Computer Architecture
Basic Concepts and Evolution
GCR Code:2thnall

Dr. Nausheen Shoaib

Book: Compute Organization and Architecture

Computer Architecture

Introduction

Computer architecture: refers to those attributes of a system visible to a programmer or, those attributes that have a direct impact on the logical execution of a program.

Intruction Set Architecture: defines instruction formats, instruction opcodes, registers, instruction and data memory; the effect of executed instructions on the registers and memory; and an algorithm for controlling instruction execution.

Examples of architectural attributes include the instruction set, the number of bits used to represent various data types (e.g., numbers, characters), I/O mechanisms, and techniques for addressing memory.

Computer organization: refers to the operational units and their interconnections that realize the architectural specifications.

Organizational attributes include those hardware details transparent to the programmer, such as control signals; interfaces between the computer and peripherals; and the memory technology used.

Structure and Function

The hierarchical nature of complex systems is essential to both their design and their description. The designer need only deal with a particular level of the system at a time. At each level, the system consists of a set of components and their interrelationships. The behavior at each level depends only on a simplified, abstracted characterization of the system at the next lower level. At each level, the designer is concerned with structure and function:

Structure: The way in which the components are interrelated.

Function: The operation of each individual component as part of the structure.

There are only four basic functions that a computer can perform:

Data processing: Data may take a wide variety of forms, and the range of processing requirements is broad.

Structure and Function

Data storage: Even if the computer is processing data on the fly (i.e., data come in and get processed, and the results go out immediately), the computer must temporarily store at least those pieces of data that are being worked on at any given moment. There is at least a short-term data storage function. Equally important, the computer performs a long-term data storage function. Files of data are stored on the computer for subsequent retrieval and update.

Data movement: The computer's operating environment consists of devices that serve as either sources or destinations of data. When data are received from or delivered to a device that is directly connected to the computer, the process is known as input–output (I/O), and the device is referred to as a peripheral. When data are moved over longer distances, to or from a remote device, the process is known as data communications.

Control: Within the computer, a control unit manages the computer's resources and orchestrates the performance of its functional parts in response to instructions.

Single Processor Computer Structure

Central processing unit (CPU): Controls the operation of the computer and performs its data processing functions; often simply referred to as processor .

Main memory: Stores data.

I/O: Moves data between the computer and its external environment.

System interconnection: Some mechanism that provides for communication among CPU, main memory, and I/O. A common example of system interconnection is by means of a system bus , consisting of a number of conducting wires to which all the other components attach.

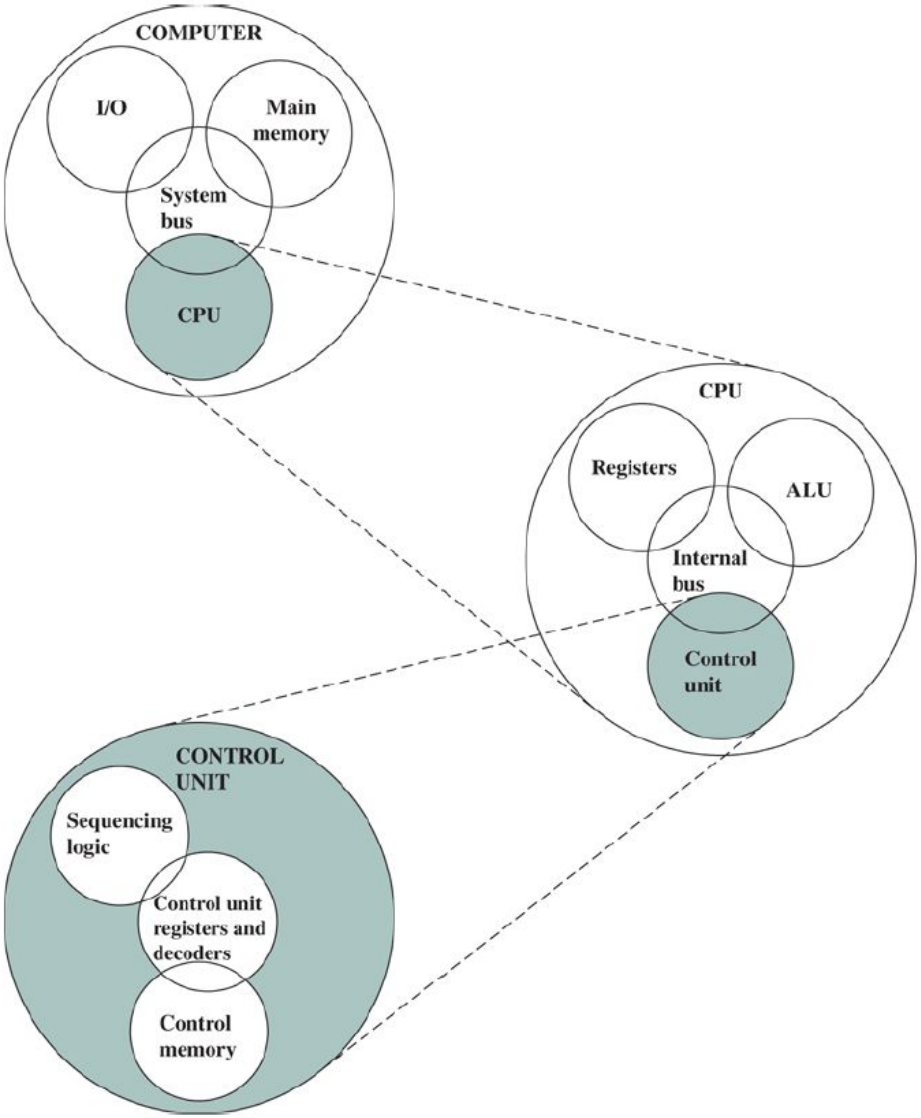


Figure 1.1 The Computer: Top-Level Structure

Single Processor Computer Structure

Control unit: Controls the operation of the CPU and hence the computer.

Arithmetic and logic unit (ALU): Performs the computer's data processing functions.

Registers: Provides storage internal to the CPU.

CPU interconnection: Some mechanism that provides for communication among the control unit, ALU, and registers.

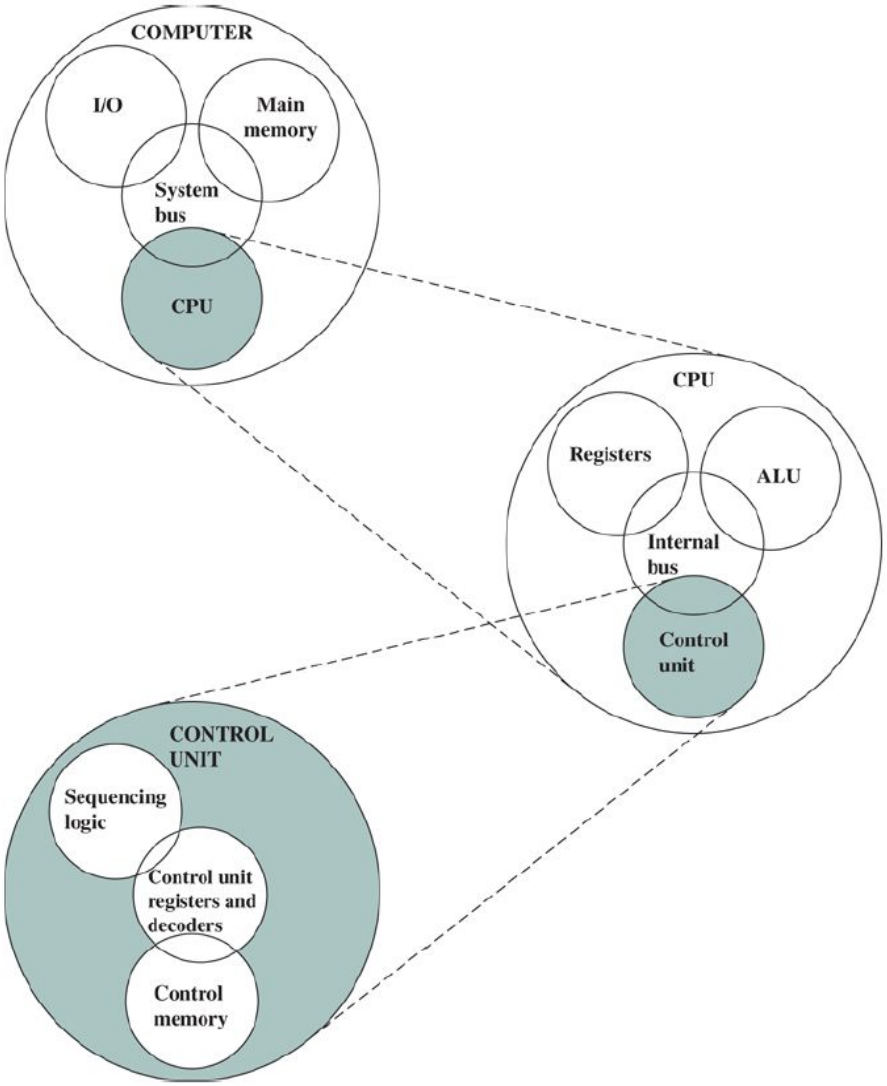


Figure 1.1 The Computer: Top-Level Structure

Multicore Computer Structure

Multicore computer: Each processing unit consisting of a control unit, ALU, registers, and cache is called a core.

Central processing unit (CPU): fetches and executes instructions. It consists of an ALU, a control unit, and registers.

Core: An individual processing unit on a processor chip. A core may be equivalent in functionality to a CPU on a single-CPU system. Other specialized processing units, such as one optimized for vector and matrix operations, are also referred to as cores.

Processor: A physical piece of silicon containing one or more cores. The processor is the computer component that interprets and executes instructions. If a processor contains multiple cores, it is referred to as a multicore processor.

Cache:performance improvement may be obtained by using multiple levels of cache, with level 1 (L1) closest to the core and additional levels (L2, L3, and so on) progressively farther from the core. In this scheme, level n is smaller and faster than level n+1.

Multicore Computer Structure

Figure 1.2 shows principal components of a typical multicore computer.

Most computers, including embedded computers in smartphones and tablets, plus personal computers,laptops, and workstations, are housed on a motherboard.

Printed circuit board (PCB): is a rigid, flat board that holds and interconnects chips and other electronic components. The board is made of layers, typically two to ten, that interconnect components via copper pathways that are etched into the board.

