CS 320
Ch 1 pp. 2-27

Computer Architecture – studies attributes of a computer system which have a direct impact on the execution of a program. Examples include: instruction formats, opcodes, registers, memory, and the algorithm for controlling instruction execution.

Computer Organization – studies the operational units and their interconnections. Examples include: control signals that are invisible to the programmer, memory technology, memory addressing mechanism, etc.

A good example of the difference between architecture and organization can be found in the multiply instruction. The architectural issue is whether or not there will be an multiply instruction in the assembly language but it is an organizational issue as to whether that instruction will be done by a special hardware unit or by software doing shifts and adds in microcode.

A *computer family* typically has a very similar architecture but changes the organization to offer different levels of performance.

1. Why would we want a family of computers?
Different performance models allow wider sales.
2. If we need to decide how many bits are going to be used to represent an integer is that an architectural or an organizational issue? **Architectural**
3. Is the issue of having volatile or nonvolatile memory on a microcontroller an architectural issue or an organizational issue? **Organizational**

Computer structure – the method by which computer components are interrelated.
Computer function – The operation of a component as part of the structure.

For a simple computer there are only four basic functions: data processing, data storage, data movement, or control.
Likewise there are only four basic structures: CPU, main memory, I/O, and System interconnection (busses).

The top level structure of a simple computer is illustrated in Figure 1.1. In general, until about 10 years ago most computers had only one CPU. More recently there has been a trend to include multiple CPUs into a single computer.

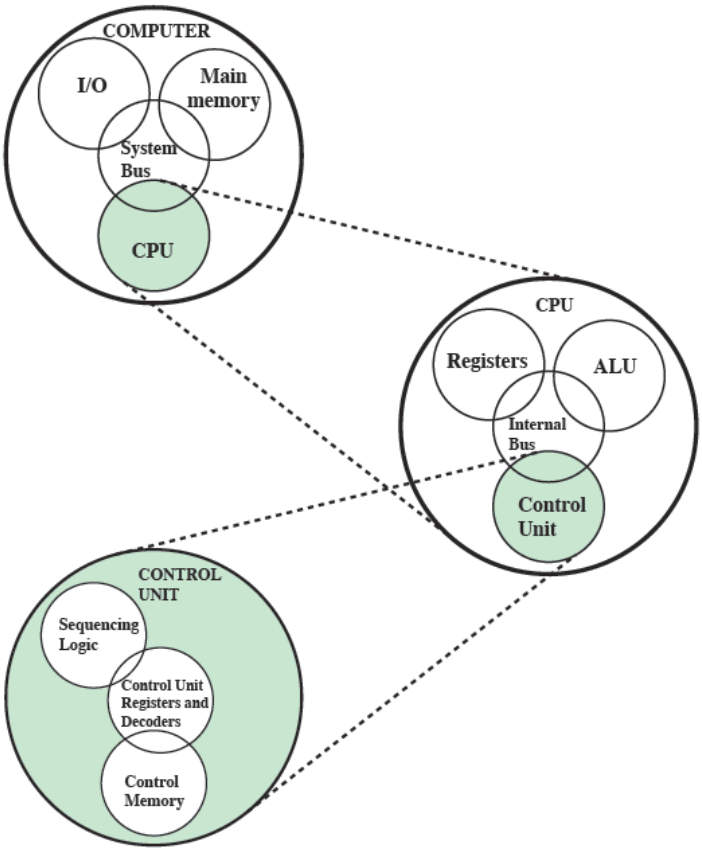

Figure 1.1 A Top-Down View of a Computer

We will accept the following definitions:
Central Processing Unit – CPU Does the fetching and execution of instructions. It has an ALU, a control unit, and registers plus communications paths between them.
Core – functionally equivalent to a CPU on a single CPU system.
Processor – A physical piece of silicon containing one or more cores.

4. What is an ALU – **combinational logic which performs fundamental operations on data such as ADD, SUB, AND, OR, SHIFT, etc. The ALU has no memory logic and is not capable of multiple operations. It typically has two inputs and one output and a mechanism allowing the choice of which operation it performs.**

5. In Figure 1.1 where would cache memory be located? – C**ache memory goes between the CPU and the main memory. It is often located on the CPU itself but in some cases it is part of the main memory system.**

Figure 1.2 shows how a multi-core machine goes together.
6. How do multiple cores make things faster? **Each core can execute a separate thread of an application so this allows things to run in parallel.**

