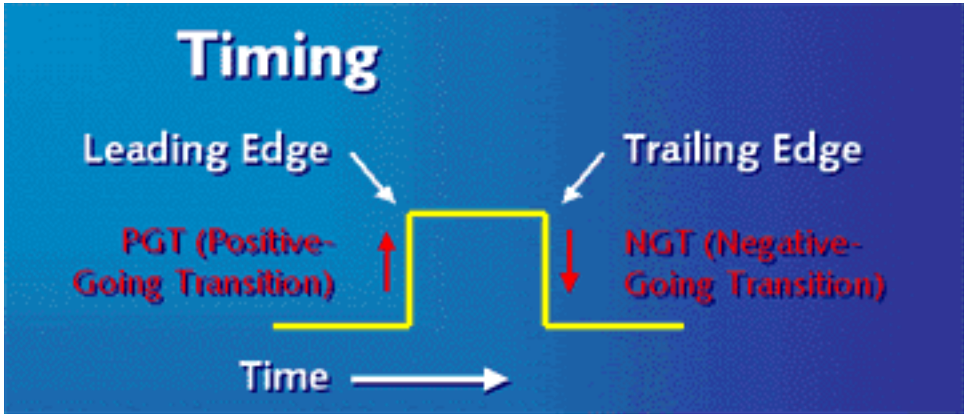
- High-level languages, spreadsheets, and database management systems use real numbers for data storage.
  - also used in graphical design packages that use vectors to plot images on the video screen
  - CAD (**computer-aided drafting/design**) systems as AUTOCAD, ORCAD
- 80386 had higher clocking speeds and included a memory management unit.
  - allowed memory resources to be allocated and managed by the operating system

- 80386 included hardware circuitry for memory management and assignment.
  - improved efficiency, reduced software overhead
  - earlier microprocessors left memory management completely to the software
- Instruction set, memory management upward-compatible with 8086, 8088, and 80286.
  - additional instructions referenced 32-bit registers and managed the memory system
- Features allowed older, 16-bit software to operate on the 80386 microprocessor.

The 80486 Microprocessor

- In 1989 Intel released the 80486.
- Highly integrated package.
  - containing well over 1.2 million transistors.
- Located within the 80486 circuit
  - memory-management unit (MMU)
  - 80387-like numeric coprocessor
  - 8K-byte cache memory system (L1 cache)
  - Full and 80386-like 32-bit microprocessor

- Internal structure of 80486 modified so about half of its instructions executed in one clock instead of two clocks.
  - in a 50 MHz version, about half of instructions executed in 25 ns (50 MIPs)
  - 50% over 80386 operated at same clock speed
  - clock doubling
- Double-clocked 80486DX2 executed instructions at 66 MHz, with memory transfers at 33 MHz.
  - called a double-clocked microprocessor



- A triple-clocked version improved speed to 100 MHz with memory transfers at 33 MHz.
  - about the same speed as 60 MHz Pentium.
- Expanded 16K-byte cache.
  - in place of standard 8K-byte cache

The Pentium Microprocessor

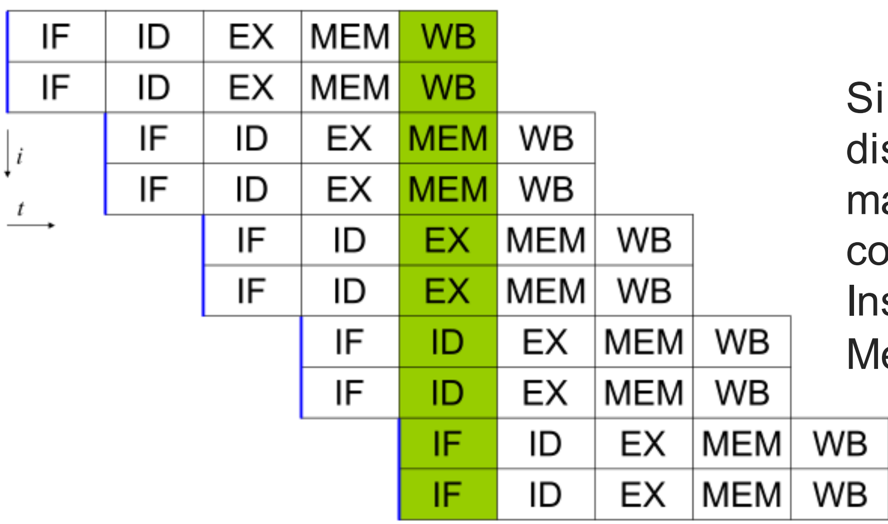
- Introduced 1993, Pentium was similar to 80386 and 80486 microprocessors.
- Originally labeled the P5 or 80586.
  - Intel decided not to use a number because it appeared to be impossible to copyright a number
- Introductory versions operated with a clocking frequency of 60 MHz & 66 MHz, and a speed of 110 MIPs.

- Double-clocked Pentium at 120 MHz and 133 MHz, also available.
  - fastest version produced 233 MHz Pentium
    a three and one-half clocked version
- Cache size was increased to 16K bytes from the 8K cache found in 80486.
- 8K-byte instruction cache and data cache.
- Memory system up to 4G bytes.
- Data bus width increased to a full 64 bits.
- Data bus transfer speed 60 MHz or 66 MHz.
  - depending on the version of the Pentium

- Wider data bus width accommodated double-precision floating-point numbers used in high-speed, vector-generated graphical displays.
  - should allow virtual reality software and video to operate at more realistic rates
- Widened data bus and higher speed allow full-frame video displays at scan rates of 30 Hz or higher.
  - comparable to commercial television

- Recent Pentium versions also included additional instructions.
  - MMX instructions (just a name, meaningless; multimedia extensions)
- Intel hoped MMX would be widely used
  - few software companies have used
  - no high-level language support for instructions
- OverDrive (P24T) for older 80486 systems.
  - The OverDrives typically possessed qualities different from 'standard' i486s with the same speed steppings.
  - Those included built-in voltage regulators, different pin-outs, write-back cache instead of write-through cache, built-in heatsinks, and fanless operation
- 63 MHz version upgrades 80486DX2 50 MHz systems; 83 MHz upgrades 66 MHz systems.
  - system performs somewhere between a 66 MHz Pentium and a 75 MHz Pentium

- Pentium OverDrive represents ideal upgrade path from the 80486 to the Pentium.
  – executes two instructions not dependent on each other, simultaneously per clocking period
  – dual integer processors most ingenious feature
  – contains two independent internal integer processors called superscaler technology
  – A superscalar CPU architecture implements a form of parallelism called instruction-level parallelism within a single processor.



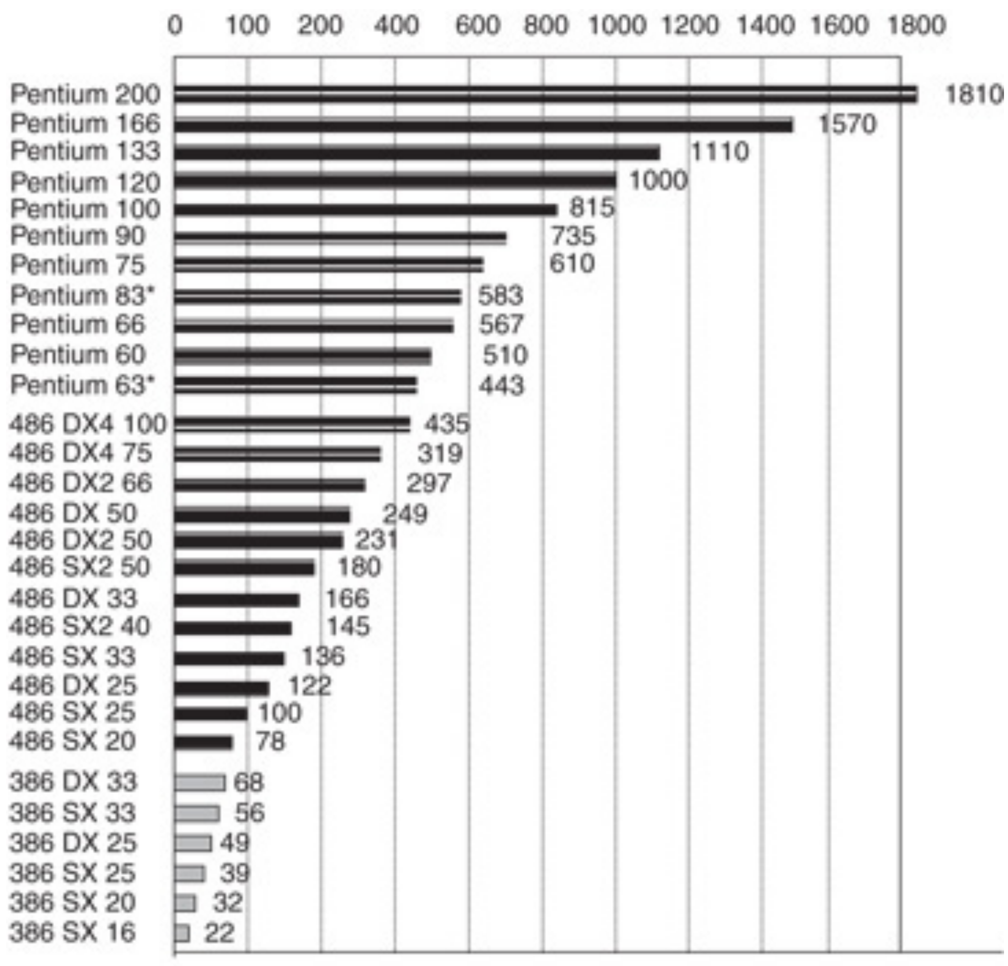
Simple superscalar pipeline. By fetching and dispatching two instructions at a time, a maximum of two instructions per cycle can be completed. (IF = Instruction Fetch, ID = Instruction Decode, EX = Execute, MEM = Memory access, WB = Register write back,

- Intel may allow Pentium to replace some RISC (**reduced instruction set computer**) machines.
- Some newer RISC processors execute more than one instruction per clock.
  - through superscaler technology
- Motorola, Apple, and IBM produce PowerPC, a RISC with two integer units and a floating-point unit.
  - boosts Macintosh performance, but slow to efficiently emulate Intel microprocessors

- Currently 6 million Apple Macintosh systems
- 260 million personal computers based on Intel microprocessors.
- 1998 reports showed 96% of all PCs shipped with the Windows operating system.
- Apple computer replaced PowerPC with the Intel Pentium in most of its computer systems.
  - appears that PowerPC could not keep pace with the Pentium line from Intel

- To compare speeds of microprocessors, Intel devised the iCOMP-rating index.
  - composite of SPEC92, ZD Bench, Power Meter
- The iCOMP1 rating index is used to rate the speed of all Intel microprocessors through the Pentium.
- Figure 1–2 shows relative speeds of the 80386DX 25 MHz version through the Pentium 233 MHz version.

Figure 1–2 The Intel iCOMP-rating index.



Note: *Pentium OverDrive, the first part of the scale is not linear, and the 166 MHz and 200 MHz are MMX technology.

- Since release of Pentium Pro and Pentium II, Intel has switched to the iCOMP2- rating.
  - scaled by a factor of 10 from the iCOMP1 index
- Figure 1–3 shows iCOMP2 index listing the Pentium III at speeds up to 1000 MHz.
- Figure 1–4 shows SYSmark 2002 for the Pentium III and Pentium 4.
- Intel has not released benchmarks that compare versions of the microprocessor since the SYSmark 2002.
  - newer available do not compare versions

Figure 1–3 The Intel iCOMP2-rating index.