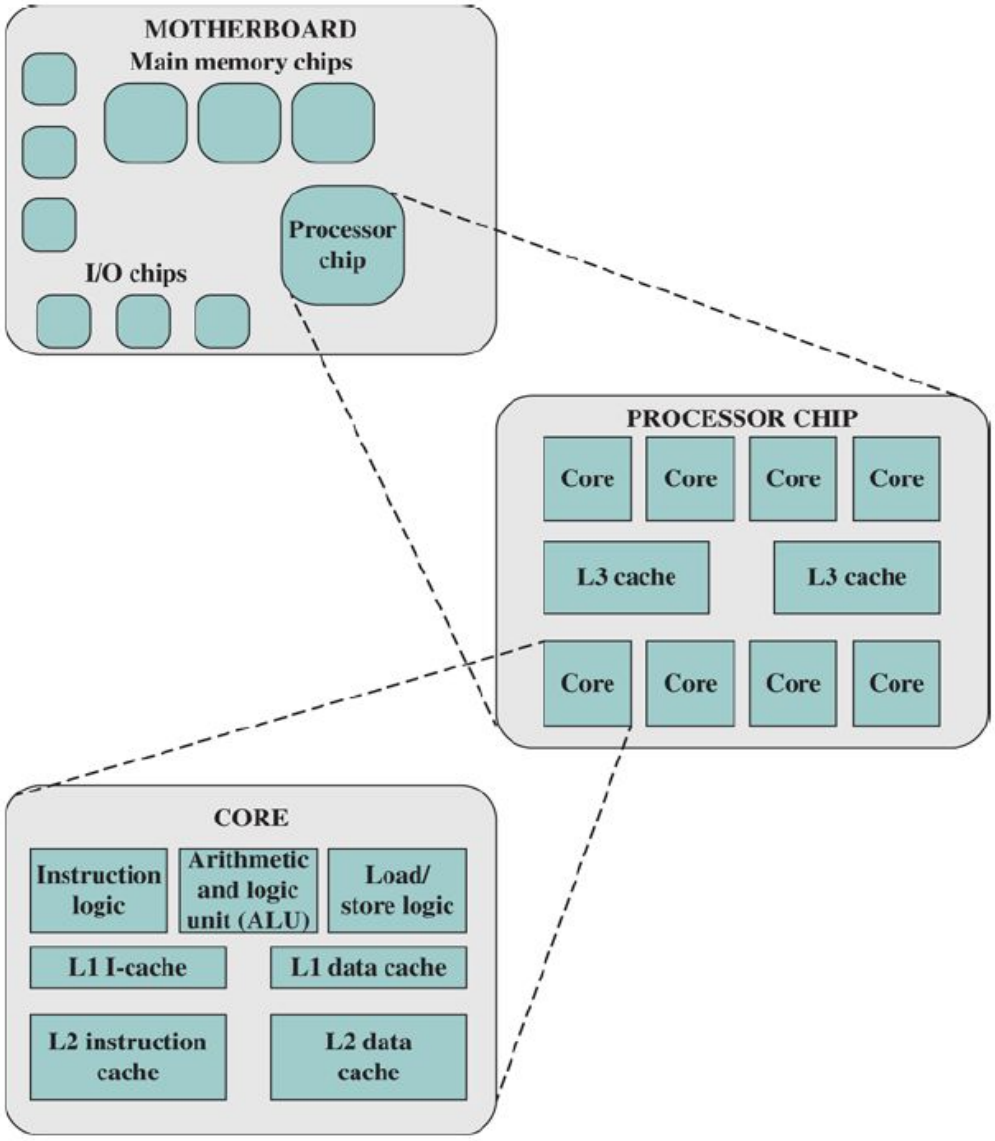


Figure 1.2 Simplified View of Major Elements of a Multicore Computer

Multicore Computer Structure

Motherboard: The printed circuit board in a computer is called a system board or motherboard, while smaller ones that plug into the slots in the main board are called **expansion boards**.

A chip is a single piece of semiconducting material, typically silicon, upon which electronic circuits and logic gates are fabricated, knowns as Integrated Circuit.

The motherboard contains a slot or socket for the processor chip, which typically contains multiple individual cores known as a multicore processor. There are also slots for memory chips, I/O controller chips etc.

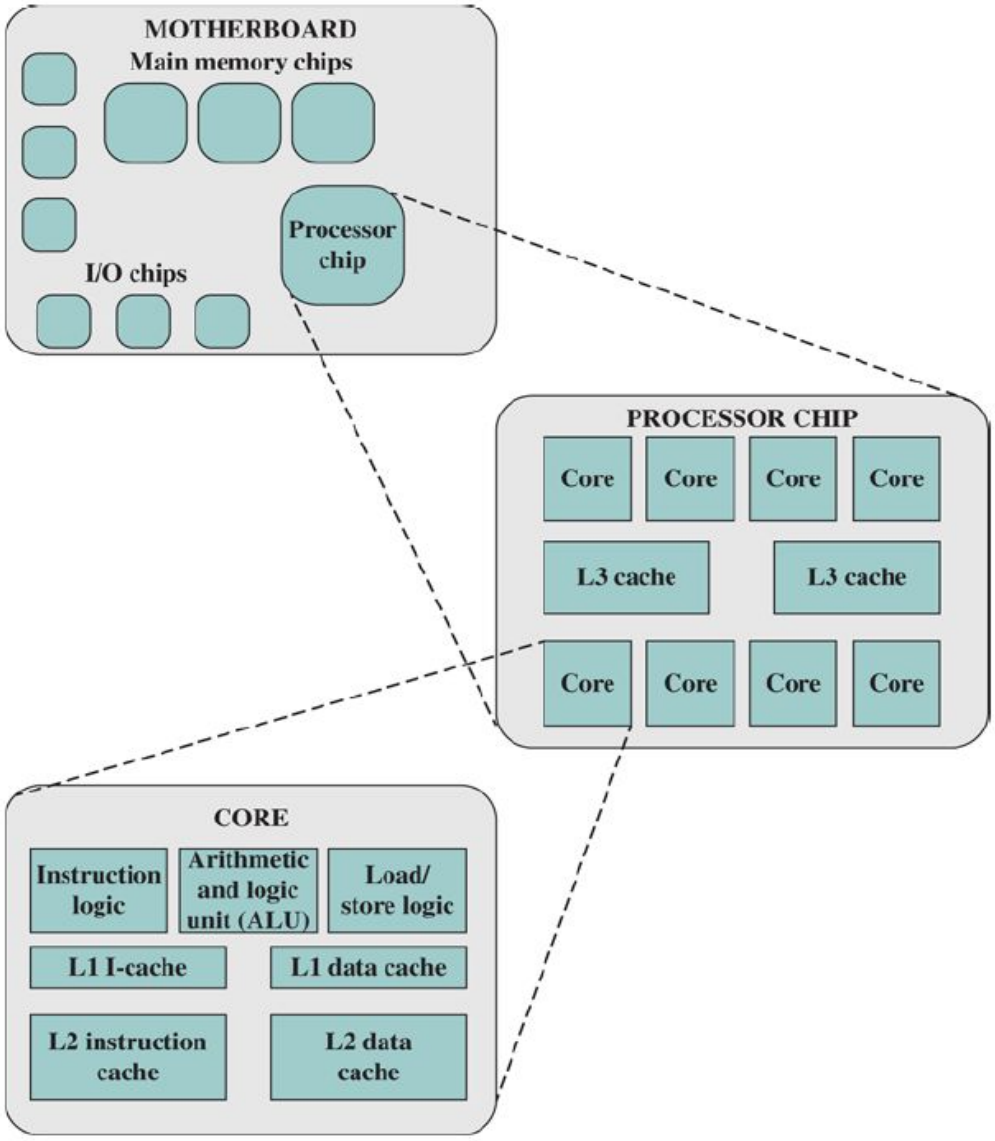


Figure 1.2 Simplified View of Major Elements of a Multicore Computer

Multicore Computer Structure

Figure 1.2 shows a processor chip that contains eight cores and an L3 cache,occupies two distinct portions of the chip surface. All cores have access to the entire L3 cache via the control circuits.

The functional elements of a core are:

Instruction logic: This includes the tasks involved in fetching instructions, and decoding each instruction to determine the instruction operation and the memory locations of any operands.

Arithmetic and logic unit (ALU): Performs the operation specified by an instruction.

Load/store logic: Manages the transfer of data to and from main memory via cache.