7. Can two cores execute the same thread? **They may be able to but this would be awkward since to gain any advantage you need to have them running in parallel so**

the next instruction may have dependencies on the previous instruction and parallel operation may not be possible

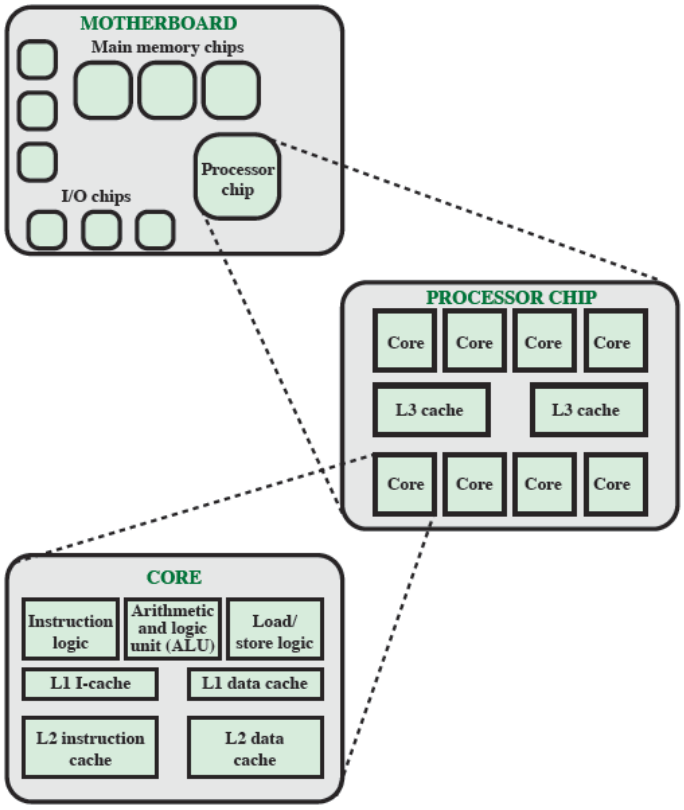

Figure 1.2 Simplified View of Major Elements of a Multicore Computer

Computer History (briefly)
The book does not do a detailed history of computers – it is murky and still provokes arguments. In general, to be the first of anything you need to use a lot of adjectives. There were computer-like devices developed and built in the 1930's most notably by John Attanasoff at Iowa State University and Conrad Zuse in Germany. The ENIAC *Electronic Numerical Integrator and Computer* was completed in 1946 and is often taken the be the first successful all-electronic computer. It was developed by Mauchly and Eckert at the University of Pennsylvania for weapons work. It was not a stored program machine. The stored program concept was first developed by John Von Neuman. Alan Turing developed (independently) the same idea at about the same time in England.

8. What is the *stored program concept?* **A computer program is stored in memory and executed sequentially one instruction at a time after fetching that instruction from memory.**

9. What did computers use before the store program concept? **They were plugged in one wire at a time on a giant plug board to make interconnections.**

Computer history is best viewed as a sequence of developments which we will call *generations.*

First Generation: Vacuum tube computers. Vacuum tubes were used for both logic and for memory. The EDVAC *Electronic Data Variable Computer* was first proposed as a stored program machine in 1945 by von Neuman. In 1946 a new machine called the IAS Computer (Instituted of Advanced Studies) at Princeton was proposed. This was completed in 1952 and is the prototype for almost all subsequent computers. The structure of the IAS machine is shown in Figure 1.6