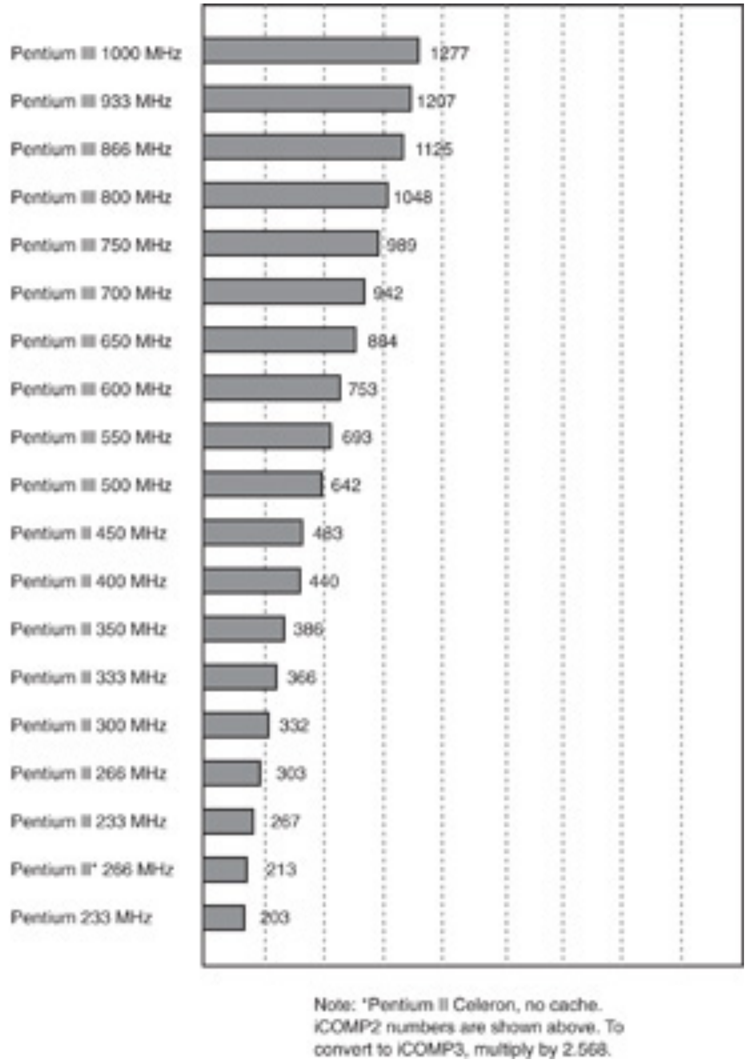
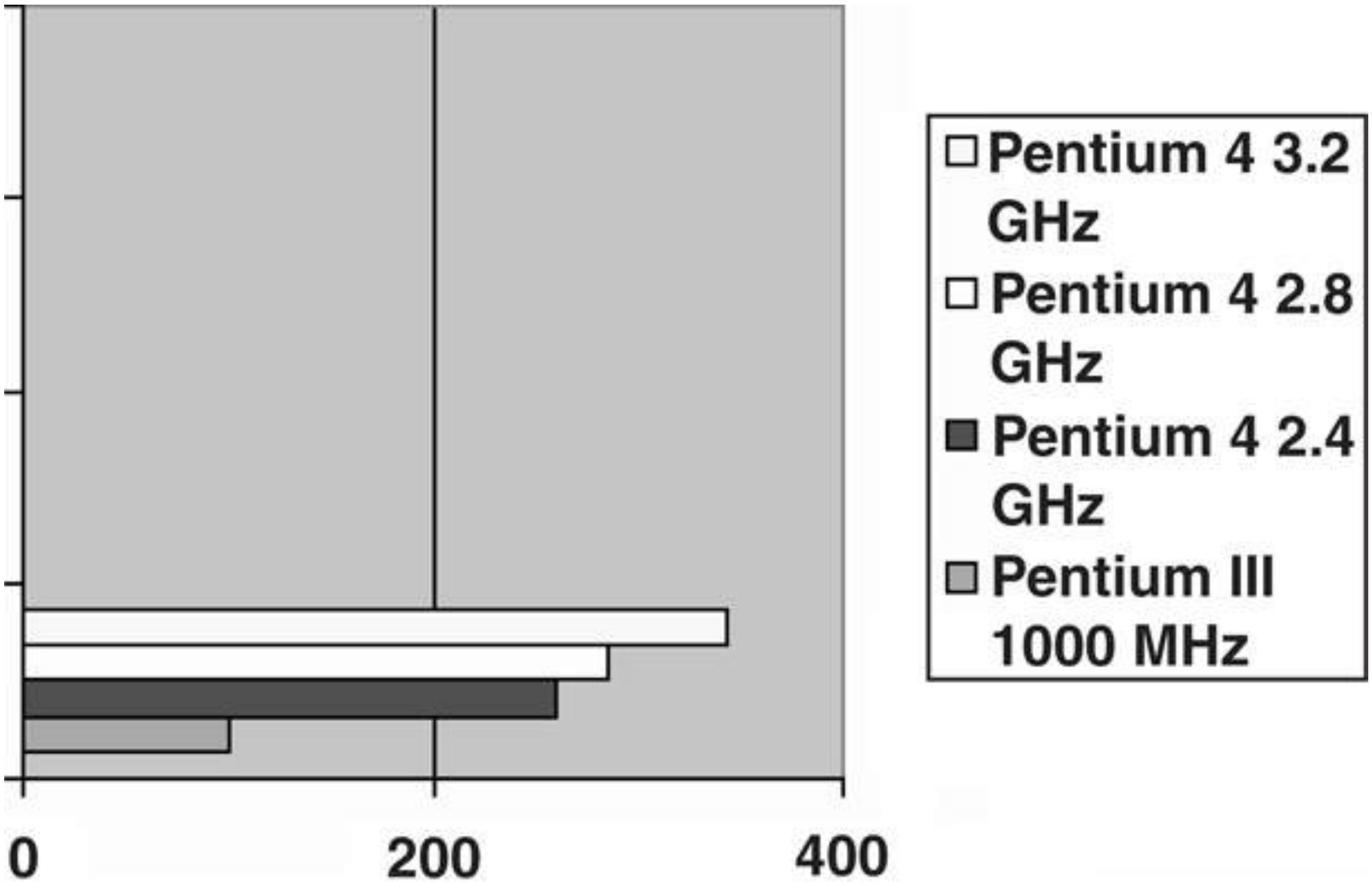


Figure 1–4 Intel microprocessor performance using SYSmark 2002.



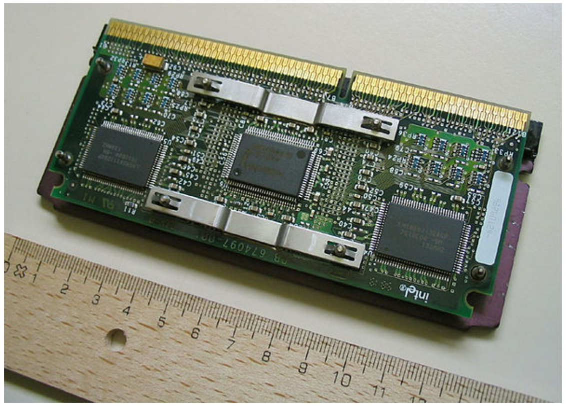
Pentium Pro Processor

- A recent entry, formerly named the P6.
- 21 million transistors, integer units, floating-point unit, clock frequency 150 and 166 MHz
- Internal 16K level-one (L1) cache.
  - 8K data, 8K for instructions
  - Pentium Pro contains 256K level-two (L2) cache
- Pentium Pro uses three execution engines, to execute up to three instructions at a time.
  - can conflict and still execute in parallel

- Pentium Pro optimized to efficiently execute 32-bit code.
  - often bundled with Windows NT rather than normal versions of Windows 95
  - Intel launched Pentium Pro for server market
- Pentium Pro can address 4G-byte or a 64G-byte memory system.
  - 36-bit address bus if configured for a 64G memory system

Pentium II and Pentium Xeon Microprocessors

- Pentium II, released 1997, represents new direction for Intel.
- Intel has placed Pentium II on a small circuit board, instead of being an integrated circuit.
  - L2 cache on main circuit board of not fast enough to function properly with Pentium II
- Microprocessor on the Pentium II module actually Pentium Pro with MMX extensions.



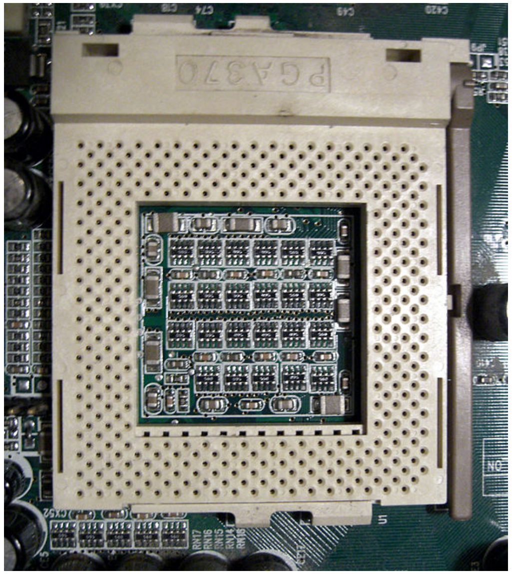
- In 1998 Intel changed Pentium II bus speed.
  - newer Pentium II uses a 100 MHz bus speed
- Higher speed memory bus requires 8 ns SDRAM.
  - replaces 10 ns SDRAM with 66 MHz bus speed

- Intel announced Xeon in mid-1998.
  - specifically designed for high-end workstation and server applications
- Xeon available with 32K L1 cache and L2 cache size of 512K, 1M, or 2M bytes.
- Xeon functions with the 440GX chip set.
- Also designed to function with four Xeons in the same system, similar to Pentium Pro.
- Newer product represents strategy change.
  - Intel produces a professional and home/business version of the Pentium II

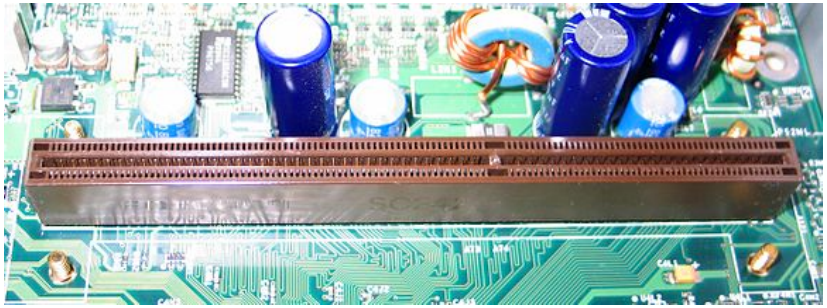
Pentium III Microprocessor

- Faster core than Pentium II; still a P6 or Pentium Pro processor.
- Available in slot 1 version mounted on a plastic cartridge.
- Also socket 370 version called a flip-chip which looks like older Pentium package.
- Pentium III available with clock frequencies up to 1 GHz.