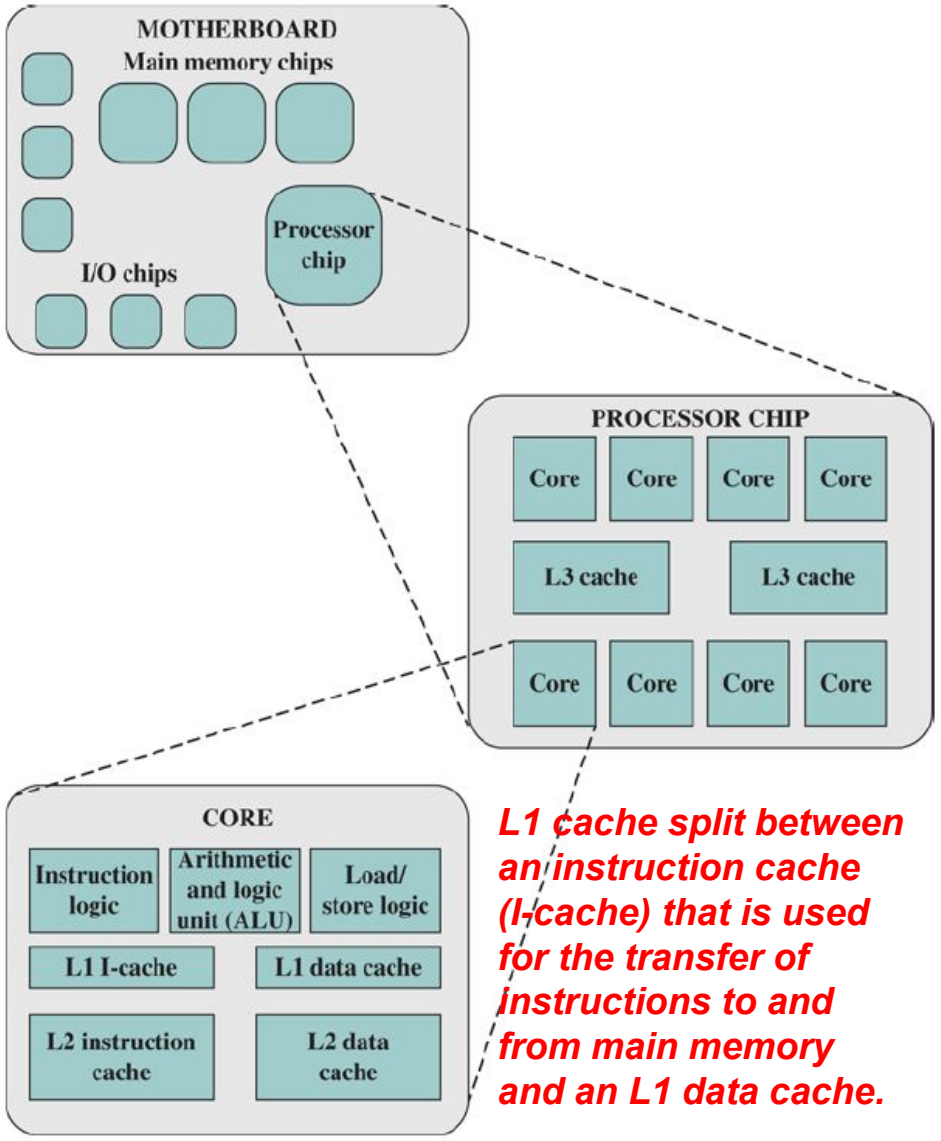


Figure 1.2 Simplified View of Major Elements of a Multicore Computer

Example of Multicore Computer Structure

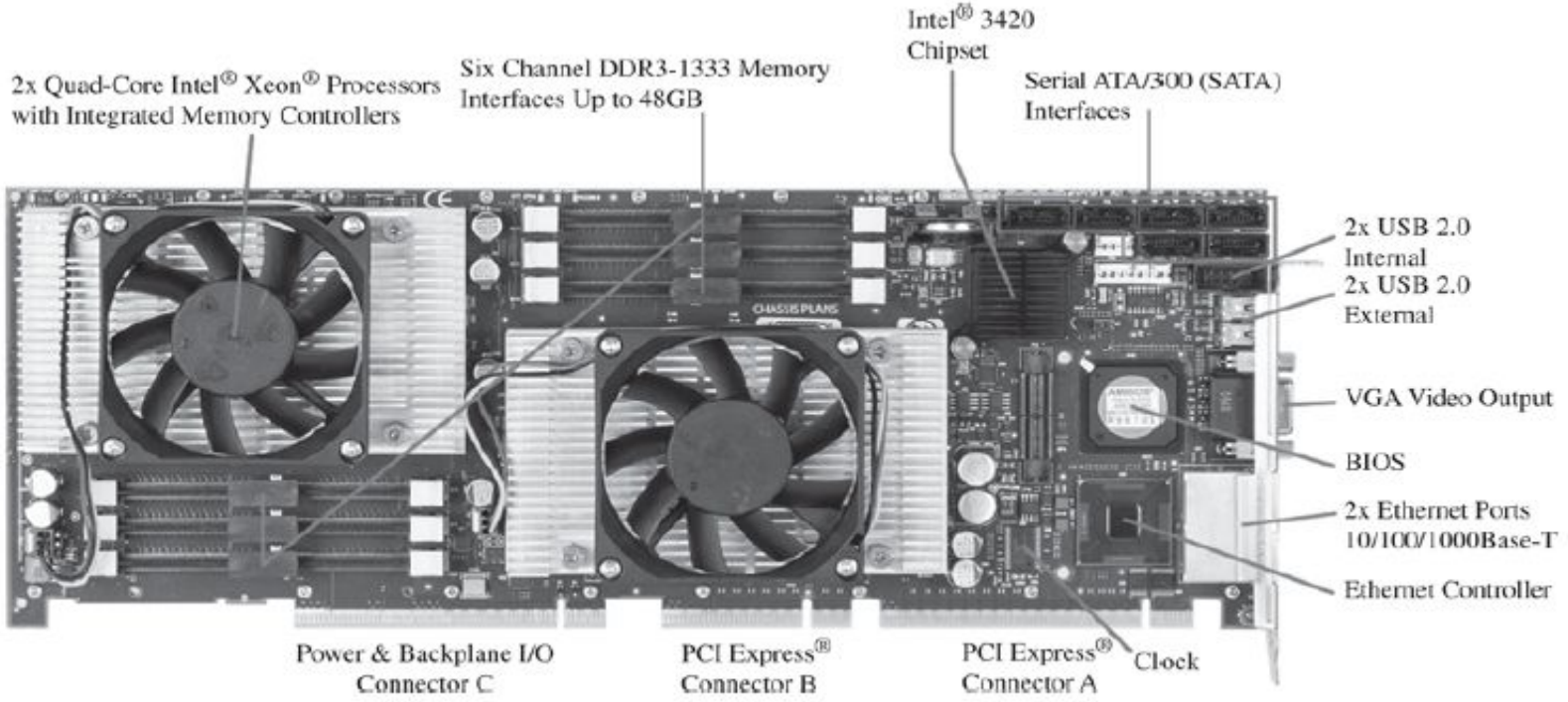


Figure 1.3 Motherboard with Two Intel Quad-Core Xeon Processors

PCI-Express slots for a high-end display adapter and for additional peripherals. **Ethernet controller and Ethernet ports** for network connections. **USB sockets** for peripheral devices. Serial ATA (SATA) sockets for connection to disk memory. **Interfaces for DDR (double data rate)** main memory chips. **Intel 3420 chipset** is an I/O controller for direct memory access operations between peripheral devices and main memory

Example of Multicore Computer Structure

Figure 1.4 is a to-scale layout of the processor chip for the IBM z13 mainframe computer. This chip has 3.99 billion transistors.

Chip has eight cores, or processors.A substantial portion of the chip is devoted to the L3 cache, which is shared by all eight cores. The L3 control logic controls traffic between the L3 cache and the cores and between the L3 cache and the external environment.

Storage control (SC): logic between the cores and the L3 cache.

Memory controller (MC) function: controls access to memory external to the chip. The GX I/O bus controls the interface to the channel adapters accessing the I/O.

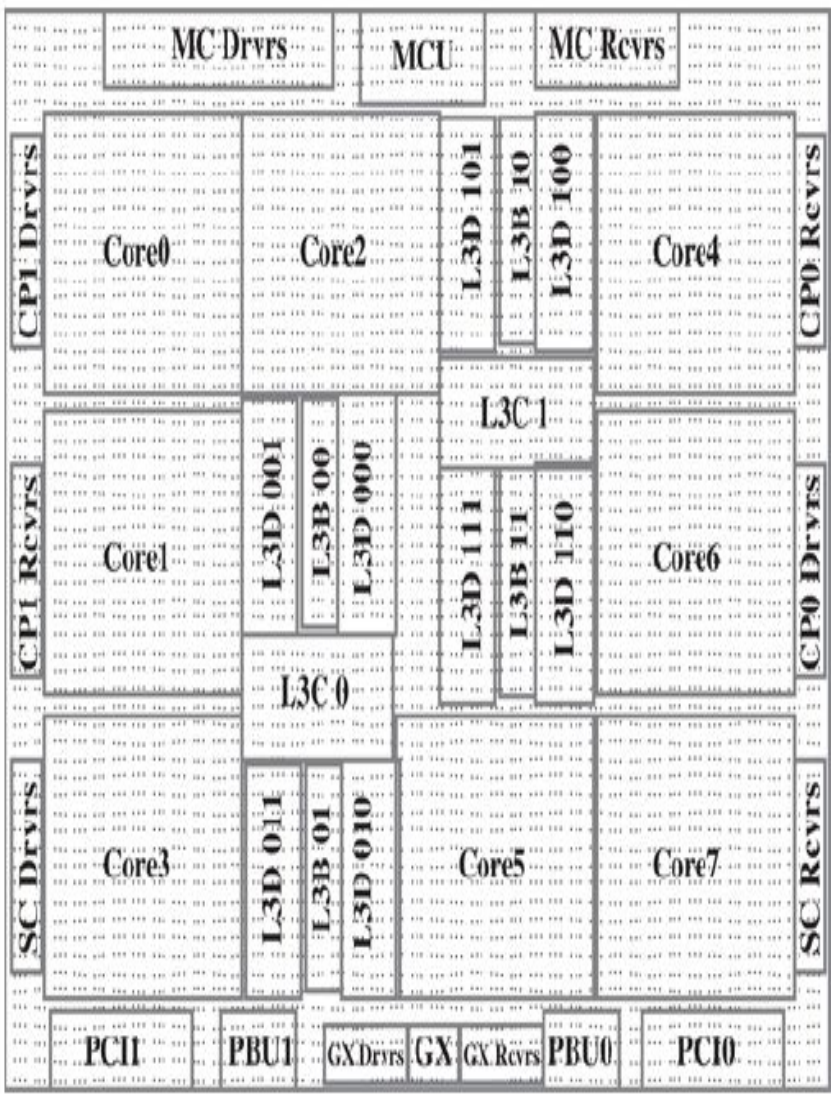


Figure 1.4 IBM z13 Processor Unit (PU) Chip Diagram

Example of Multicore Computer Structure

Figure 1.5 shows z13 instruction set architecture, referred to as the z/Architecture.

ISU (instruction sequence unit): Determines the sequence in which instructions are executed and referred to as a superscalar architecture.

It enables the out-of-order (OOO) pipeline. It tracks register names, OOO instruction dependency, and handling of instruction resource dispatch.

IFB (instruction fetch and branch) and ICM (instruction cache and merge): These two subunits contain the 128-kB instruction cache, branch prediction logic, instruction fetching controls, and buffers. The relative size of these subunits is the result of the elaborate branch prediction design.

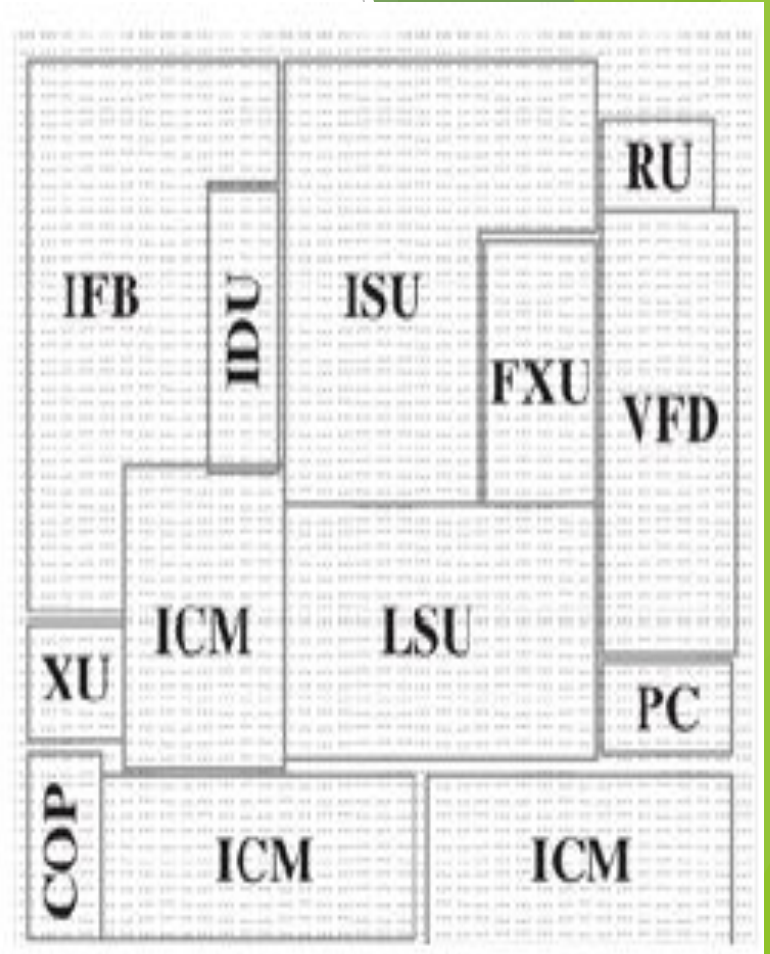


Figure 1.5 IBM z13 Core Layout

equip the processor with multiple processing units to handle several instructions in parallel in each processing stage. several instructions start execution in the same clock cycle and the process is said to use multiple issue, known as 'Superscalar Processors'.

Example of Multicore Computer Structure

IDU (instruction decode unit): IDU is fed from the IFU buffers, and is responsible for the parsing and decoding of all z/Architecture operation codes.

LSU (load-store unit): contains the 96-kB L1 data cache, and manages data traffic between the L2 data cache and the functional execution units. It is responsible for handling all types of operand accesses of all lengths, modes, and formats as defined in the z/Architecture.

XU (translation unit): This unit translates logical addresses from instructions into physical addresses in main memory. The XU also contains a translation lookaside buffer (TLB) used to speed up memory access.

TLB is a memory cache that stores the recent translations of virtual memory to physical memory.