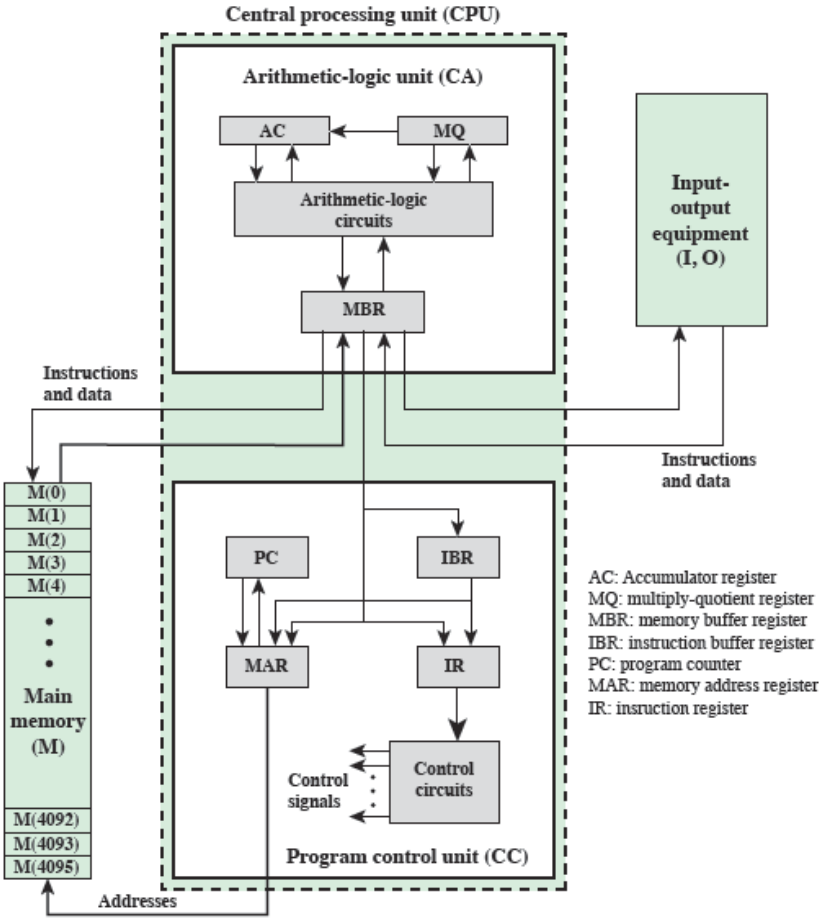

Figure 1.6 IAS Structure

10. What is the purpose of the PC? MAR?, IBR?, IR?

PC – Program Counter – holds the address of the next instruction in memory.
MAR – Memory Address Register – holds the memory address
IBR – Instruction Buffer Register – holds the instruction when it returns from memory
IR – Instruction Register – holds the instruction that is being decoded and executed.

11. Each word in the IAS was 40 bits long. How long was an address which addressed memory? **There are 4096 words in memory so we need 12-bits for an address.**
12. The opcode for each instruction was fixed at 8-bits. How many instructions can we have? **256.**
Table 1.1 gives the IAS instruction set.

Table 1.1 The IAS Instruction Set

Instruction Type	Opcode	Symbolic Representation	Description
Data transfer	00001010	LOAD MQ	Transfer contents of register MQ to the accumulator AC
	00001001	LOAD MQ,M(X)	Transfer contents of memory location X to MQ
	00100001	STOR M(X)	Transfer contents of accumulator to memory location X
	00000001	LOAD M(X)	Transfer M(X) to the accumulator
	00000010	LOAD –M(X)	Transfer –M(X) to the accumulator
	00000011	LOAD |M(X)|	Transfer absolute value of M(X) to the accumulator
	00000100	LOAD –|M(X)|	Transfer –|M(X)| to the accumulator
Unconditional branch	00001101	JUMP M(X,0:19)	Take next instruction from left half of M(X)
	00001110	JUMP M(X,20:39)	Take next instruction from right half of M(X)
Conditional branch	00001111	JUMP+ M(X,0:19)	If number in the accumulator is nonnegative, take next instruction from left half of M(X)
	00010000	JUMP+ M(X,20:39)	If number in the accumulator is nonnegative, take next instruction from right half of M(X)
Arithmetic	00000101	ADD M(X)	Add M(X) to AC; put the result in AC
	00000111	ADD |M(X)|	Add |M(X)| to AC; put the result in AC
	00000110	SUB M(X)	Subtract M(X) from AC; put the result in AC
	00001000	SUB |M(X)|	Subtract |M(X)| from AC; put the remainder in AC
	00001011	MUL M(X)	Multiply M(X) by MQ; put most significant bits of result in AC, put least significant bits in MQ
	00001100	DIV M(X)	Divide AC by M(X); put the quotient in MQ and the remainder in AC
	00010100	LSH	Multiply accumulator by 2; i.e., shift left one bit position
	00010101	RSH	Divide accumulator by 2; i.e., shift right one position
Address modify	00010010	STOR M(X,8:19)	Replace left address field at M(X) by 12 rightmost bits of AC
	00010011	STOR M(X,28:39)	Replace right address field at M(X) by 12 rightmost bits of AC

13. Why are the first three bits of every opcode 000? **They can support up to 256 instructions but they are only using 21 so only a 5-bit opcode was necessary.**
14. How is a conditional branch done? **There is a jump instruction that looks at the contents of the accumulator to see if it is positive or negative (sign bit). The next instruction could be either the right half or the left half of the next word.**

Second Generation Vacuum tubes are replaced by transistors. The transistor was invented in 1947 at Bell Labs but they remained very expensive until the mid to late fifties. Memory cycle time fell from about 30 µsec to 1.4 µsec. Power requirements were greatly reduced as was size. Second generation machines contained on the order of 10,000 transistors.