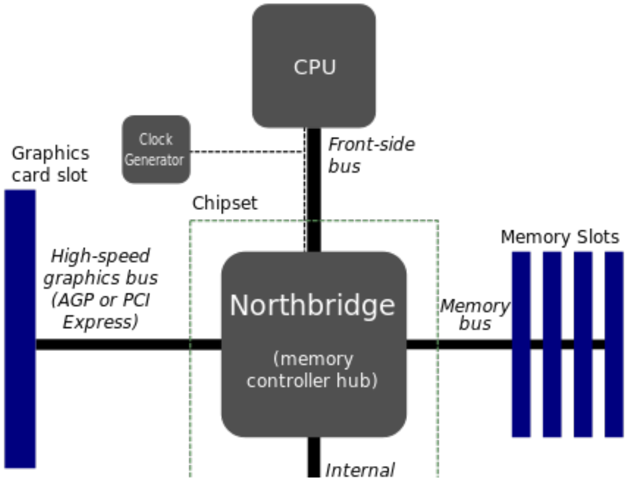
socket 370



slot 1



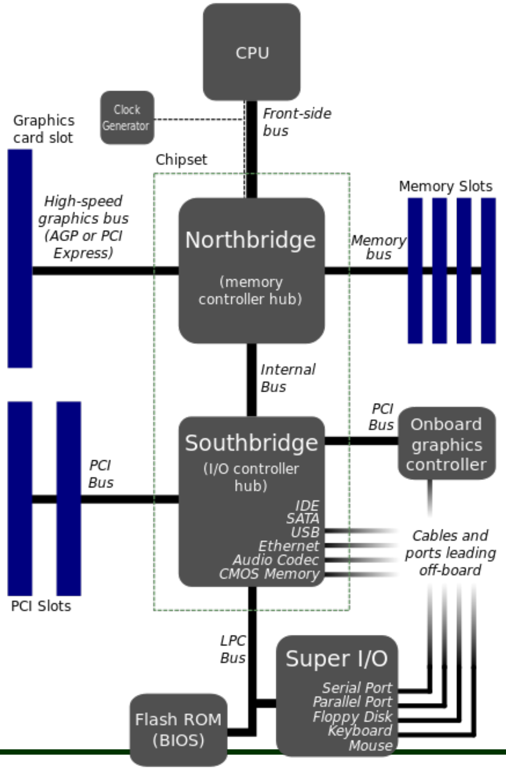
- Slot 1 version contains a 512K cache; flip-chip version contains 256K cache.
- Flip-chip version runs at clock speed; Slot 1 cache version runs at one-half clock speed (Downgrade).
- Both versions use 100 Mhz memory bus.
  - Celeron memory bus clock speed 66 MHz
- Front side bus (FSB) connection, microprocessor to memory controller, PCI controller, and AGP controller, now either 100 or 133 MHz.
  - this change has improved performance
  - memory still runs at 100 MHz



Pentium 4 and Core2 Microprocessors

- Pentium 4 first made available in late 2000.
  - most recent version of Pentium called Core2
  - uses Intel P6 architecture
- Pentium 4 available to 3.2 GHz and faster.
  - supporting chip sets use RAMBUS or DDR memory in place of SDRAM technology
- Core2 is available at speeds of up to 3 GHz.
  - improvement in internal integration, at present the 0.045 micron or 45 nm technology

- A likely change is a shift from aluminum to copper interconnections inside the microprocessor.
- Copper is a better conductor.
  - should allow increased clock frequencies
  - especially true now that a method for using copper has surfaced at IBM
- Another event to look for is a change in the speed of the front side bus.
  - increase beyond current maximum 1033 MHz



Pentium 4 and Core2, 64-bit and Multiple Core Microprocessors

- Recent modifications to Pentium 4 and Core2 include a 64-bit core and multiple cores.
- 64-bit modification allows address of over 4G bytes of memory through a 64-bit address.
  - 40 address pins in these newer versions allow up to 1T (terabytes) of memory to be accessed
- Also allows 64-bit integer arithmetic.
  - less important than ability to address more memory

- Biggest advancement is inclusion of multiple cores.
  - each core executes a separate task in a program
- Increases speed of execution if program is written to take advantage of multiple cores.
  - called **multithreaded** applications
- Intel manufactures dual and quad core versions; number of cores will likely increase to eight or even sixteen.

Multi-threaded application

Java Thread Example

Here is a small example. First it prints out the name of the thread executing the `main()` method. This thread is assigned by the JVM. Then it starts up 10 threads and give them all a number as name (`"" + i`). Each thread then prints its name out, and then stops executing.

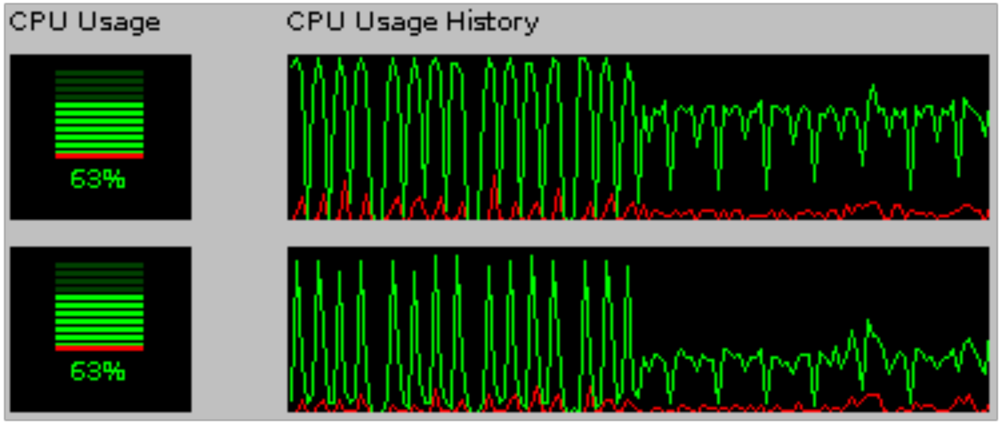
```
public class ThreadExample {

  public static void main(String[] args){
    System.out.println(Thread.currentThread().getName());
    for(int i=0; i<10; i++){
      new Thread("" + i){
        public void run(){
          System.out.println("Thread: " + getName() + " running");
        }
      }.start();
    }
  }
}
```

- Multiple cores are current solution to providing faster microprocessors.
- Intel recently demonstrated Core2 containing 80 cores, using 45 nm fabrication technology.
- Intel expects to release an 80-core version some time in the next 5 years.
  - http://www.cnet.com/news/intel-shows-off-80-core-processor/
- Fabrication technology will become slightly smaller with 35 nm and possibly 25 nm technology.

The Future of Microprocessors

- No one can make accurate predictions.
- Success of Intel should continue.
- Change to RISC technology may occur; more likely improvements to new hyper-threading (i.e., two virtual cores) technology.
  - joint effort by Intel and Hewlett-Packard
- New technology embodies CISC instruction set of 80X86 family.
  - software for the system will survive



- Basic premise is many microprocessors communicate directly with each other.
  - allows parallel processing without any change to the instruction set or program
- Current superscaler technology uses many microprocessors; all share same register set.
  - new technology contains many microprocessors
  - each contains its own register set linked with the other microprocessors' registers
- Offers true parallel processing without writing any special program.

- In 2002, Intel released a new architecture 64 bits in width with a 128-bit data bus.
- Named Itanium; joint venture called EPIC (Explicitly Parallel Instruction Computing) of Intel and Hewlett-Packard.
- The Itanium architecture allows greater parallelism than traditional architectures.
- 128 general-purpose integer and 128 floating-point registers; 64 predicate registers.
- Many execution units to ensure enough hardware resources for software.

Figure 1–5a Conceptual views of the 80486, Pentium Pro, Pentium II, Pentium III, Pentium 4, and Core2 microprocessors.

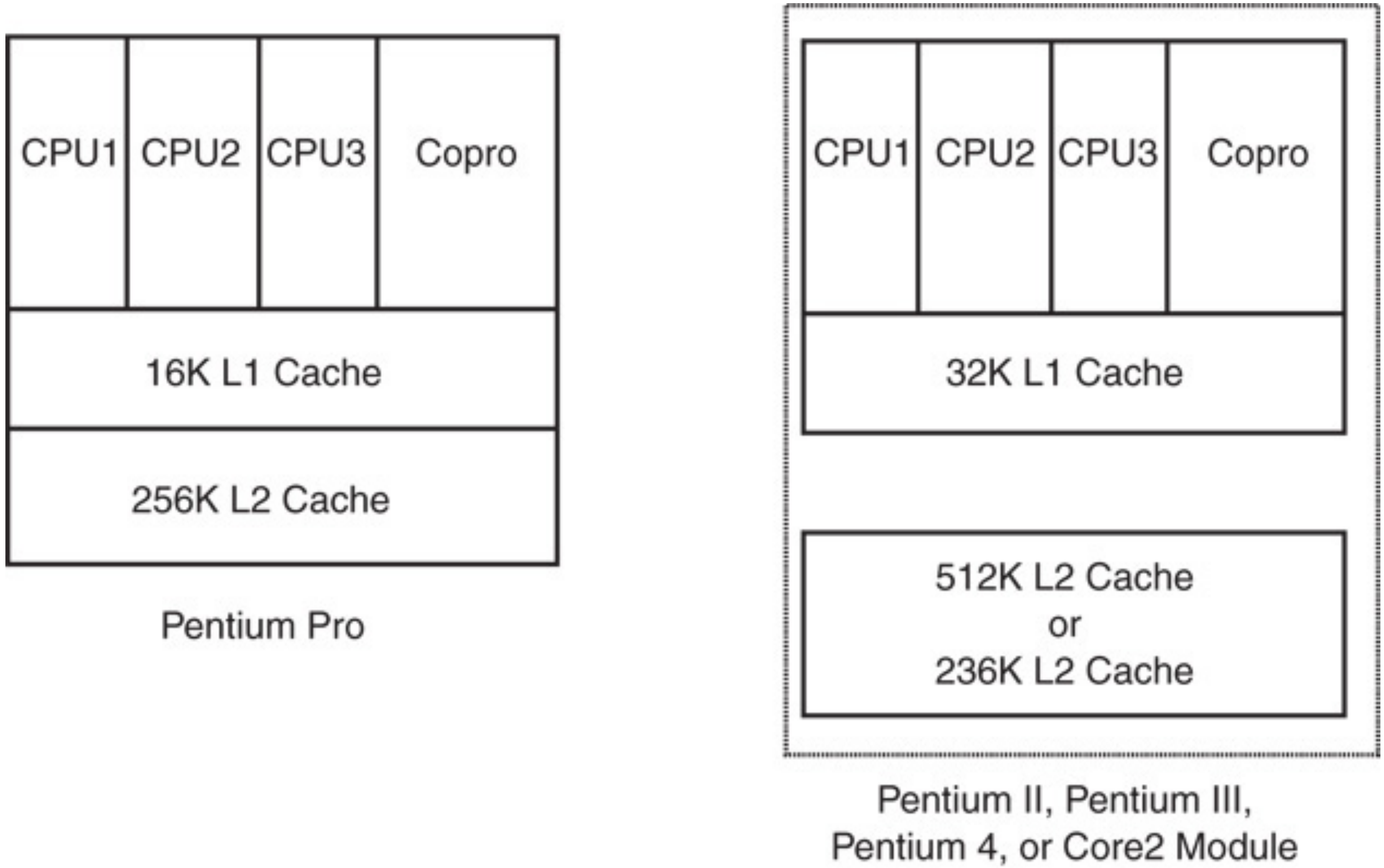
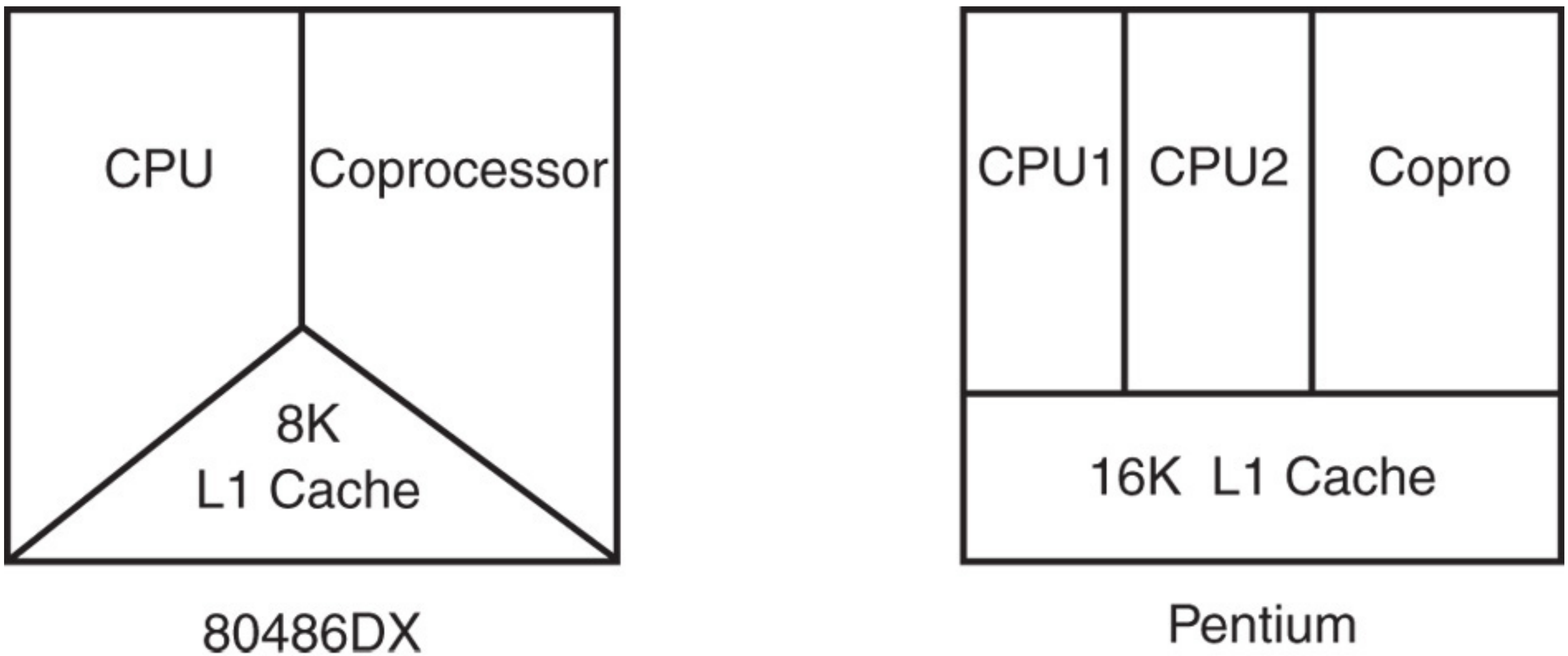


Figure 1–5b Conceptual views of the 80486, Pentium Pro, Pentium II, Pentium III, Pentium 4, and Core2 microprocessors.