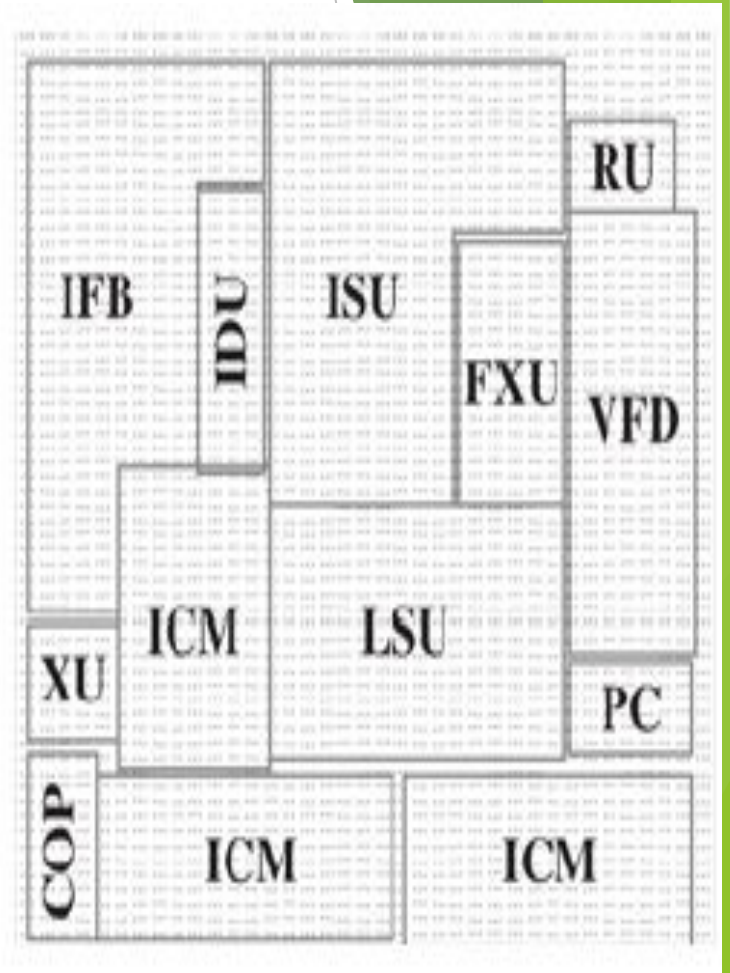


Figure 1.5 IBM z13 Core Layout

Example of Multicore Computer Structure

PC (core pervasive unit): Used for instrumentation and error collection.

FXU (fixed-point unit): executes fixed-point arithmetic operations.

VFU (vector and floating-point units): The binary floating-unit part handles all binary and hexadecimal floating-point operations, as well as fixed-point multiplication operations.

The decimal floating-unit part handles both fixed-point and floating-point operations on numbers that are stored as decimal digits. The vector execution part handles vector operations.

RU (recovery unit): keeps a copy of the complete state of the system that includes all registers, collects hardware fault signals, and manages the hardware recovery actions.

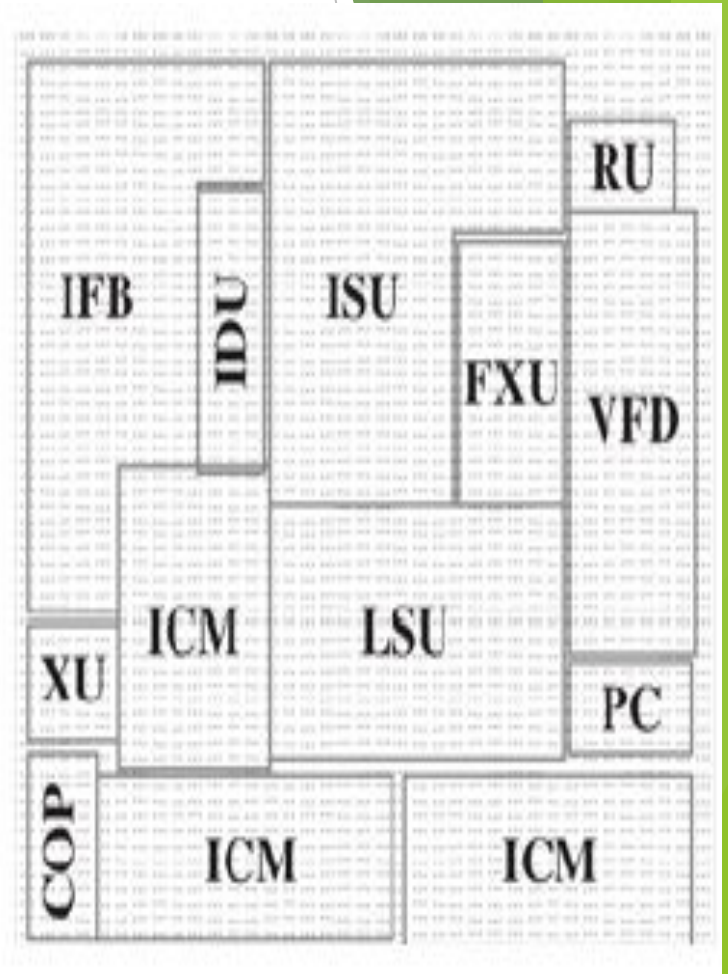


Figure 1.5 IBM z13 Core Layout

Example of Multicore Computer Structure

COP (dedicated co-processor): responsible for data compression and encryption functions for each core.

L2D: A 2-MB L2 data cache for all memory traffic other than instructions.

L2I: A 2-MB L2 instruction cache.

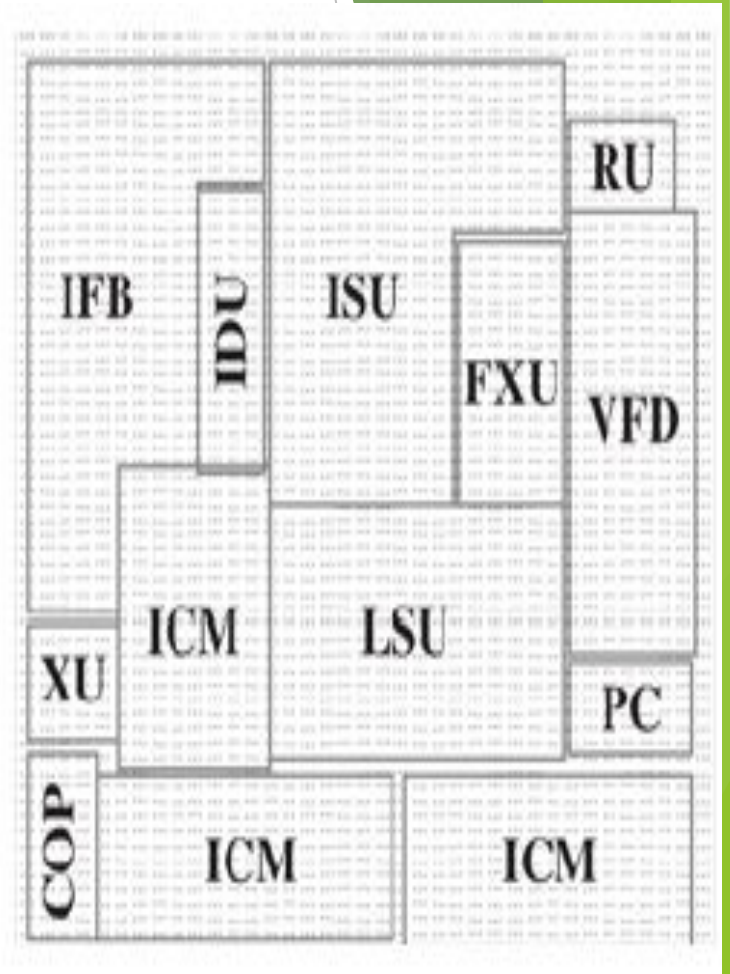


Figure 1.5 IBM z13 Core Layout

The IAS Computer

The first generation of computers used vacuum tubes for digital logic elements and memory.

The most famous first-generation computer, known as IAS computer.

A fundamental design approach first implemented in the IAS computer is known as the stored program concept.

This structure was outlined in von Neumann's, which is worth quoting in parts.

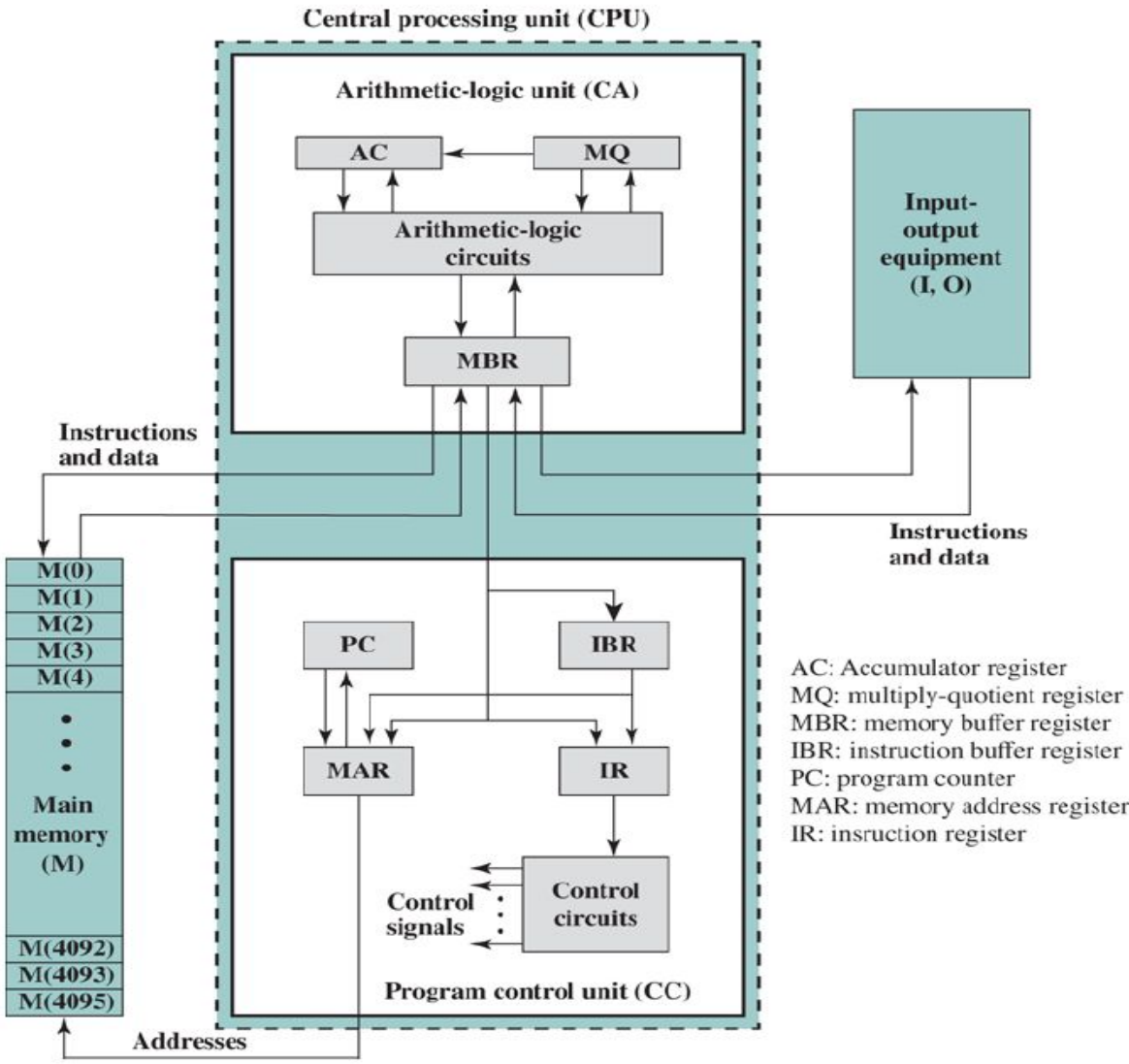


Figure 1.6 IAS Structure

The IAS Computer

First part CA: perform the elementary operations of arithmetic most frequently. These are addition, subtraction, multiplication, and division.