Third Generation Integrated circuits replace discrete transistors. The I.C. was invented at Texas Instruments about 1958 and by the early sixties they were being widely used commercially. IBM System 360 (1964) and DEC PDP-8 (1964). The 360 system was the first *family* of computers and the PDP-8 was the first mini-computer. Third generations machines had on the order of hundreds of thousands of transistors.

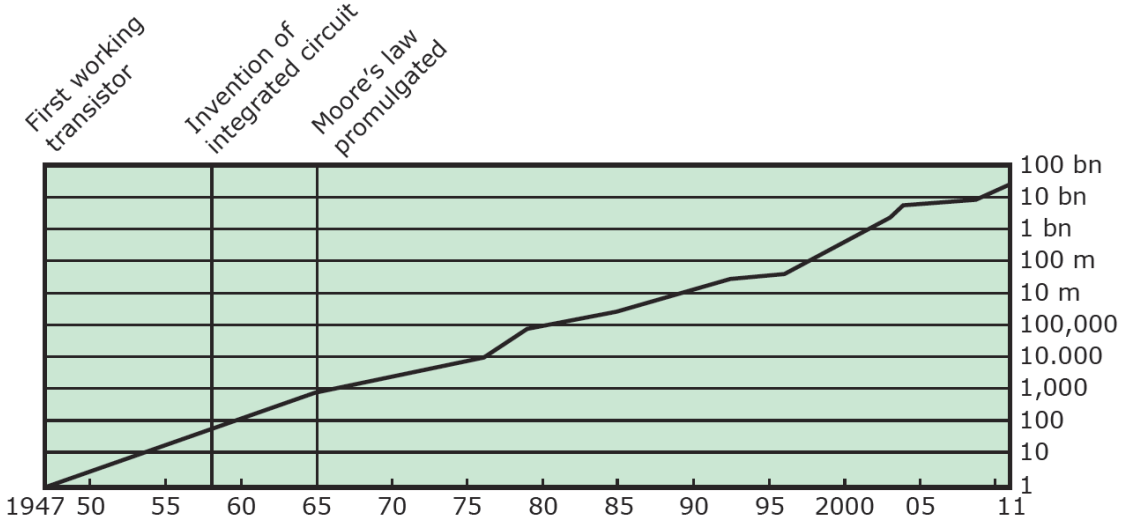

Table 1.2 Computer Generations

Generation	Approximate Dates	Technology	Typical Speed (operations per second)
1	1946–1957	Vacuum tube	40,000
2	1957–1964	Transistor	200,000
3	1965–1971	Small and medium scale integration	1,000,000
4	1972–1977	Large scale integration	10,000,000
5	1978–1991	Very large scale integration	100,000,000
6	1991-	Ultra large scale integration	>1,000,000,000

Memory
From very early on computer's made use of core memory – magnetic core.
Do a short side lecture on this. Access time was about 0.5µsec to read and 1 µsec to write. Cost was about a penny/bit.

Fairchild produced the first "relatively capacious" memory chip. It had 256 bits of memory which comes down to just 32 bytes. It was read only and could be read in 70 nano-seconds.

Microprocessor

Table 1.3 Evolution of Intel Microprocessors (page 1 of 2)

(a) 1970s Processors

	4004	8008	8080	8086	8088
Introduced	1971	1972	1974	1978	1979
Clock speeds	108 kHz	108 kHz	2 MHz	5 MHz, 8 MHz, 10 MHz	5 MHz, 8 MHz
Bus width	4 bits	8 bits	8 bits	16 bits	8 bits
Number of transistors	2,300	3,500	6,000	29,000	29,000
Feature size (μm)	10	8	6	3	6
Addressable memory	640 Bytes	16 KB	64 KB	1 MB	1 MB

(b) 1980s Processors

	80286	386TM DX	386TM SX	486TM DX CPU
Introduced	1982	1985	1988	1989
Clock speeds	6 MHz - 12.5 MHz	16 MHz - 33 MHz	16 MHz - 33 MHz	25 MHz - 50 MHz
Bus width	16 bits	32 bits	16 bits	32 bits
Number of transistors	134,000	275,000	275,000	1.2 million
Feature size (μm)	1.5	1	1	0.8 - 1
Addressable memory	16 MB	4 GB	16 MB	4 GB
Virtual memory	1 GB	64 TB	64 TB	64 TB
Cache	—	—	—	8 kB