- Clock frequencies seemed to have peaked.
- Surge to multiple cores has begun.
- Memory speed a consideration.
  - speed of dynamic RAM memory has not changed for many years.
- Push to static RAM memory will eventually. increase the performance of the PC.
  - main problem with large static RAM is heat
  - static RAM operates 50 times faster than dynamic RAM

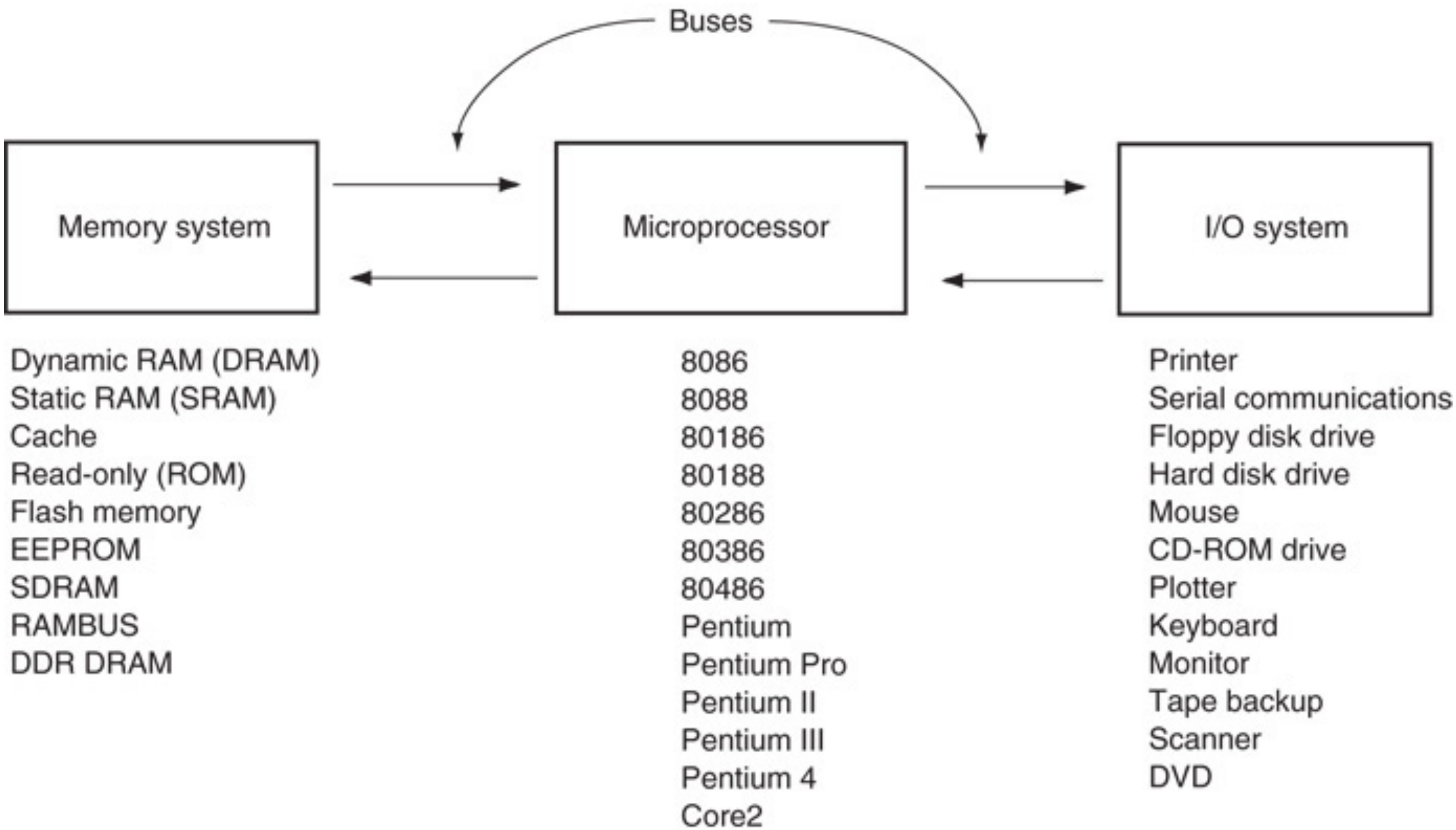
- Speed of mass storage another problem.
  - transfer speed of hard disk drives has changed little in past few years
  - new technology needed for mass storage
- Flash memory could be solution.
  - write speed comparable to hard disk memory
- Flash memory could store the operation system for common applications.
  - would allow operating system to load in a second or two instead of many seconds now required

1–2 THE MICROPROCESSOR-BASED PERSONAL COMPUTER SYSTEM

- Computers have undergone many changes recently.
- Machines that once filled large areas reduced to small desktop computer systems because of the microprocessor.
  - although compact, they possess computing power only dreamed of a few years ago

- Figure 1–6 shows block diagram of the personal computer.
- Applies to any computer system, from early mainframe computers to the latest systems.
- Diagram composed of three blocks interconnected by buses.
  - a **bus** is the set of common connections that carry the same type of information

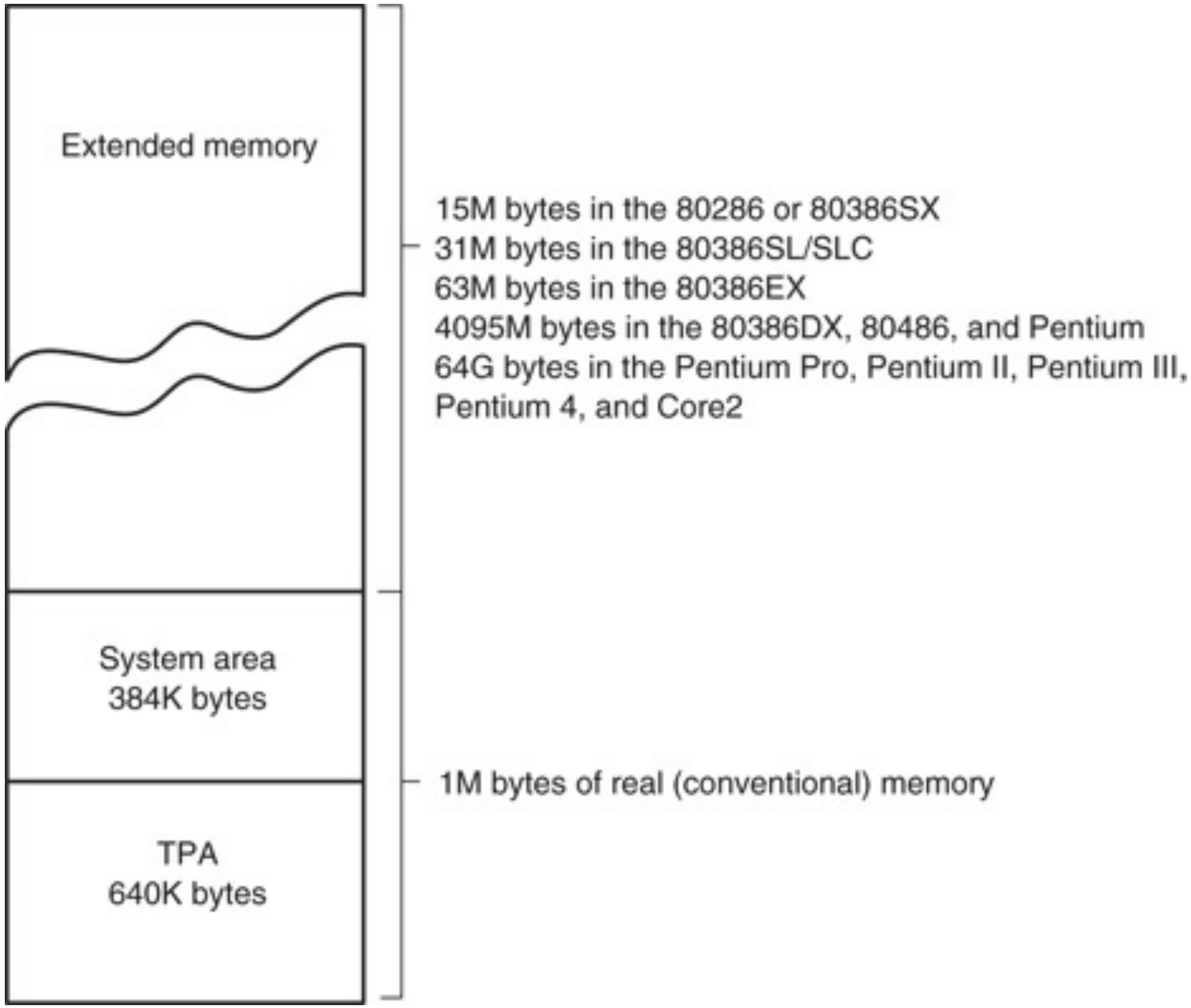
Figure 1–6 The block diagram of a microprocessor-based computer system.



The Memory and I/O System

- Memory structure of all Intel-based personal computers similar.
- Figure 1–7 illustrates memory map of a personal computer system.
- This map applies to any IBM personal computer.
  - also any IBM-compatible clones in existence

Figure 1–7 The memory map of a personal computer.



PEARSON
The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Pro Processor, Pentium II, Pentium, 4, and Core2 with 64-bit Extensions Architecture, Programming, and Interfacing, Eighth Edition
Barry B. Brey

- Main memory system divided into three parts:
  - TPA (transient program area)
  - system area
  - XMS (extended memory system)
- Type of microprocessor present determines whether an extended memory system exists.
- First 1M byte of memory often called the real or conventional memory system.
  - Intel microprocessors designed to function in this area using real mode operation

- 80286 through the Core2 contain the TPA (640K bytes) and system area (384K bytes).
  - also contain extended memory
  - often called AT class machines
- The PS/l and PS/2 by IBM are other versions of the same basic memory design.
- Also referred to as ISA (industry standard architecture) or EISA (extended ISA).
- The PS/2 referred to as a micro-channel architecture or ISA system.
  - depending on the model number

- Pentium and ATX class machines feature addition of the PCI (**peripheral component interconnect**) bus.
  - now used in all Pentium through Core2 systems
- Extended memory up to 15M bytes in the 80286 and 80386SX; 4095M bytes in 80486 80386DX, Pentium microprocessors.
- The Pentium Pro through Core2 computer systems have up to 1M less than 4G or 1 M less than 64G of extended memory.
- Servers tend to use the larger memory map.