Second part CC: The logical control of the device, that is, the proper sequencing of its operations, can be most efficiently carried out by a central control.

Third part M: carry out long and complicated sequences of operations (specifically of calculations) must have a considerable memory.

Parts CA, CC (together C), and M are input and output of device. Ability to maintain input and output with some medium called the outside recording medium of the device: R

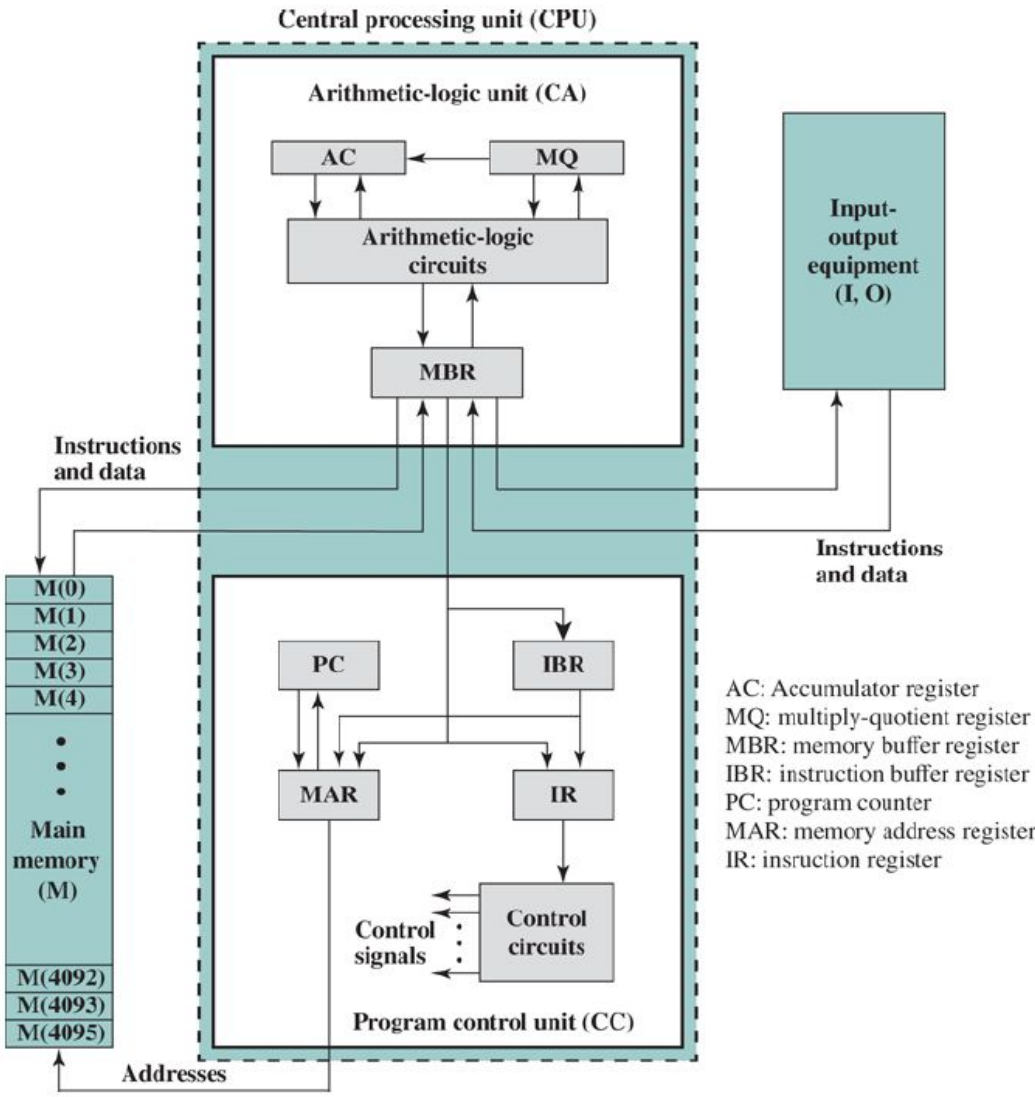


Figure 1.6 IAS Structure

The IAS Computer

Fourth Part I: The device must transfer information from R into its specific parts C and through I.

Fifth Part O: make all transfers from M (by O) into R, and never directly from C.

With rare exceptions, all of today's computers have this same general structure and function and are thus referred to as ***von Neumann machines***.

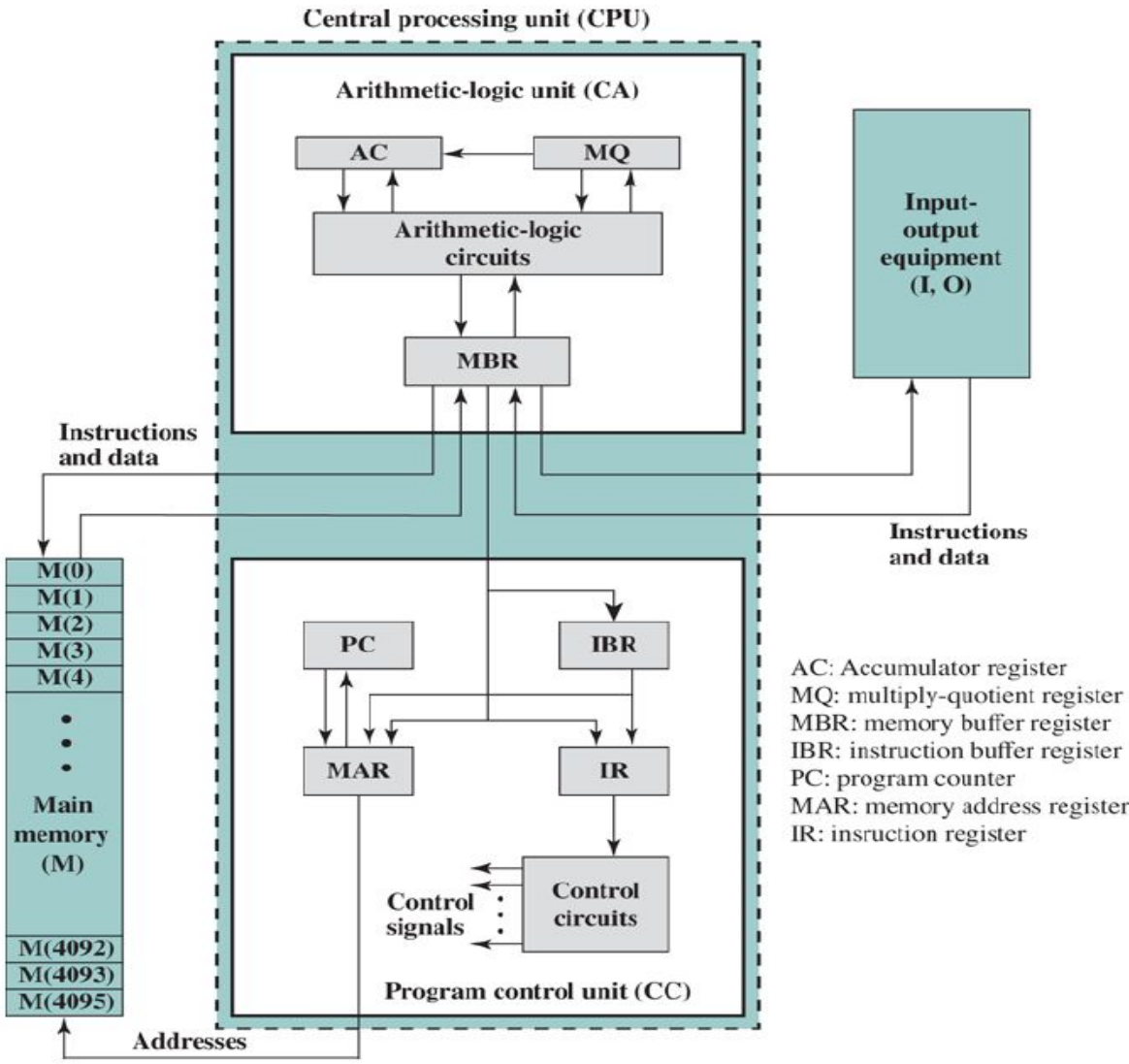


Figure 1.6 IAS Structure

The IAS Computer

Memory buffer register (MBR): Contains a word to be stored in memory or sent to I/O unit, or receive a word from memory or from I/O unit.

Memory address register (MAR): Specifies the address in memory of the word to be written from or read into the MBR.

Instruction register (IR): Contains the 8-bit opcode instruction being executed.

Instruction buffer register (IBR): Employed to hold temporarily the right-hand instruction from a word in memory.

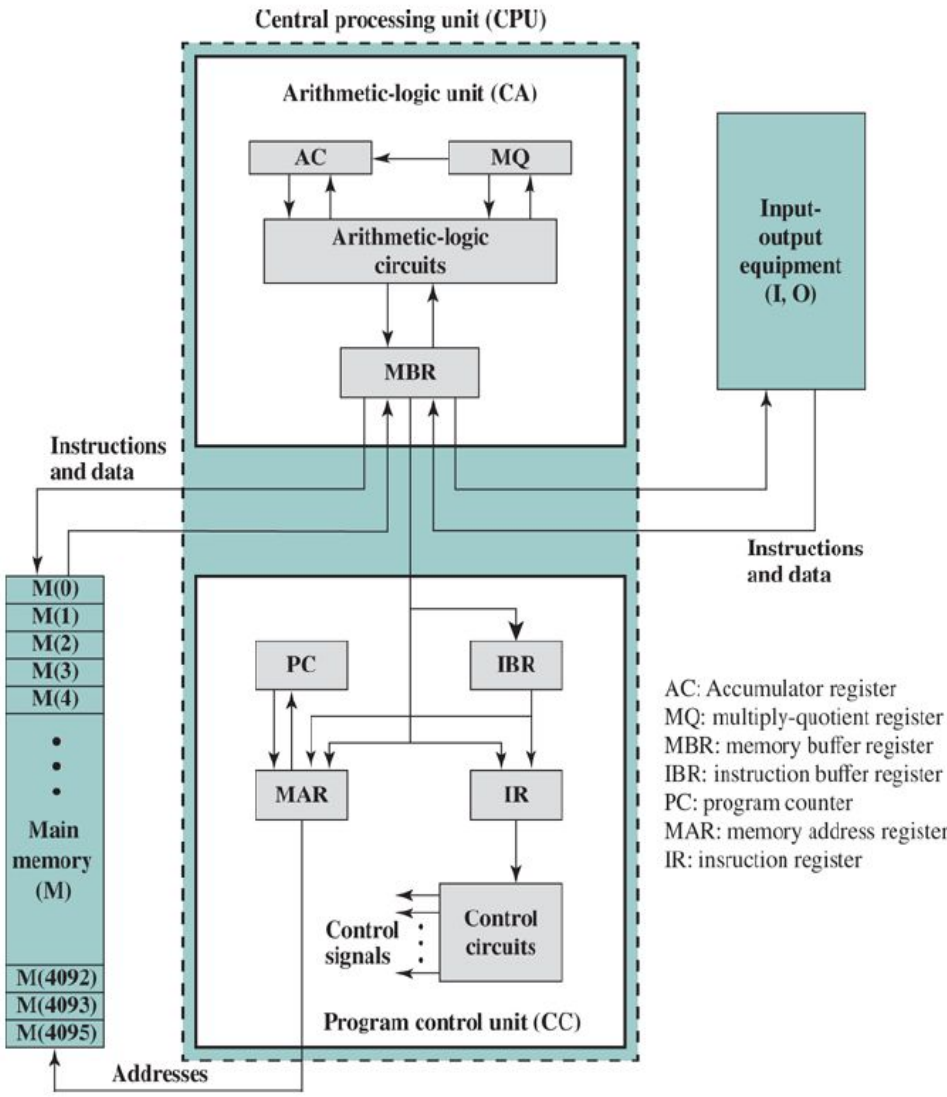


Figure 1.6 IAS Structure

The IAS Computer

Program counter (PC): Contains the address of the next instruction pair to be fetched from memory.