Table 1.3 Evolution of Intel Microprocessors (page 2 of 2)

(c) 1990s Processors

	486TM SX	Pentium	Pentium Pro	Pentium II
Introduced	1991	1993	1995	1997
Clock speeds	16 MHz - 33 MHz	60 MHz - 166 MHz,	150 MHz - 200 MHz	200 MHz - 300 MHz
Bus width	32 bits	32 bits	64 bits	64 bits
Number of transistors	1.185 million	3.1 million	5.5 million	7.5 million
Feature size (μm)	1	0.8	0.6	0.35
Addressable memory	4 GB	4 GB	64 GB	64 GB
Virtual memory	64 TB	64 TB	64 TB	64 TB
Cache	8 kB	8 kB	512 kB L1 and 1 MB L2	512 kB L2

(d) Recent Processors

	Pentium III	Pentium 4	Core 2 Duo	Core i7 EE 4960X
Introduced	1999	2000	2006	2013
Clock speeds	450 - 660 MHz	1.3 - 1.8 GHz	1.06 - 1.2 GHz	4 GHz
Bus width	64 bits	64 bits	64 bits	64 bits
Number of transistors	9.5 million	42 million	167 million	1.86 billion
Feature size (nm)	250	180	65	22
Addressable memory	64 GB	64 GB	64 GB	64 GB
Virtual memory	64 TB	64 TB	64 TB	64 TB
Cache	512 kB L2	256 kB L2	2 MB L2	1.5 MB L2/15 MB L3
Number of cores	1	1	2	6

Ch. 1 & 2 pp. 28-74

Evolution of x86 architecture

1. The Intel x86 processor line is regarded as an example of a *Complex Instruction Set Computer (CISC)* while the ARM line of embedded computer are an example of a *Reduced Instruction Set Computer (RISC).* What is the primary difference in these two computer architectures?

CISC	**RISC**
• Large number of complex instructions	• Few instructions
• Relatively few general purpose registers	• Load and Store memory access only
• Not pipelined (too complex)	• Lots of GP registers
• Lots of support for multitasking and memory management.	• pipelined
	• On board cache
• Must be microprogrammed	• Hardwired controller

Refer to the list of x86 processors on the following page.

2. All of the x86 line is said to be made up of *upward compatible* processors. What does this mean? **Next processor in line will easily run the previous processor's code. But the previous processor may not be able to run the code from the later processor.**

3. The 8080 was very widely used and appeared in several hobby machines (Altair, etc.) It was also widely used in industry and in 2000 the Sojourner Truth mars rover had an 8080 processor in the rover. Why was this little 8-bit processor not replaced. **Existing software plus it was available in a radiation hardened version and its characteristics and electronics were very well known and understood.**

4. The 80386 was the first processor capable of multitasking. Why would previous processors not be capable of this? **They had no means to protect memory and isolate processes. This is, in part, a hardware function.**

5. The author mentions that the 8088 is a variant of the 8086 and was first used on the IBM-PC. Why? **The 8088 was an 8086 with an 8-bit interface which made it a 16-bit processor that was cheaper to use. It could also address 1 Mbyte of memory which, at the time – no 8-bit processor could do.**

6. The author notes that the Core i7 EE 4960X has about the same package size as the 8086 yet it runs 800 times faster than an 8086 and it has 64,000 times as many transistors. How is this possible? **Cost is all in the design. Otherwise it's like stamping out records. In addition, the line size used for wires on the chip has shrunk significantly.**