- Many 80486 systems use **VESA** local, VL bus to interface disk and video to the microprocessor at the local bus level.
  - allows 32-bit interfaces to function at same clocking speed as the microprocessor
  - recent modification supporting 64-bit data bus has generated little interest
- ISA/EISA standards function at 8 MHz.
- PCI bus is a 32- or 64-bit bus.
  - specifically designed to function with the Pentium through Core2 at a bus speed of 33 MHz.

- Three newer buses have appeared.
- USB (**universal serial bus**).
  - intended to connect peripheral devices to the microprocessor through a serial data path and
    a twisted pair of wires
- Data transfer rates are 10 Mbps for USB1.
- Increase to 480 Mbps in USB2.

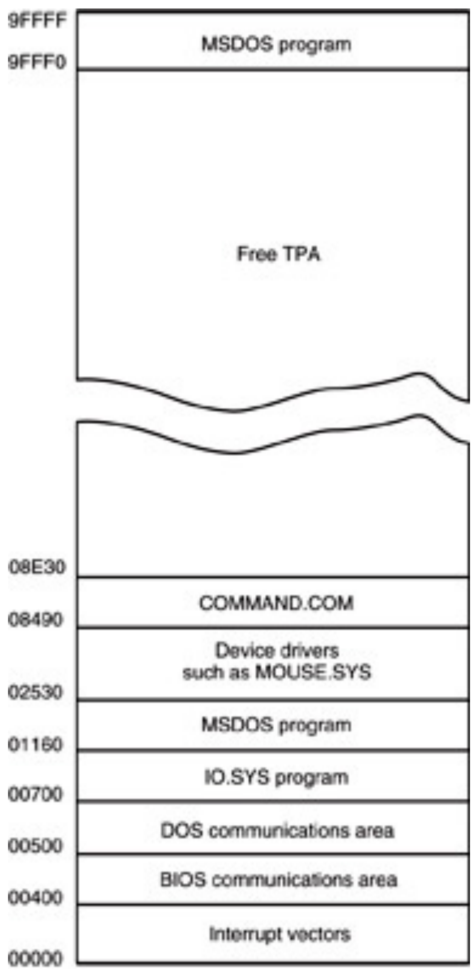
- AGP (**advanced graphics port**) for video cards.
- The port transfers data between video card and microprocessor at higher speeds.
  - 66 MHz, with 64-bit data path
- Latest AGP speed 8X or 2G bytes/second.
  - video subsystem change made to accommodate new DVD players for the PC.

- Latest new buses are serial ATA interface (**SATA**) for hard disk drives; PCI Express bus for the video card.
- The SATA bus transfers data from PC to hard disk at rates of 150M bytes per second; 300M bytes for SATA-2.
  - serial ATA standard will eventually reach speeds of 450M bytes per second
- PCI Express bus video cards operate at 16X speeds today.

The TPA

- The transient program area (TPA) holds the DOS (**disk operating system**) operating system; other programs that control the computer system.
  - the TPA is a DOS concept and not really applicable in Windows
  - also stores any currently active or inactive DOS application programs
  - length of the TPA is 640K bytes

Figure 1–8 The memory map of the TPA in a personal computer. (Note that this map will vary between systems.)



- DOS memory map shows how areas of TPA are used for system programs, data and drivers.
  - also shows a large area of memory available for application programs
  - hexadecimal number to left of each area represents the memory addresses that begin and end each data area

- Hexadecimal memory addresses number each byte of the memory system.
  - a hexadecimal number is a number represented in radix 16 or base 16
  - each digit represents a value from 0 to 9 and from A to F
- Often a hexadecimal number ends with an H to indicate it is a hexadecimal value.
  - 1234H is 1234 hexadecimal
  - also represent hexadecimal data as 0xl234 for a 1234 hexadecimal

- Interrupt vectors access DOS, BIOS (basic I/O system), and applications.
- Areas contain transient data to access I/O devices and internal features of the system.
  - these are stored in the TPA so they can be changed as DOS operates

- The IO.SYS loads into the TPA from the disk whenever an MSDOS system is started.
- IO.SYS contains programs that allow DOS to use keyboard, video display, printer, and other I/O devices often found in computers.
- The IO.SYS program links DOS to the programs stored on the system BIOS ROM.

- **Drivers** are programs that control installable I/O devices.
  - mouse, disk cache, hand scanner, CD-ROM memory (**Compact Disk Read-Only Memory**), DVD (**Digital Versatile Disk**), or installable devices, as well as programs
- Installable drivers control or drive devices or programs added to the computer system.
- DOS drivers normally have an extension of .SYS; MOUSE.SYS.
- DOS version 3.2 and later files have an extension of .EXE; EMM386.EXE.

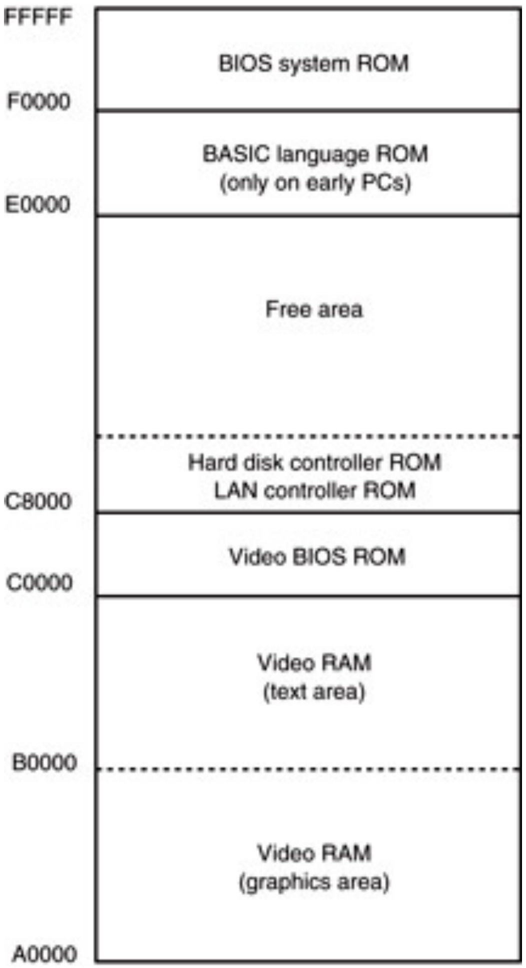
- Though not used by Windows, still used to execute DOS applications, even with Win XP.
- Windows uses a file called SYSTEM.INI to load drivers used by Windows.
- Newer versions of Windows have a registry added to contain information about the system and the drivers used.
- You can view the registry with the REGEDIT program.

- COMMAND.COM (**command processor**) controls operation of the computer from the keyboard when operated in the DOS mode.
- COMMAND.COM processes DOS commands as they are typed from the keyboard.
- If COMMAND.COM is erased, the computer cannot be used from the keyboard in DOS mode.
  - never erase COMMAND.COM, IO.SYS, or MSDOS.SYS to make room for other software
  - your computer will not function

The System Area

- Smaller than the TPA; just as important.
- The **system area** contains programs on read-only (ROM) or flash memory, and areas of read/write (RAM) memory for data storage.
- Figure 1–9 shows the system area of a typical personal computer system.
- As with the map of the TPA, this map also includes the hexadecimal memory addresses of the various areas.

Figure 1–9 The system area of a typical personal computer.



- First area of system space contains video display RAM and video control programs on ROM or flash memory.
  - area starts at location A0000H and extends to C7FFFH
  - size/amount of memory depends on type of video display adapter attached

- Display adapters generally have video RAM at A0000H–AFFFFH.
  - stores graphical or bit-mapped data
- Memory at B0000H–BFFFFH stores text data.
- The video BIOS on a ROM or flash memory, is at locations C0000H–C7FFFH.
  - contains programs to control DOS video display
- C8000H–DFFFFH is often open or free.
  - used for expanded memory system (EMS) in PC or XT system; upper memory system in an AT

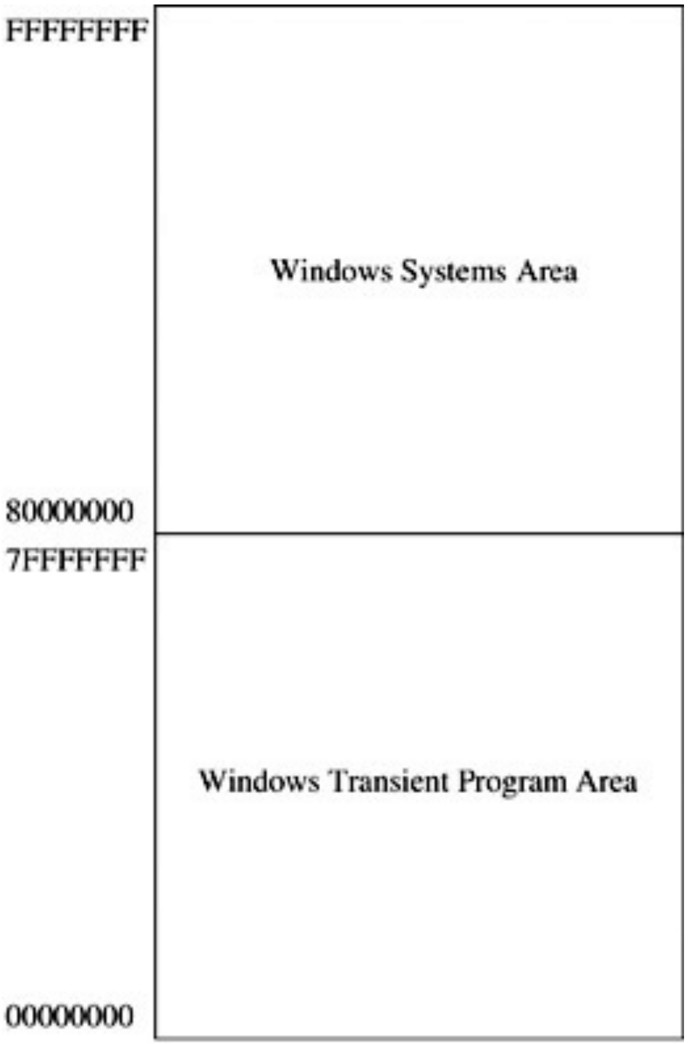
- Expanded memory system allows a 64K-byte page frame of memory for use by applications.
  - page frame (D0000H - DFFFFH) used to expand memory system by switching in pages of memory from EMS into this range of memory addresses
- Locations E0000H–EFFFFH contain cassette BASIC on ROM found in early IBM systems.
  - often open or free in newer computer systems
- Video system has its own BIOS ROM at location C0000H.

- System BIOS ROM is located in the top 64K bytes of the system area (F0000H–FFFFFH).
  - controls operation of basic I/O devices connected to the computer system
  - does not control operation of video
- The first part of the system BIOS (F0000H–F7FFFH) often contains programs that set up the computer.
- Second part contains procedures that control the basic I/O system.

Windows Systems

- Modern computers use a different memory map with Windows than DOS memory maps.
- The Windows memory map in Figure 1–10 has two main areas; a TPA and system area.
- The difference between it and the DOS memory map are sizes and locations of these areas.

Figure 1–10 The memory map used by Windows XP.



- TPA is first 2G bytes from locations 00000000H to 7FFFFFFFH.
- Every Windows program can use up to 2G bytes of memory located at linear addresses 00000000H through 7FFFFFFFH.
- System area is last 2G bytes from 80000000H
to FFFFFFFFH.

- Memory system physical map is much different.
- Every process in a Windows Vista, XP, or 2000 system has its own set of page tables.
- The process can be located anywhere in the memory, even in noncontiguous pages.
- The operating system assigns physical memory to application.
  - if not enough exists, it uses the hard disk for any that is not available

I/O Space

- I/O devices allow the microprocessor to communicate with the outside world.
- I/O (input/output) space in a computer system extends from I/O port 0000H to port FFFFH.
  - **I/O port address** is similar to a memory address
  - instead of memory, it addresses an I/O device
- Figure 1–11 shows the I/O map found in many personal computer systems.

Figure 1–11 Some I/O locations in a typical personal computer.