Accumulator (AC) and multiplier quotient (MQ): Employed to hold temporarily operands and results of ALU operations.

For example, the result of multiplying two 40-bit numbers is an 80-bit number; the most significant 40 bits are stored in the AC and the least significant in the MQ.

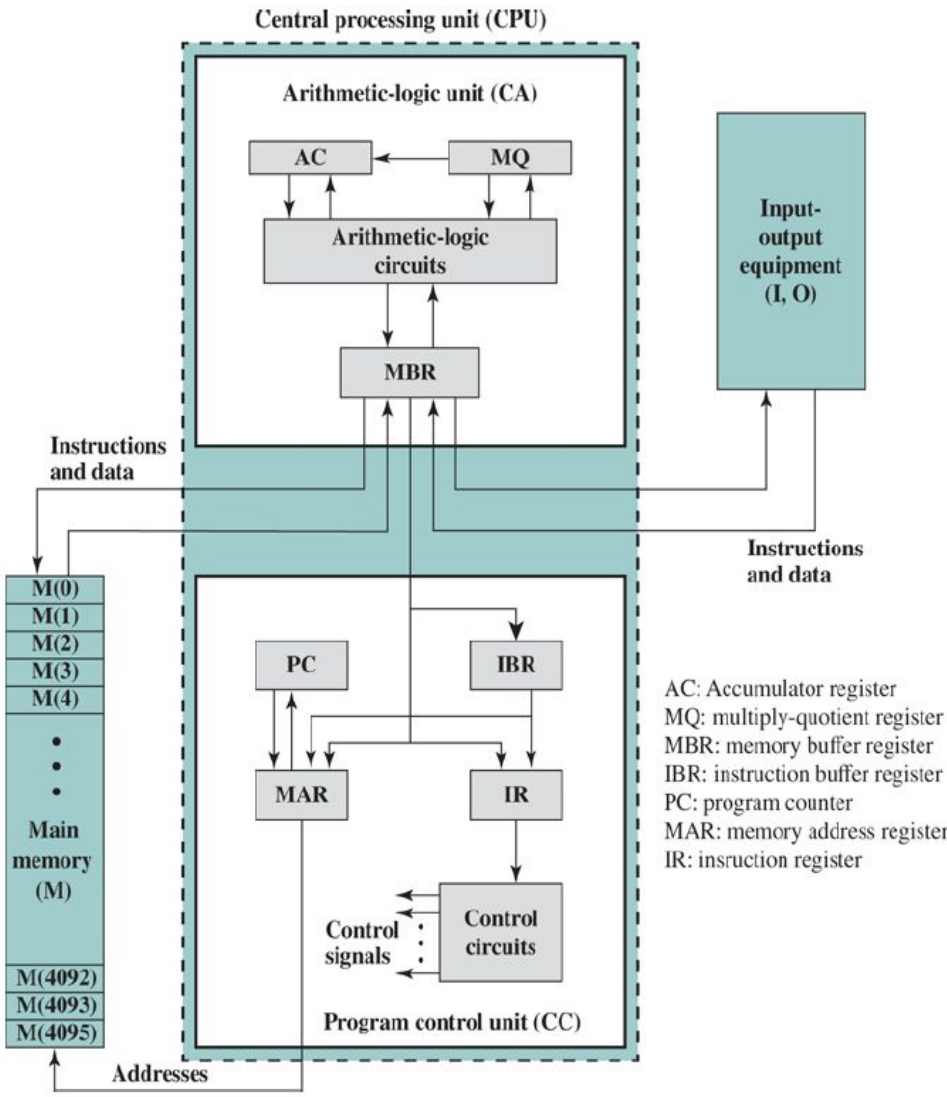


Figure 1.6 IAS Structure

The IAS Computer

The memory of the IAS consists of 4,096 storage locations called words, of 40 binary digits (bits) each. Both data and instructions are stored there. Numbers are represented in binary form and each instruction is a binary code.

Figure 1.7 illustrates these formats. Each number is represented by a sign bit and a 39-bit value. A word may alternatively contain two 20-bit instructions, with each instruction consisting of an 8-bit operation code (opcode) specifying the operation to be performed and a 12-bit address designating one of the words in memory (numbered from 0 to 999).

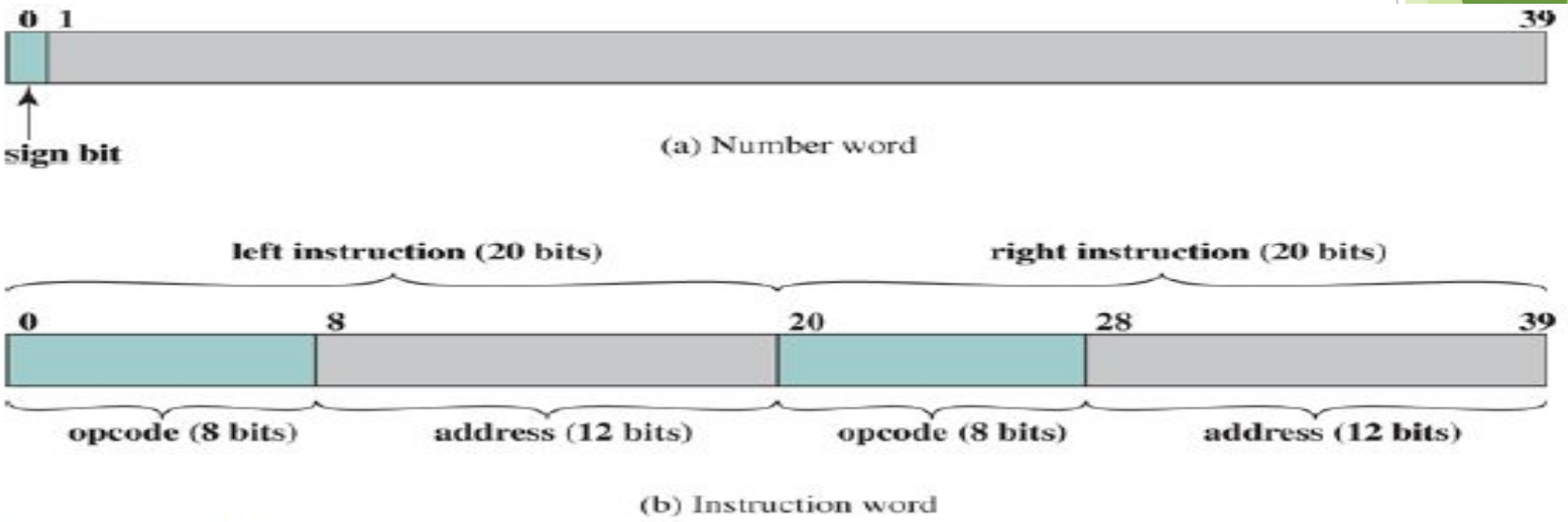


Figure 1.7 IAS Memory Formats

The IAS Computer

Figure 1.7b shows the opcode portion (first 8 bits) specifies which of the 21 instructions is to be executed. The address portion (remaining 12 bits) specifies which of the 4,096 memory locations is to be involved in the execution of the instruction.

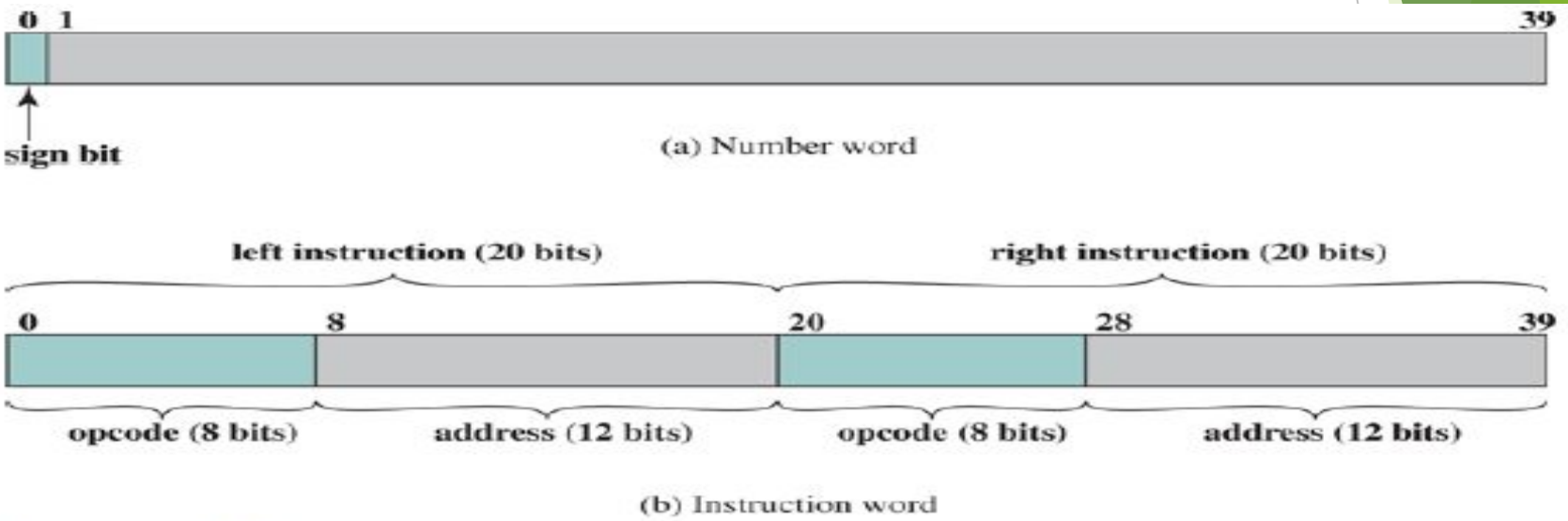


Figure 1.7 IAS Memory Formats

The IAS Computer

The IAS operates by repeatively performing an instruction cycle,shown in Figure 1.8.

Each instruction cycle consists of two subcycles.

During the fetch cycle, the opcode of the next instruction is loaded into the IR and the address portion is loaded into the MAR.

This instruction may be taken from the IBR, or it can be obtained from memory by loading a word into the MBR, and then down to the IBR, IR, and MAR.