- **8080:** The world's first general-purpose microprocessor. This was an 8-bit machine, with an 8-bit data path to memory. The 8080 was used in the first personal computer, the Altair.
- **8086:** A far more powerful, 16-bit machine. In addition to a wider data path and larger registers, the 8086 sported an instruction cache, or queue, that prefetches a few instructions before they are executed. A variant of this processor, the 8088, was used in IBM's first personal computer, securing the success of Intel. The 8086 is the first appearance of the x86 architecture.
- **80286:** This extension of the 8086 enabled addressing a 16-MB memory instead of just 1 MB.
- **80386:** Intel's first 32-bit machine, and a major overhaul of the product. With a 32-bit architecture, the 80386 rivaled the complexity and power of minicomputers and mainframes introduced just a few years earlier. This was the first Intel processor to support multitasking, meaning it could run multiple programs at the same time.
- **80486:** The 80486 introduced the use of much more sophisticated and powerful cache technology and sophisticated instruction pipelining. The 80486 also offered a built-in math coprocessor, offloading complex math operations from the main CPU.
- **Pentium:** With the Pentium, Intel introduced the use of superscalar techniques, which allow multiple instructions to execute in parallel.
- **Pentium Pro:** The Pentium Pro continued the move into superscalar organization begun with the Pentium, with aggressive use of register renaming, branch prediction, data flow analysis, and speculative execution.
- **Pentium II:** The Pentium II incorporated Intel MMX technology, which is designed specifically to process video, audio, and graphics data efficiently.
- **Pentium III:** The Pentium III incorporates additional floating-point instructions: The Streaming SIMD Extensions (SSE) instruction set extension added 70 new instructions designed to increase performance when exactly the same operations are to be performed on multiple data objects. Typical applications are digital signal processing and graphics processing.
- **Pentium 4:** The Pentium 4 includes additional floating-point and other enhancements for multimedia.[11]
- **Core:** This is the first Intel x86 microprocessor with a dual core, referring to the implementation of two cores on a single chip.
- **Core 2:** The Core 2 extends the Core architecture to 64 bits. The Core 2 Quad provides four cores on a single chip. More recent Core offerings have up to 10 cores per chip. An important addition to the architecture was the Advanced Vector Extensions instruction set that provided a set of 256-bit, and then 512-bit, instructions for efficient processing of vector data.

[11]With the Pentium 4, Intel switched from Roman numerals to Arabic numerals for model numbers.

Embedded Systems

7. What is an *Embedded System*? **A dedicated computer within a product.**

8. Give some examples of embedded systems in your home. **Microwave, dishwasher, oven, furnace, air conditioner, thermostat, cell phone, TV, camera, calculator, printer, car (many), toothbrush, …**

9. How is an embedded processor different from one intended for general use? **Embedded processors have specialized I/O (A/D, CAN bus, etc) and most have some large amount of programmable read only memory.**

10. How is the operating system done in an embedded system? **Many have no operating system. Others piggyback onto an existing operating system such as the Linux system in the Raspberry Pi. There are some operating systems that are specialized for embedded systems. The author mentions TinyOS (used in wireless systems) and there are various types of real time operating systems available as well.**

12. Explain the difference between a microprocessor and a microcontroller.

***Microprocessor* – Typically a general purpose computer CPU on a chip. It may have multiple onboard caches and multiple cores.**

***Microcontroller* – Typically has ROM to hold the program plus a number of more specialized forms of I/O including timers, A/D, Serial ports, parallel ports, and various bus interfaces. It also comes in a wide range of sizes from 1-bit to 32-bits and it operates at a relatively slow speed – a few MHz instead of at GHz. Microcontrollers are made for dedicated applications.**

13. What are *Deeply Embedded Systems?* **The microcontroller in a microwave oven or your furnace maybe examples. They are systems which have a computer designed specifically for one task. For example, a microwave oven controller has a port that handles the keyboard and safety interlocks on the oven.**

ARM Architecture

14. Who builds the ARM processor and what are its origins? **In the early 1980s the BBC provided a grant to develop the Acorn computer as part of a literacy project. As a personal computer the Acorn was mostly a failure. But the Acorn contract allowed the company to develop a RISC machine (Acorn RISC Machine or ARM). VLSI Technology in Silicon Valley built the chips for Acorn which were modestly successful. But Acorn was not big enough to carry on further development so they joined VLSI and Apple to form ARM Ltd in England. The Acorn RISC Machine became the Advanced RISC Machine. ARM now is a multi-national company.**

15. Does ARM build processors? **No. They design processors and license intellectual property to others around the world who build chips.**

16. The main family of processors licensed by ARM is the ARM Cortex family. There are three of these labeled the Cortex-A, Cortex-R, and Cortex-M. What is the purpose of these three processors?

Cortex-A application processors for mobile devices, smartphones, e-book readers, digital TV, DSL, and modems.

Cortex-R real-time applications at high clock rates (200 MHz to 800 MHz). Used in automotive braking systems, disk controllers, etc.

Cortex-M microcontroller applications. Widely used in general purpose microcontroller applications.

17. The textbook takes the Cortex M3 processor as representative of ARM. What are the general characteristics of the Cortex M3?

• 32-bit processor – both instructions and data are 32-bits
• 3-stage pipeline in CPU but no cache
• Lots of Flash memory and significant SRAM
• Has a memory protection unit
• Has support for USB, serial, analog, and other serial busses.
• Has program and data in separate memory with separate busses. (Harvard)
• Has support for clock management, power management, and security.