- Access to most I/O devices should always be made through Windows, DOS, or BIOS function calls.
- The map shown is provided as a guide to illustrate the I/O space in the system.

- The area below I/O location 0400H is considered reserved for system devices
- Area available for expansion extends from I/O port 0400H through FFFFH.
- Generally, 0000H - 00FFH addresses main board components; 0100H - 03FFH handles devices located on plug-in cards or also on the main board.
- The limitation of I/O addresses between 0000 and 03FFH comes from original standards specified by IBM for the PC standard.

The Microprocessor

- Called the CPU (**central processing unit**).
- The controlling element in a computer system.
- Controls memory and I/O through connections called buses.
  - buses select an I/O or memory device, transfer data between I/O devices or memory and the microprocessor, control I/O and memory systems
- Memory and I/O controlled via instructions stored in memory, executed by the microprocessor.

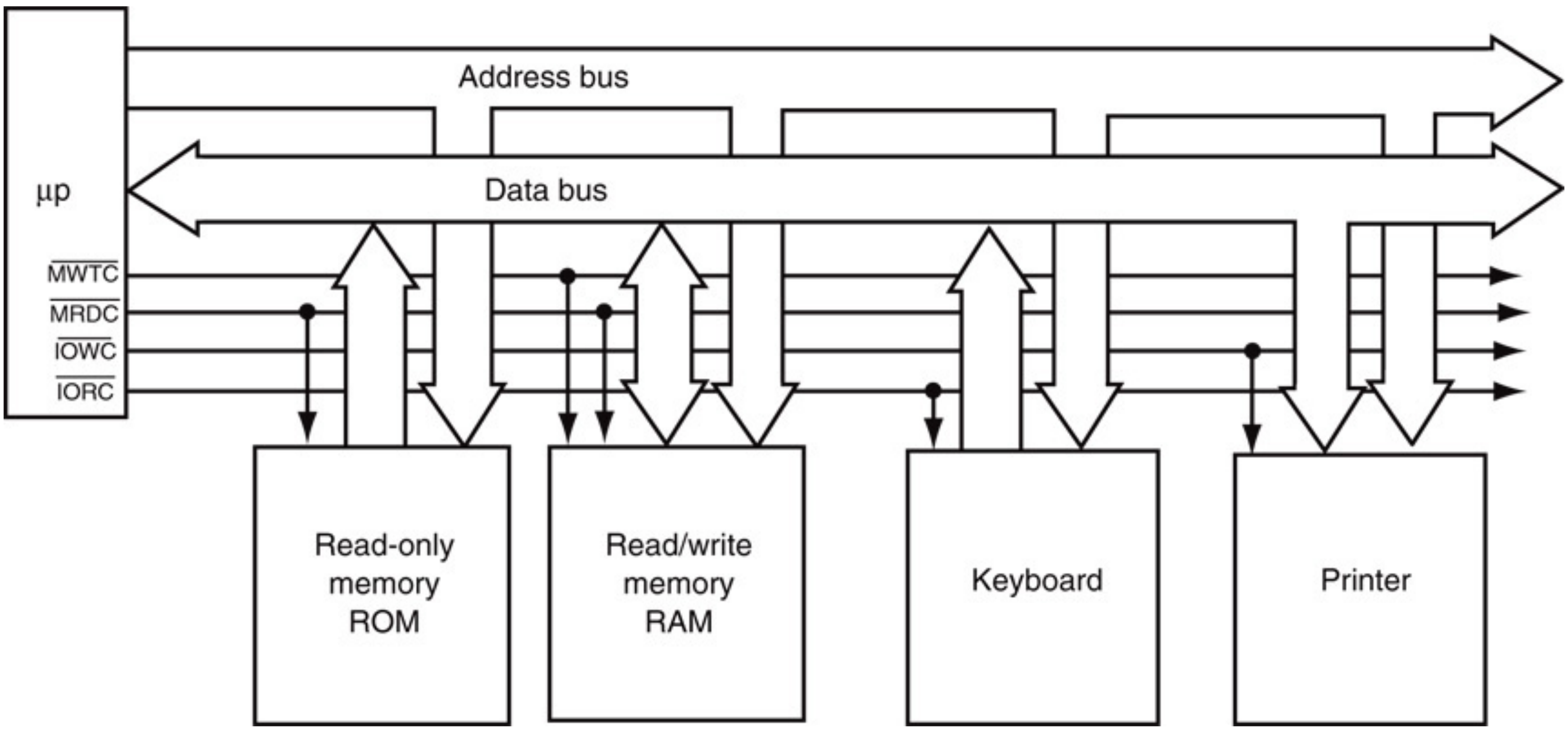
- Microprocessor performs three main tasks:
  - data transfer between itself and the memory or I/O systems
  - simple arithmetic and logic operations
  - program flow via simple decisions
- Power of the microprocessor is capability to execute billions of millions of instructions per second from a program or software (**group of instructions**) stored in the memory system.
  - stored programs make the microprocessor and computer system very powerful devices

- Another powerful feature is the ability to make simple decisions based upon numerical facts.
  - a microprocessor can decide if a number is zero, positive, and so forth
- These decisions allow the microprocessor to modify the program flow, so programs appear to think through these simple decisions.

Buses

- A common group of wires that interconnect components in a computer system.
- Transfer address, data, & control information between microprocessor, memory and I/O.
- Three buses exist for this transfer of information: address, data, and control.
- Figure 1–12 shows how these buses interconnect various system components.

Figure 1–12 The block diagram of a computer system showing the address, data, and control bus structure.



- The address bus requests a memory location from the memory or an I/O location from the I/O devices.
  - if I/O is addressed, the address bus contains a 16-bit I/O address from 0000H through FFFFH.
  - if memory is addressed, the bus contains a memory address, varying in width by type of
    microprocessor.
- 64-bit extensions to Pentium provide 40 address pins, allowing up to 1T byte of memory to be accessed.

- The data bus transfers information between the microprocessor and its memory and I/O address space.
- Data transfers vary in size, from 8 bits wide to 64 bits wide in various Intel microprocessors.
  - 8088 has an 8-bit data bus that transfers 8 bits of data at a time
  - 8086, 80286, 80386SL, 80386SX, and 80386EX transfer 16 bits of data
  - 80386DX, 80486SX, and 80486DX, 32 bits
  - Pentium through Core2 microprocessors transfer 64 bits of data

- Advantage of a wider data bus is speed in applications using wide data.
- Figure 1–13 shows memory widths and sizes of 8086 through Core2 microprocessors.
- In all Intel microprocessors family members, memory is numbered by byte.
- Pentium through Core2 microprocessors contain a 64-bit-wide data bus.

Figure 1–13a The physical memory systems of the 8086 through the Core2 microprocessors.

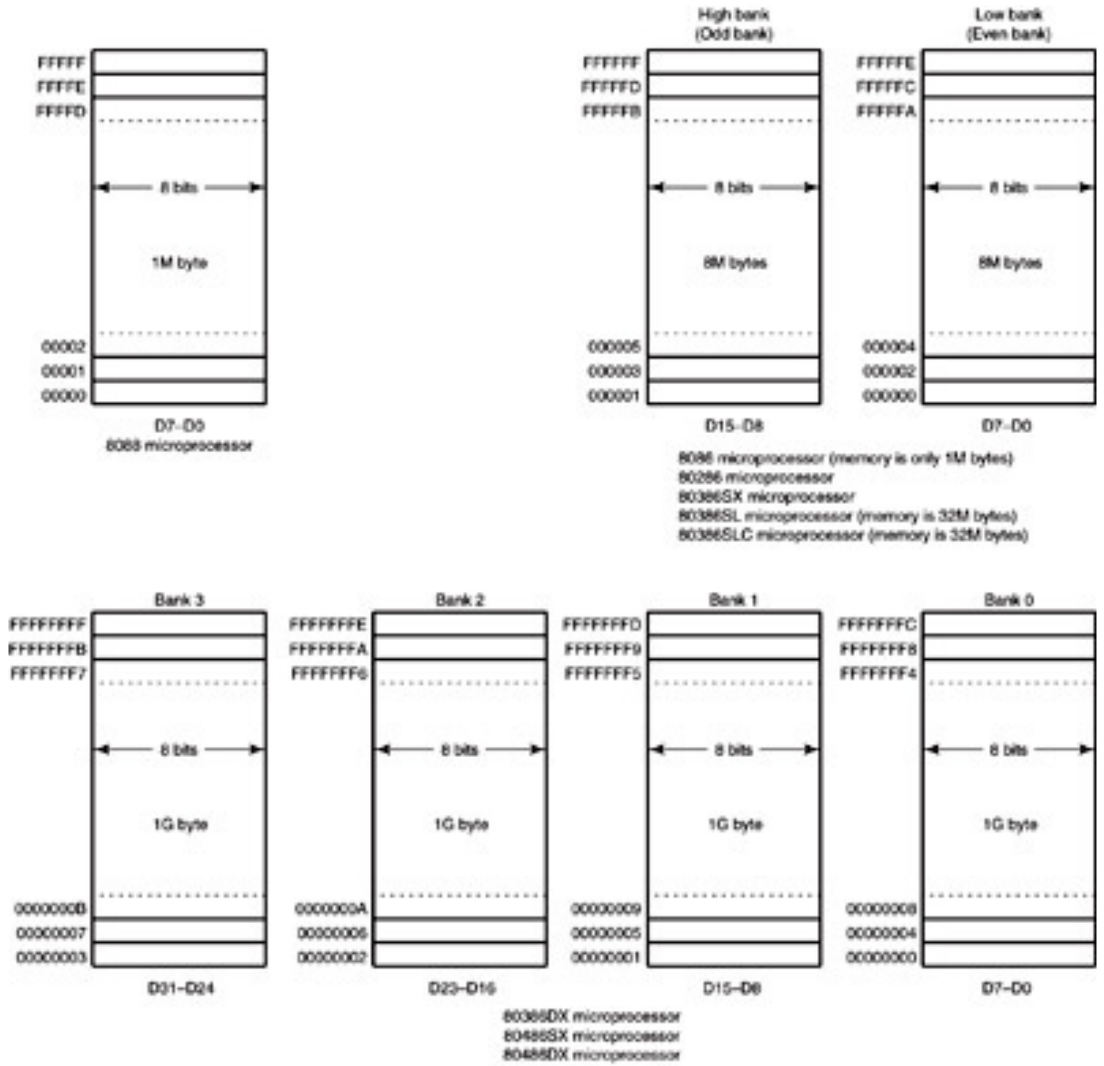
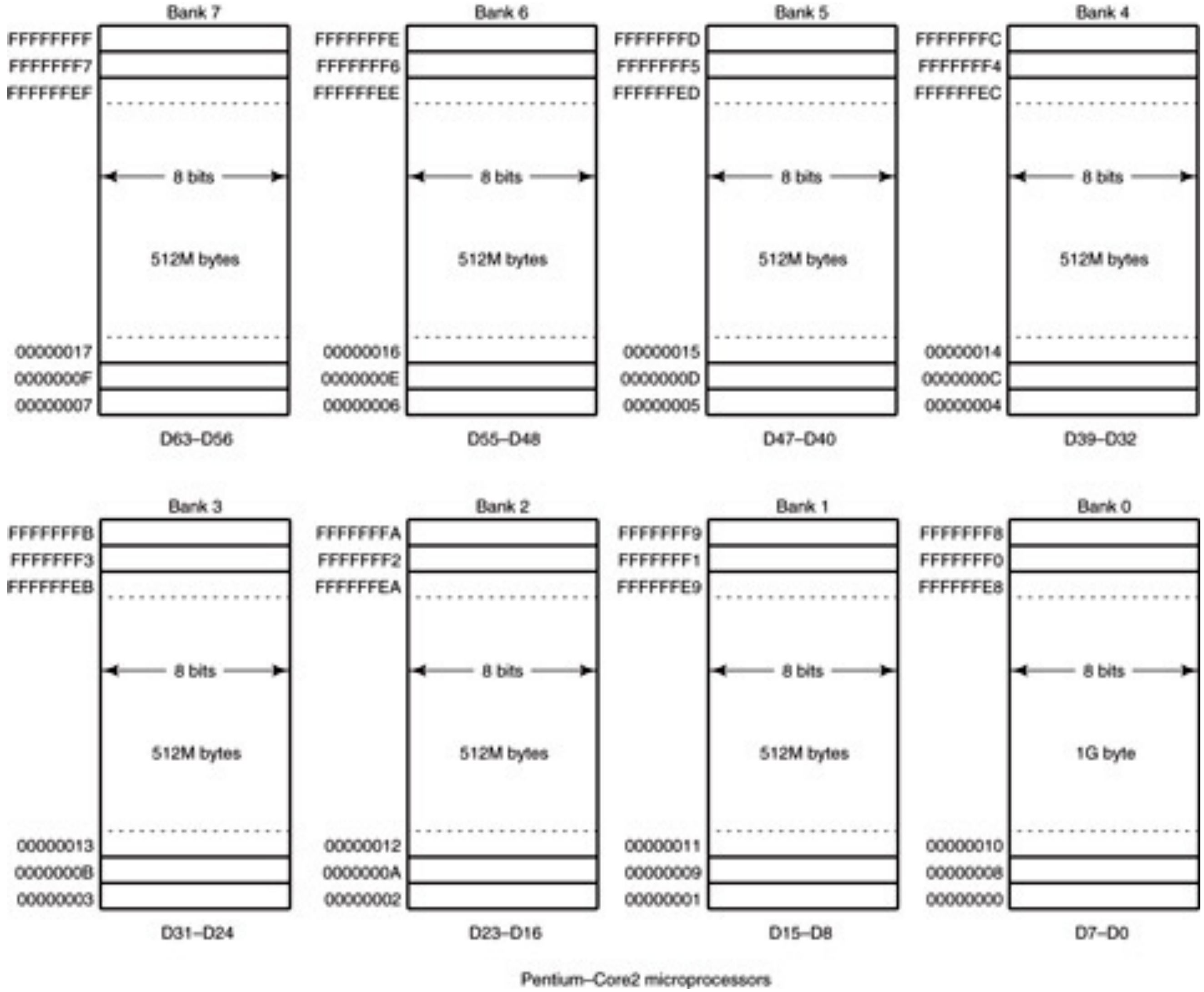