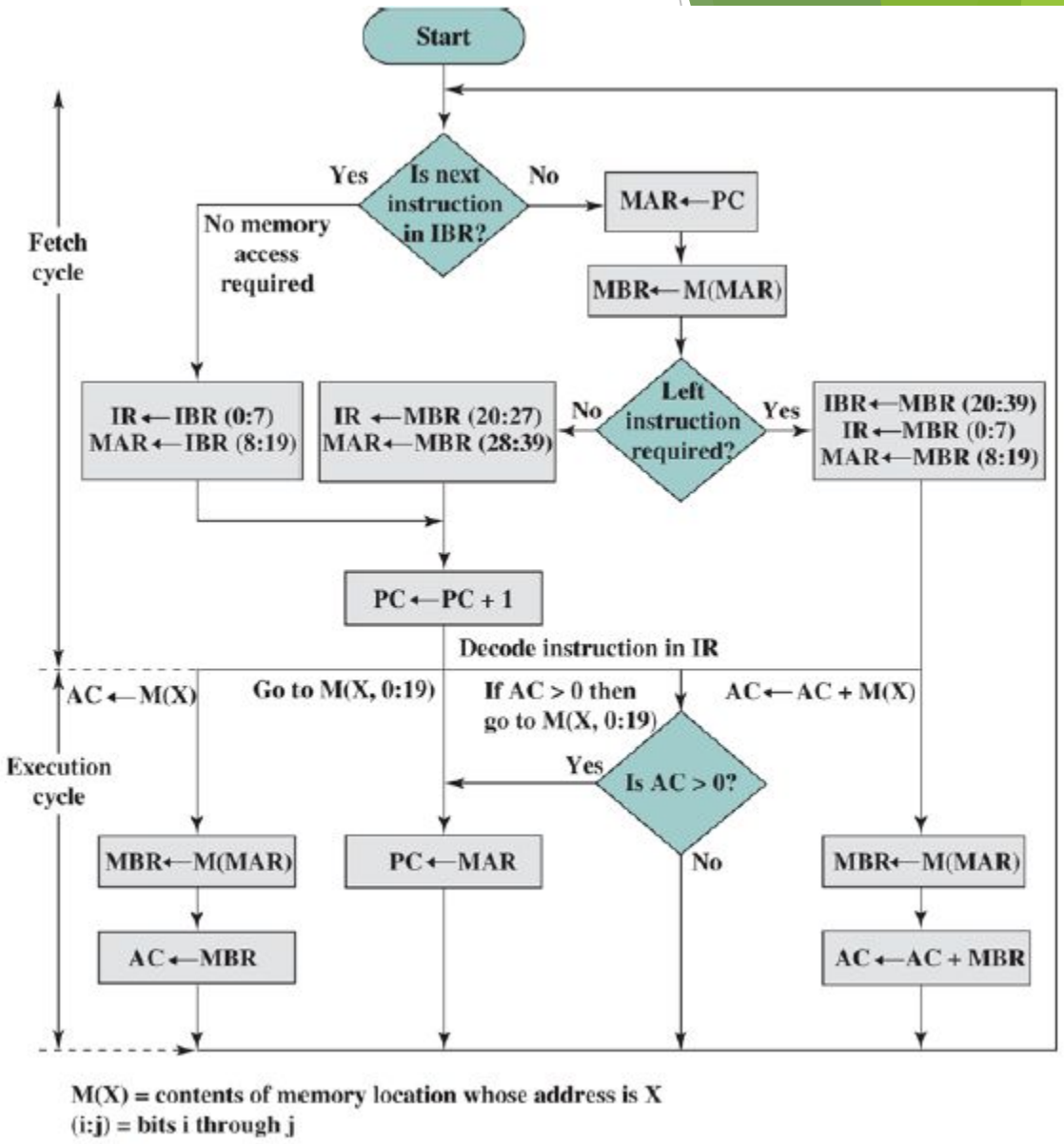


Figure 1.8 Partial Flowchart of IAS Operation

The IAS Computer

Why the indirection? These operations are controlled by electronic circuitry and result in the use of data paths.

To simplify the electronics,There is only one register that is used to specify the address in memory for a read or write and only one register used for the source or destination.

Once the opcode is in the IR, the execute cycle is performed.

Control circuitry interprets the opcode and executes the instruction by sending out the appropriate control signals to cause data to be moved or an operation to be performed by the ALU.

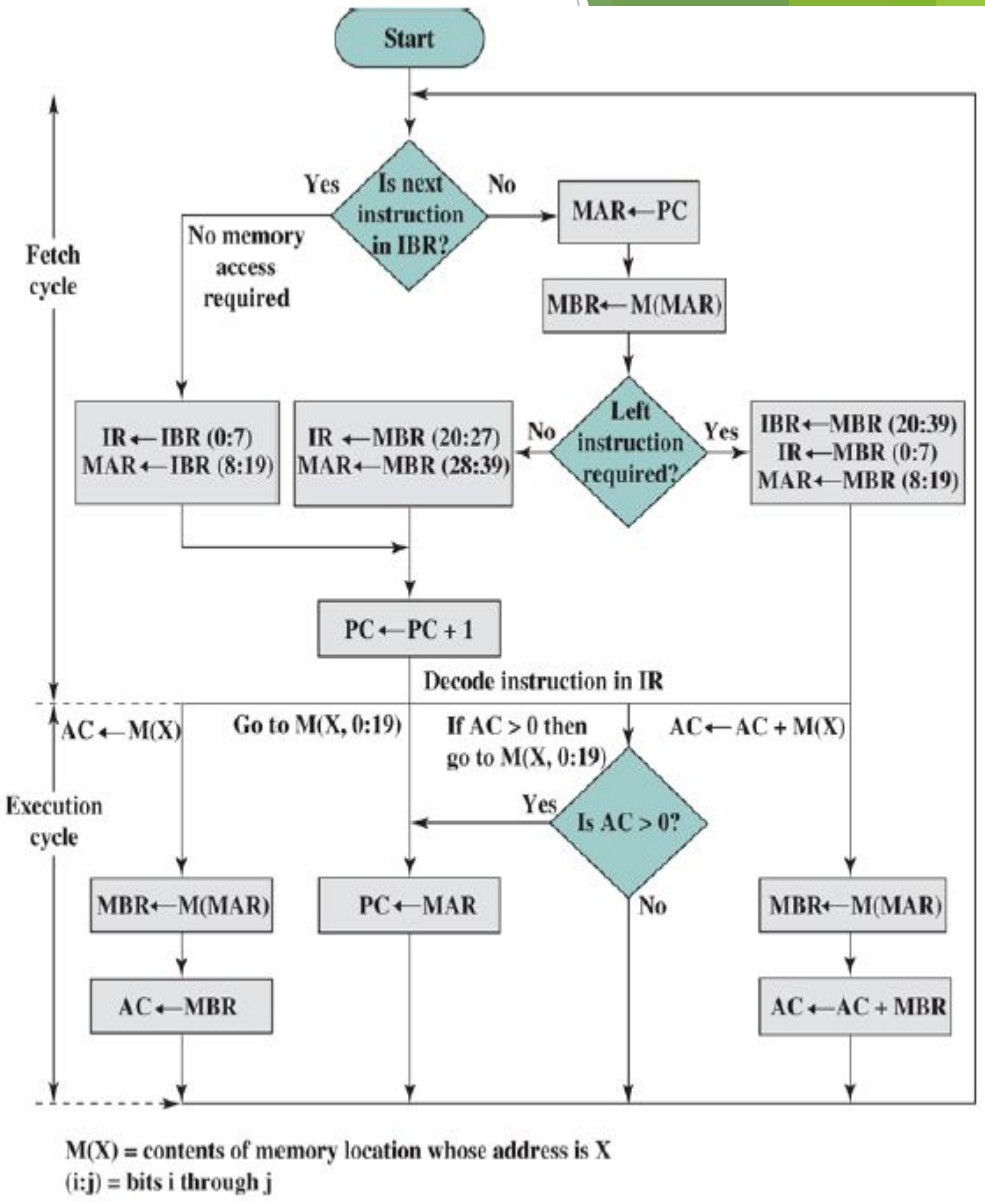


Figure 1.8 Partial Flowchart of IAS Operation

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; that is, shift left one bit position
	00010101	RSH	Divide accumulator by 2; that is, 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

The IAS Computer

Data transfer: Move data between memory and ALU registers or between two ALU registers.

Unconditional branch: Normally, the control unit executes instructions in sequence from memory. This sequence can be changed by a branch instruction, which facilitates repetitive operations.

Conditional branch: The branch can be made dependent on a condition, thus allowing decision points.

Arithmetic: Operations performed by the ALU.

Address modify: Permits addresses to be computed in the ALU and then inserted into instructions stored in memory. This allows a program considerable addressing flexibility.

Gates, Memory Cells, Chips, and Multichip Modules

Gates: A device that implements a simple Boolean or logical function. For example, an AND gate with inputs A and B and output C implements the expression IF A AND B ARE TRUE THEN C IS TRUE. Such devices are called gates because they control data flow in much the same way that canal gates control the flow of water.

Memmory Cells: A device that can store one bit of data; that is, the device can be in one of two stable states at any time. By interconnecting large numbers of these fundamental devices.

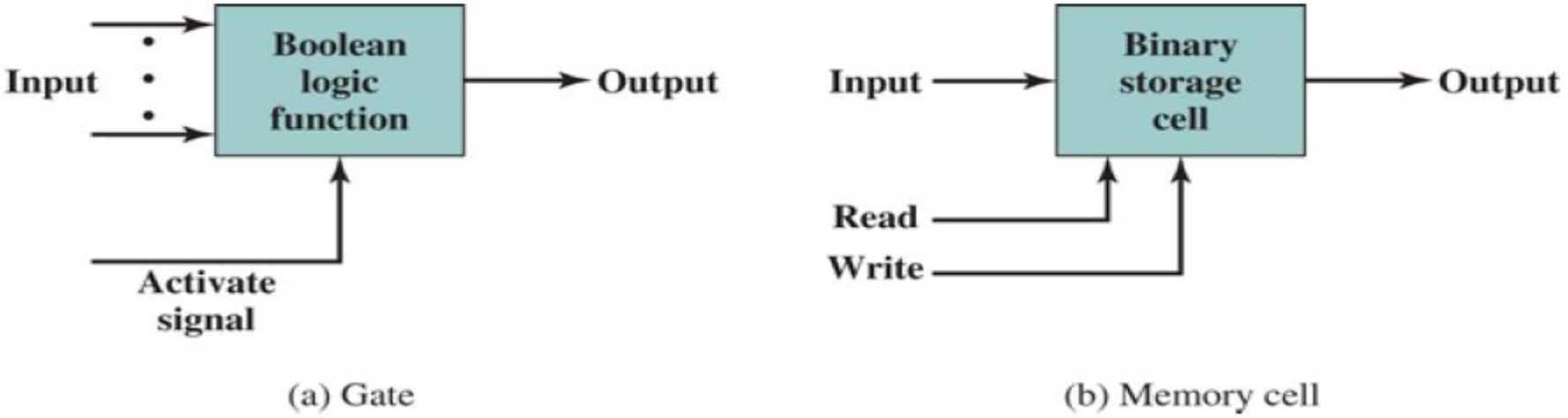


Figure 1.9 Fundamental Computer Elements

Gates, Memory Cells, Chips, and Multichip Modules

We can relate this to our four basic functions as follows:

Data storage: Provided by memory cells.

Data processing: Provided by gates.

Data movement: The paths among components are used to move data from memory to memory and from memory through gates to memory.

Control: The paths among components can carry control signals. For example, a gate will have one or two data inputs plus a control signal input that activates the gate. When the control signal is ON, the gate performs its function on the data inputs and produces a data output.

Memory cell will store the bit that is on its input lead when the WRITE control signal is ON and will place the bit that is in the cell on its output lead when the READ control signal is ON.