18. What is *CLOUD* computing? **Model whereby configurable computing resources are made available in a ubiquitous manner by way of an internet connection.**

19. What is *CLOUD* storage? **Makes large amount of memory storage available by way of an internet connection.**

20. What are *CLOUD* services? **Provides software and applications to users via an internet connection.**

Chapter 2

1. The author states that computer power is virtually free and cheaper than toilet paper. What has made this possible?
2. Who was Gordon Moore and what is Moore's Law?
   Ans. Moore was one of the Founders of Intel. Memory chip size will double every 1-1/2 years.
3. Three of the several techniques used to speed up data processing include 1) branch prediction 2) data flow analysis 3) Speculative execution 4) Superscalar execution.
   A) What is branch prediction?
   B) What is data flow analysis?
   Ans. Analyze instruct stream to do some out of order operations.
   C) What is speculative execution?
   Ans. Both sides of a branch.
   D) What is superscalar execution?
   Ans. Multiple parallel pipelines allow multiple instruction to be executed each cycle.
4. Give an example of a very fast and a very slow I/O device. Ans. Gigabit Ethernet or graphics are high and mouse and keyboard are low

After this question is answered show Figure 2.10 and comment on figure and what it means.

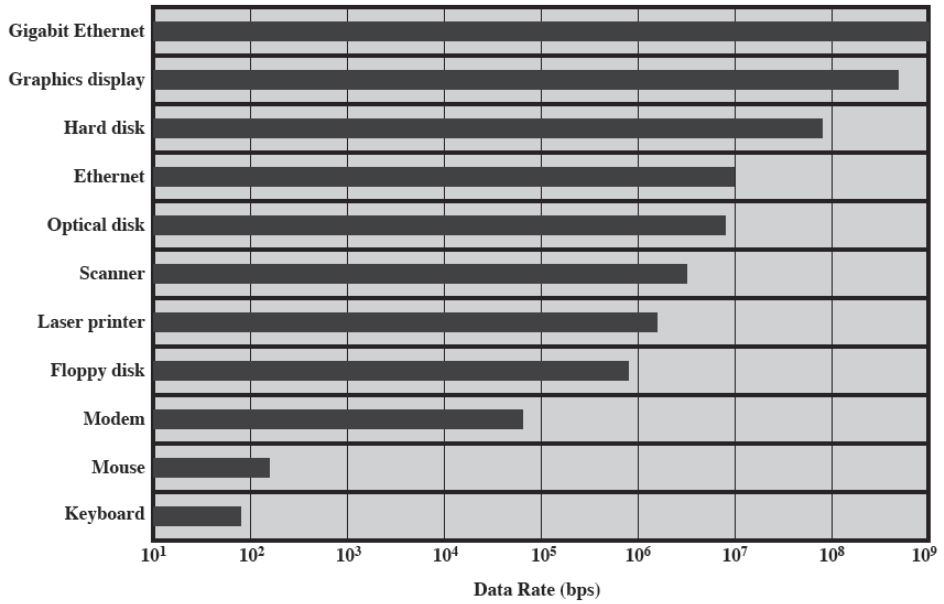

Figure 2.1 p. 49

5. Why is it necessary for the processor and the memory to run at approximately the same speed?
6. Explain the implications of Figure 2.2 p 51. (Graph of logic and memory improvement factor vs time)