Figure 1–13b The physical memory systems of the 8086 through the Core2 microprocessors.



- Control bus lines select and cause memory or I/O to perform a read or write operation.
- In most computer systems, there are four control bus connections:
- $\overline{MRDC}$ **(memory read control)**
- $\overline{MWTC}$ **(memory write control)**
- $\overline{IORC}$ **(I/O read control)**
- $\overline{IOWC}$ **(I/O write control)**.
- overbar indicates the control signal is active-low; (active when logic zero appears on control line)

- The microprocessor reads a memory location by sending the memory an address through the address bus.
- Next, it sends a memory read control signal to cause the memory to read data.
- Data read from memory are passed to the microprocessor through the data bus.
- Whenever a memory write, I/O write, or I/O read occurs, the same sequence ensues.

1–3 NUMBER SYSTEMS

- Use of a microprocessor requires working knowledge of numbering systems.
  - binary, decimal, and hexadecimal
- This section provides a background for these numbering systems.
- Conversions are described.
  - decimal and binary
  - decimal and hexadecimal
  - binary and hexadecimal

Digits

- Before converting numbers between bases, digits of a number system must be understood.
- First digit in any numbering system is always zero.
- A decimal (base 10) number is constructed with 10 digits: 0 through 9.
- A base 8 (**octal**) number; 8 digits: 0 through 7.
- A base 2 (**binary**) number; 2 digits: 0 and 1.

- If the base exceeds 10, additional digits use letters of the alphabet, beginning with an A.
  - a base 12 number contains 10 digits: 0 through 9, followed by A for 10 and B for 11
- Note that a base 10 number does contain a *10* digit.
  - a base 8 number does not contain an *8* digit
- Common systems used with computers are decimal, binary, and hexadecimal (base 16).
  - many years ago octal numbers were popular

Positional Notation

- Once digits are understood, larger numbers are constructed using positional notation.
  – position to the left of the units position is the tens position
  – left of tens is the hundreds position, and so forth
- An example is decimal number 132.
  – this number has 1 hundred, 3 tens, and 2 units
- Exponential powers of positions are critical for understanding numbers in other systems.

- Exponential value of each position:
  - the units position has a weight of 10^0, or 1
  - tens position a weight of 10^1, or 10
  - hundreds position has a weight of 10^2, or 100
- Position to the left of the radix (**number base**) point is always the units position in system.
  - called a decimal point only in the decimal system
  - position to left of the binary point always 2^0, or 1
  - position left of the octal point is 8^0, or 1
- Any number raised to its zero power is always one (1), or the units position.

- Position to the left of the units position always the number base raised to the first power.
  - in a decimal system, this is 10^1, or 10
  - binary system, it is 2^1, or 2
  - 11 decimal has a different value from 11 binary
- 11 decimal has different value from 11 binary.
  - decimal number composed of 1 ten, plus 1 unit;
    a value of 11 units
  - binary number 11 is composed of 1 two, plus 1 unit: a value of 3 decimal units
  - 11 octal has a value of 9 decimal units

- In the decimal system, positions right of the decimal point have negative powers.
  - first digit to the right of the decimal point has a value of 10^{-1}, or 0.1.
- In the binary system, the first digit to the right of the binary point has a value of 2^{-1}, or 0.5.
- Principles applying to decimal numbers also generally apply to those in any other system.
- To convert a binary number to decimal, add weights of each digit to form its decimal equivalent.

Conversion from Decimal

- To convert from any number base to decimal, determine the weights or values of each position of the number.
- Sum the weights to form the decimal equivalent.

Conversion to Decimal

- Conversions from decimal to other number systems more difficult to accomplish.
- To convert the whole number portion of a number to decimal, divide by 1 radix.
- To convert the fractional portion, multiply by the radix.

Whole Number Conversion from Decimal

- To convert a decimal whole number to another number system, divide by the radix and save remainders as significant digits of the result.
- An algorithm for this conversion:
  - divide the decimal number by the radix
    (number base)
  - save the remainder
    (first remainder is the least significant digit)
  - repeat steps 1 and 2 until the quotient is zero

- To convert 10 decimal to binary, divide it by 2.
  - the result is 5, with a remainder of 0
- First remainder is units position of the result.
  - in this example, a 0
- Next, divide the 5 by 2; result is 2, with a remainder of 1.
  - the 1 is the value of the twos (2^1) position
- Continue division until the quotient is a zero.
- The result is written as 1010_2 from the bottom to the top.

Example 1.5 Convert $(75)_{10}$ to its binary equivalent

2 | 75	Remainder	
2 | 37	1	
2 | 18	1	
2 | 9	0	Read in
2 | 4	1	reverse order
2 | 2	0	
1	0	

So, $(75)_{10}=(001011)_2$

- To convert 10 decimal to base 8, divide by 8.
  - a 10 decimal is a 12 octal.
- For decimal to hexadecimal, divide by 16.
  - remainders will range in value from 0 through 15
  - any remainder of 10 through 15 is converted to letters A through F for the hexadecimal number
  - decimal number 109 converts to a 6DH

Converting from a Decimal Fraction

- Conversion is accomplished with multiplication by the radix.
- Whole number portion of result is saved as a significant digit of the result.
  - fractional remainder again multiplied by the radix
  - when the fraction remainder is zero, multiplication ends
- Some numbers are never-ending (repetend).
  - a zero is never a remainder

- Algorithm for conversion from a decimal fraction:
  - multiply the decimal fraction by the radix (number base).
  - save the whole number portion of the result (even if zero) as a digit; first result is written immediately to the right of the radix point
  - repeat steps 1 and 2, using the fractional part of step 2 until the fractional part of step 2 is zero
- Same technique converts a decimal fraction into any number base.

Binary-Coded Hexadecimal

- **Binary-coded hexadecimal** (BCH) is a hexadecimal number written each digit is represented by a 4-bit binary number.
- BCH code allows a binary version of a hexadecimal number to be written in a form easily converted between BCH and hexadecimal.
- Hexadecimal represented by converting digits to BCH code with a space between each digit.

Complements

- At times, data are stored in complement form to represent negative numbers.
- Two systems used to represent negative data:
  - **radix**
  - **radix – 1** complement (earliest)

1–4 COMPUTER DATA FORMATS

- Successful programming requires a precise understanding of data formats.
- Commonly, data appear as ASCII, Unicode, BCD, signed and unsigned integers, and floating-point numbers (real numbers).
- Other forms are available but are not commonly found.

ASCII and Unicode Data

- ASCII (**American Standard Code for Information Interchange**) data represent alphanumeric characters in computer memory.
- Standard ASCII code is a 7-bit code.
  - eighth and most significant bit used to hold parity
- If used with a printer, most significant bits are 0 for alphanumeric printing; 1 for graphics.
- In PC, an extended ASCII character set is selected by placing 1 in the leftmost bit.

- Extended ASCII characters store:
  - some foreign letters and punctuation
  - Greek & mathematical characters
  - box-drawing & other special characters
- Extended characters can vary from one printer to another.
- ASCII control characters perform control functions in a computer system.
  - clear screen, backspace, line feed, etc.
- Enter control codes through the keyboard.
  - hold the Control key while typing a letter

- Many Windows-based applications use the **Unicode** system to store alphanumeric data.
  - stores each character as 16-bit data
- Codes 0000H–00FFH are the same as standard ASCII code.
- Remaining codes, 0100H–FFFFH, store all special characters from many character sets.
- Allows software for Windows to be used in many countries around the world.
- For complete information on Unicode, visit: *http://www.unicode.org*

BCD (Binary-Coded Decimal) Data

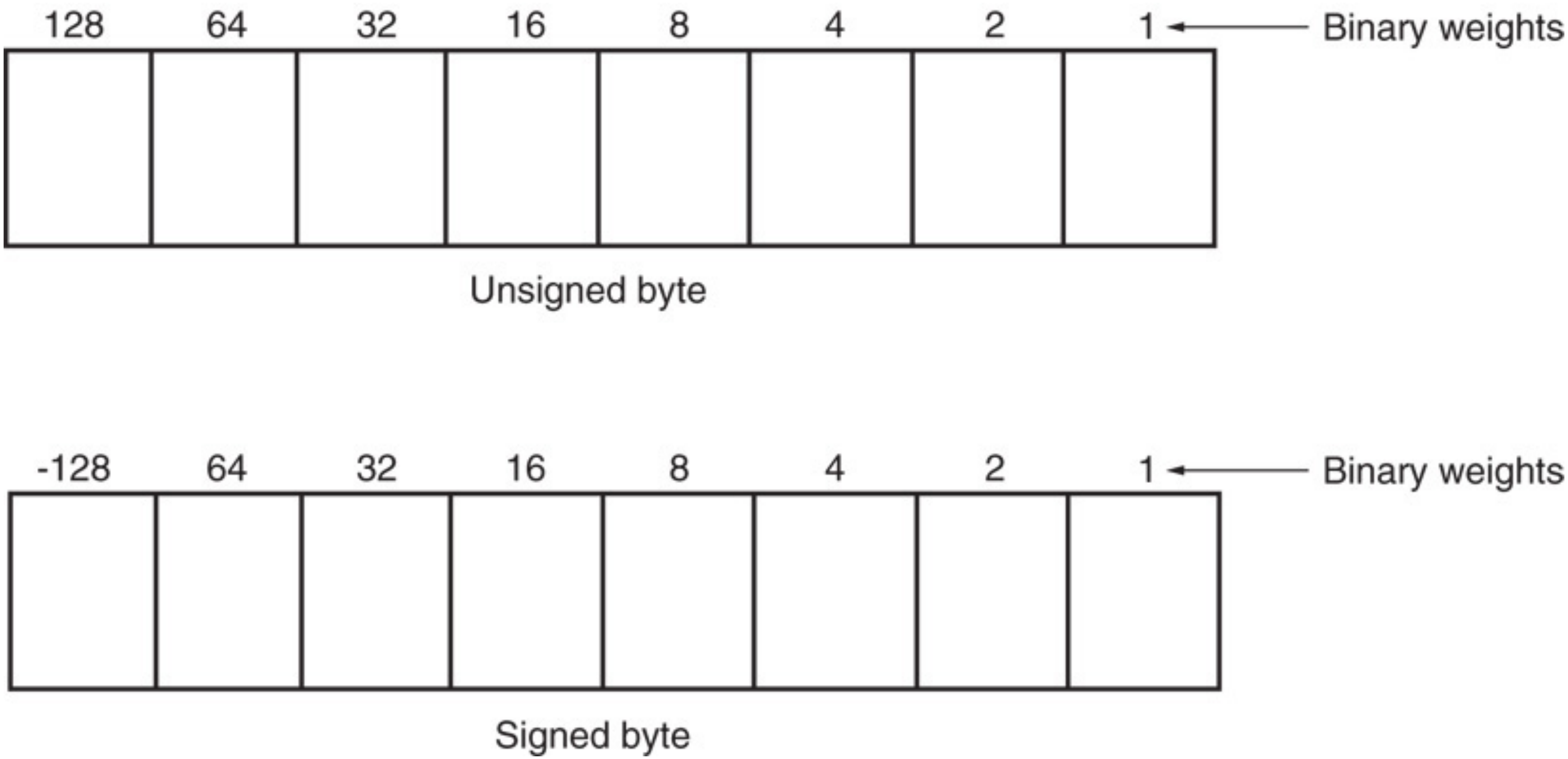
- The range of a BCD digit extends from 0000_2 to 1001_2, or 0–9 decimal, stored in two forms:
- Stored in packed form:
  - packed BCD data stored as two digits per byte;
  - used for BCD addition and subtraction in the instruction set of the microprocessor
- Stored in unpacked form:
  - unpacked BCD data stored as one digit per byte
  - returned from a keypad or keyboard

- Applications requiring BCD data are point-of-sales terminals.
  - also devices that perform a minimal amount of simple arithmetic
- If a system requires complex arithmetic, BCD data are seldom used.
  - there is no simple and efficient method of performing complex BCD arithmetic

Byte-Sized Data

- Stored as *unsigned* and *signed* integers.
- Difference in these forms is the weight of the leftmost bit position.
  - value 128 for the unsigned integer
  - *minus* 128 for the signed integer
- In signed integer format, the leftmost bit represents the sign bit of the number.
  - also a weight of *minus* 128

Figure 1–14 The unsigned and signed bytes illustrating the weights of each binary-bit position.

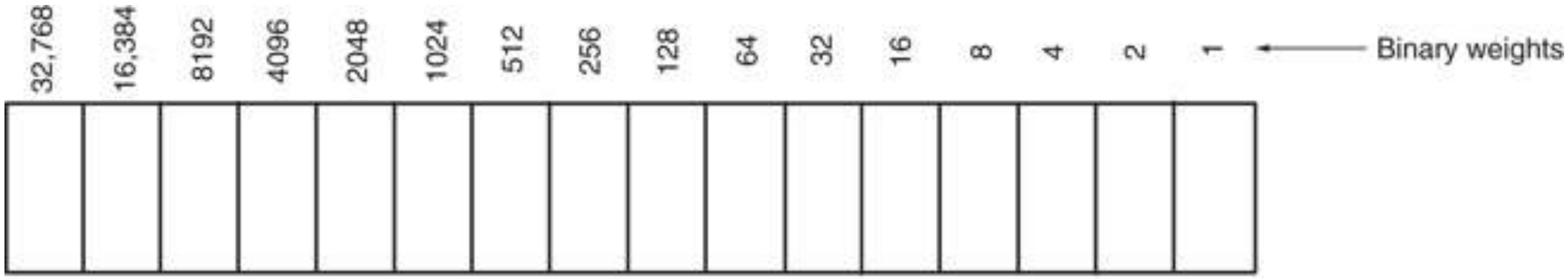


- Unsigned integers range 00H to FFH (0–255)
- Signed integers from –128 to 0 to + 127.
- Negative signed numbers represented in this way are stored in the two's complement form.
- Evaluating a signed number by using weights of each bit position is much easier than the act of two's complementing a number to find its value.
  - especially true in the world of calculators designed for programmers

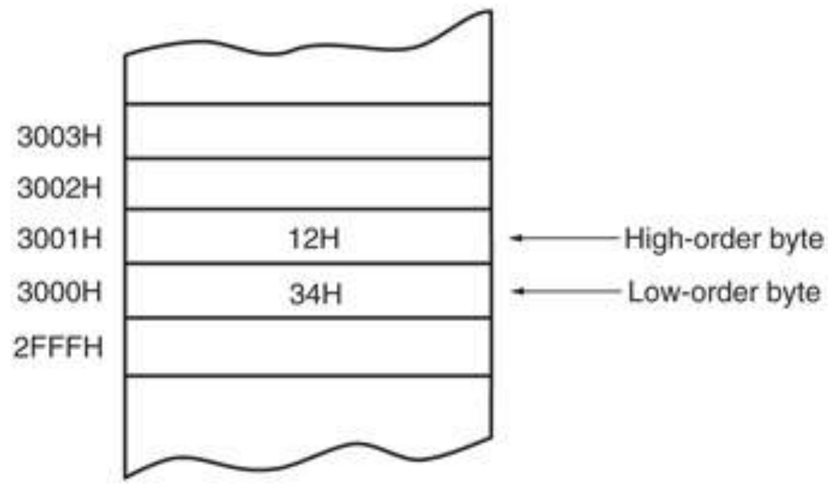
Word-Sized Data

- A word (16-bits) is formed with two bytes of data.
- The least significant byte always stored in the lowest-numbered memory location.
- Most significant byte is stored in the highest.
- This method of storing a number is called the **little endian** format.

Figure 1–15 The storage format for a 16-bit word in (a) a register and (b) two bytes of memory.

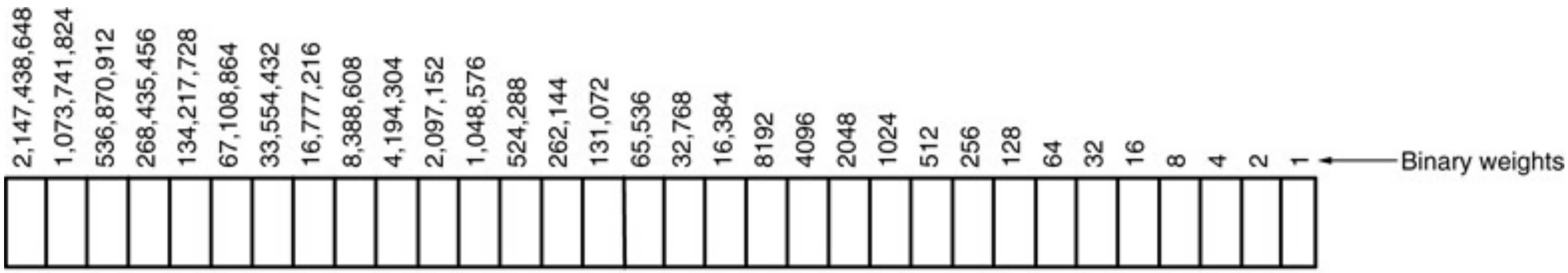


(a) Unsigned word

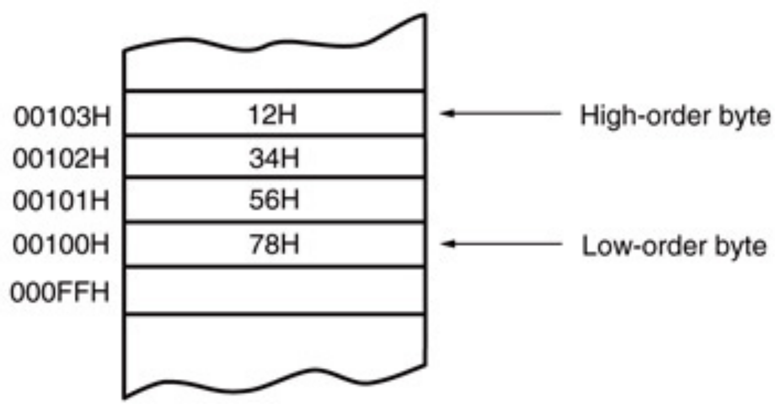


(b) The contents of memory location 3000H and 3001H are the word 1234H.

Figure 1–16 The storage format for a 32-bit word in (a) a register and (b) 4 bytes of memory.



(a) Unsigned doubleword



(b) The contents of memory location 00100H–00103H are the doubleword 12345678H.

- Alternate method is called the **big endian** format.
- Numbers are stored with the lowest location containing the most significant data.
- Not used with Intel microprocessors.
- The big endian format is used with the Motorola family of microprocessors.

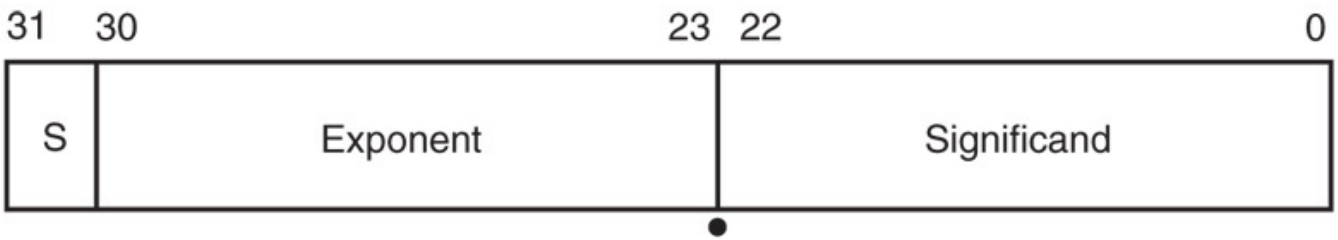
Doubleword-Sized Data

- **Doubleword-sized data** requires four bytes of memory because it is a 32-bit number.
  - appears as a product after a multiplication
  - also as a dividend before a division
- Define using the assembler directive **define doubleword(s)**, or **DD.**
  - also use the DWORD directive in place of DD

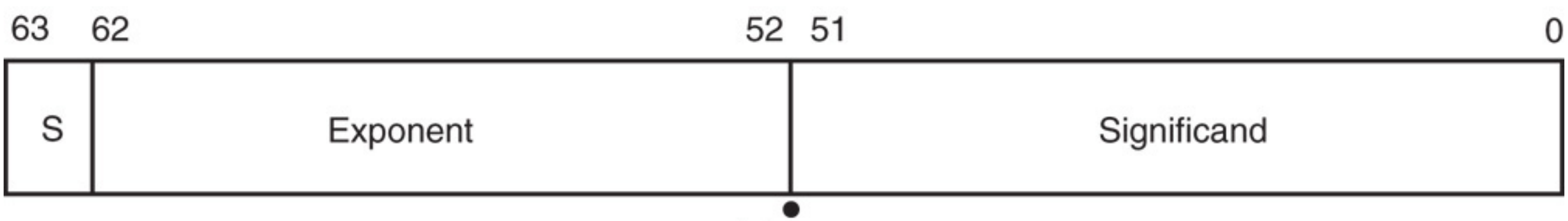
Real Numbers

- Since many high-level languages use Intel microprocessors, real numbers are often encountered.
- A real, or a **floating-point number** contains two parts:
  - a mantissa, significand, or fraction
  - an exponent.
- A 4-byte number is called **single-precision.**
- The 8-byte form is called **double-precision.**

Figure 1–17 The floating-point numbers in (a) single-precision using a bias of 7FH and (b) double-precision using a bias of 3FFH.



(a)



(b)

- The assembler can be used to define real numbers in single- & double-precision forms:
  - use the DD directive for single-precision 32-bit numbers
  - use **define quadword(s)**, or DQ to define 64-bit double-precision real numbers
- Optional directives are REAL4, REAL8, and REAL10.
  - for defining single-, double-, and extended precision real numbers