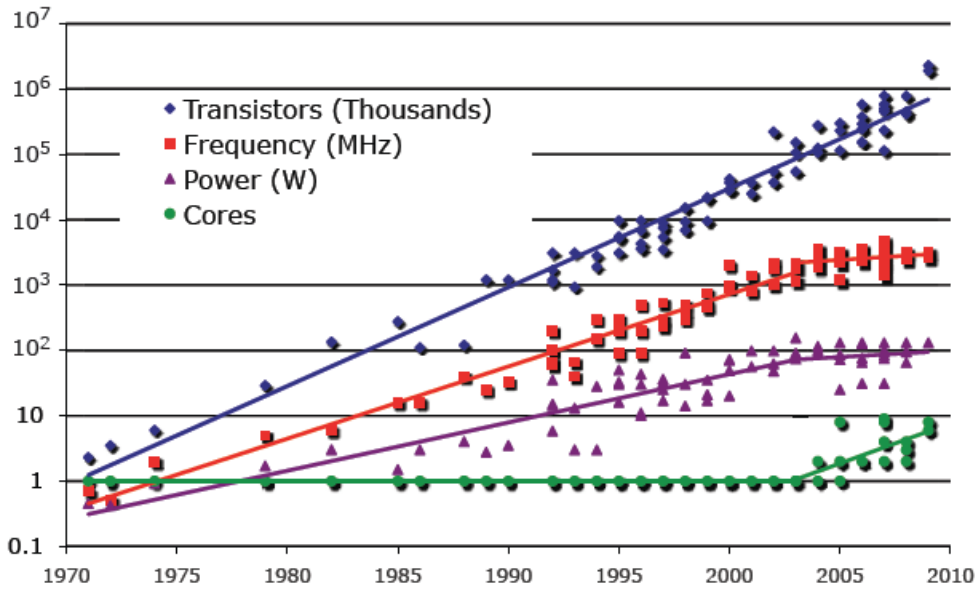

Figure 2.11 Processor Trends

Figure 2.2

6. Why has processor speed risen more quickly than memory speed?
7. What architectural changes can be done to address the disparity between memory speed and processor speed?
   Ans. 1. Increase the chip bit width, 2. cache on RAM, 3. multiple caches, 4. use higher speed bus connections.
8. What is a multicore computer?
   Ans. A computer which has a CPU with multiple processors on the same chip.
9. How many cores per chip are we currently able to produce.
   Ans. More than 50 cores per CPU
10. What is a GPU and how is it different from other processors?
    Ans. Graphics Processor Unit – a core designed to perform parallel operations on graphics data. Many supercomputers are now arrays of GPUs
11. With regard to multiple processors what is Amdahl's Law?
    Ans. Gives the speedup generated by using more processors to solve a problem.

$$\text{Speedup} = \frac{1}{(1-f)+f/N}$$

where f – is the fraction of the code that can be done in parallel and N is the number of processors.

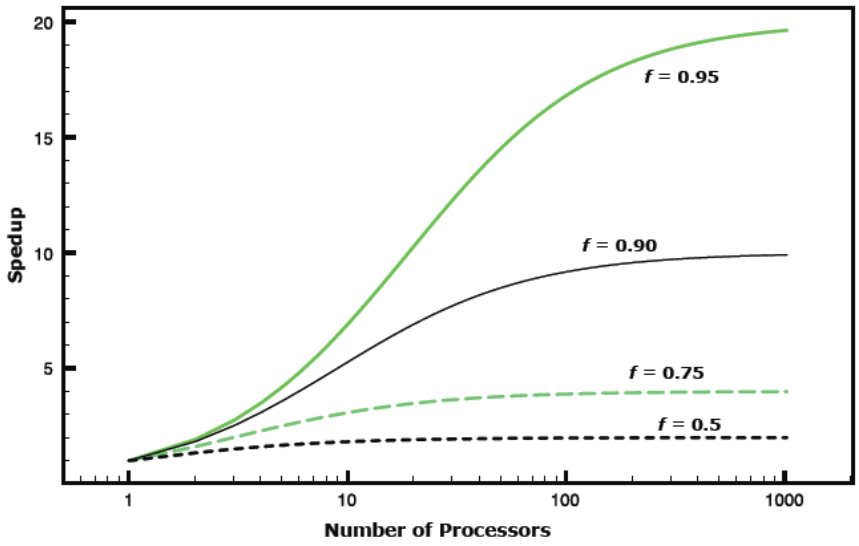

Figure 2.14 Amdahl's Law for Multiprocessors

Figure 2.4

12. In Amdahl's Law what happens as N goes to infinity. **The speedup is inherently limited by the fraction of code which must be run sequentially.**
13. What is Little's Law? **The average number of items in a queuing system equals the average rate at which items arrive multiplied by the average time that an item spends in the system. L = λW where L is the average number of units in the system, λ is the average rate per unit of time, and W is number of units of time the units stay in the system.**
14. Gives some examples of where Little's Law applies. **A multicore machine handing numerous threads, processes waiting to print, a server queue, …**
15. With regard to instruction execution rate what is the **CPI?**
    Ans. Average Cycles per instruction.
16. With regard to instruction execution rate what is the **MIPS?**
    Ans. Millions of Instructions per second executed.
17. With regard to instruction execution rate what is a **FLOP** or a **MFLOP?**
    Ans. Floating Point Operations per Second.
18. What is Benchmark program? **A program that can be run on multiple machines to determine the performance.**
19. Why are benchmark programs written in a HLL rather than in assemble code? **They need to run on multiple machines which have different instruction sets.**