Computer consists of gates, memory cells, and interconnections among these elements. The gates and memory cells are constructed of simple electronic components, such as transistors and capacitors.

Gates, Memory Cells, Chips, and Multichip Modules

Transistors: Fundamental building block of digital circuits used to construct processors, memories, and other digital logic devices is the transistor.

Transistor is made of silicon or semiconductor material that can change its electrical state when pulsed. In its normal state, the material may be non-conductive or conductive, either impeding or allowing current flow. When voltage is applied to the gate, the transistor changes its state.

Microelectronic Chips: Integrated circuit exploits the fact that such components as transistors, resistors, and conductors can be fabricated from a semiconductor such as silicon. I

It is merely an extension of the solid-state art to fabricate an entire circuit in a tiny piece of silicon rather than assemble discrete components made from separate pieces of silicon into the same circuit. Many transistors can be produced at the same time on a single wafer of silicon. These transistors can be connected with a process of metallization to form circuits.

Gates, Memory Cells, Chips, and Multichip Modules

Figure 1.10 depicts the key concepts in an integrated circuit.

A thin wafer of silicon is divided into a matrix of small areas, each a few millimeters square. The identical circuit pattern is fabricated in each area, and the wafer is broken up into chips.

Each chip consists of many gates and/or memory cells plus a number of input and output attachment points.

This chip is then packaged in housing that protects it and provides pins for attachment to devices beyond the chip.

A number of these packages can then be interconnected on a printed circuit board to produce larger and more complex circuits.

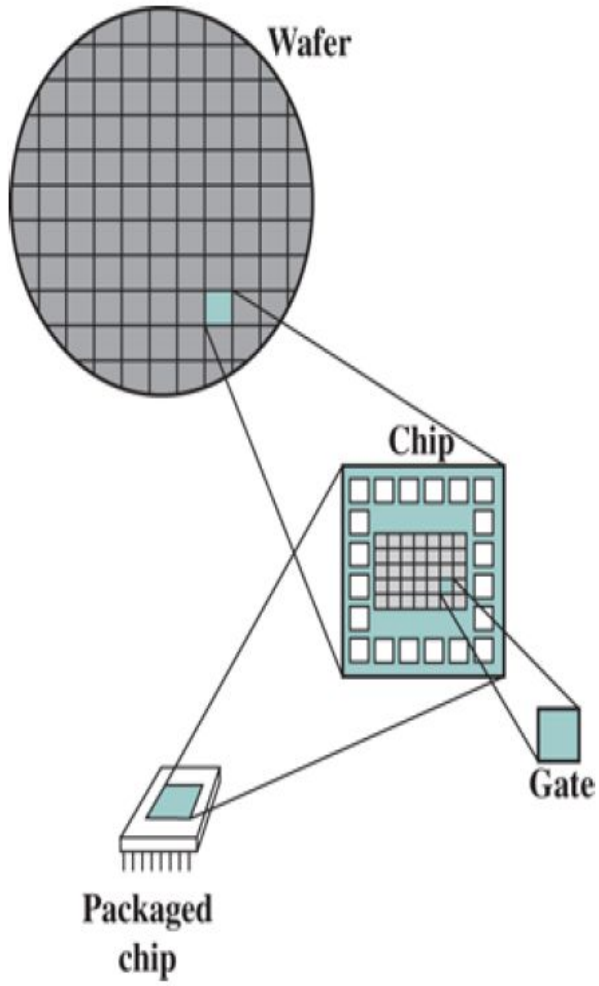
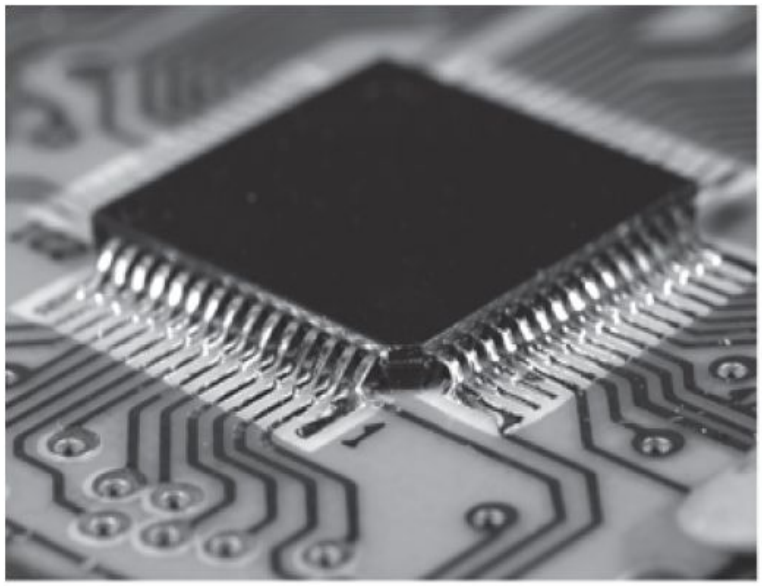


Figure 1.10 Relationship among Wafer, Chip, and Gate

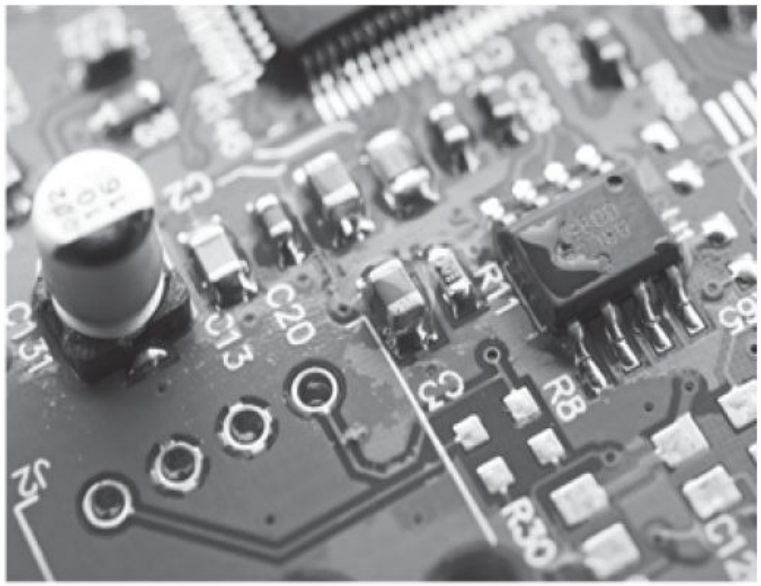
Gates, Memory Cells, Chips, and Multichip Modules

Figure 1.11a indicates a packaged processor or memory chip. Figure 1.11b shows a packaged chip wired onto a motherboard.

Only a few gates or memory cells could be reliably manufactured and packaged together. These early integrated circuits are referred to as **small-scale integration (SSI).** As time went on, it became possible to pack more and more components on the same chip.



(a) Close-up of packaged chip



(b) Chip on motherboard

Figure 1.11 Processor or Memory Chip on Motherboard

Gates, Memory Cells, Chips, and Multichip Modules

Moore's Law: Moore observed that the number of transistors that could be put on a single chip was doubling every year, and correctly predicted that this pace would continue into the near future. To the surprise of many, including Moore, the pace continued year after year and decade after decade.

Figure 1.12 reflects the famous Moore's law.

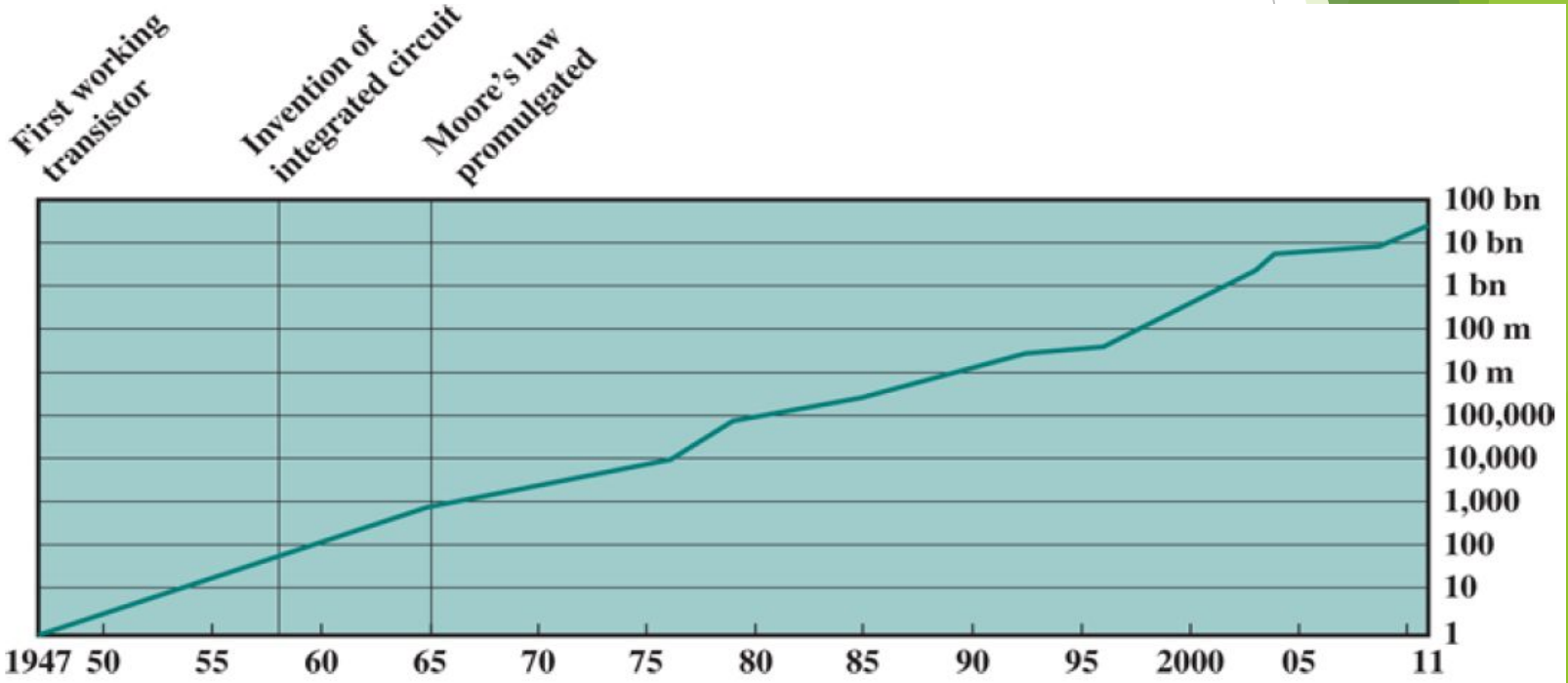


Figure 1.12 Growth in Transistor Count on Integrated Circuits

Gates, Memory Cells, Chips, and Multichip Modules

Consequences of Moore's law:

1. The cost of a chip has remained virtually unchanged during this period of rapid growth in density. This means that the cost of computer logic and memory circuitry has fallen at a dramatic rate.

2. Because logic and memory elements are placed closer together on more densely packed chips, the electrical path length is shortened, increasing operating speed.

3. The computer becomes smaller, making it more convenient to place in a variety of environments.

4. There is a reduction in power requirements.

5. The interconnections on the integrated circuit are much more reliable than solder connections. With more circuitry on each chip, there are fewer interchip connections.

Gates, Memory Cells, Chips, and Multichip Modules

Multichip Module: The basic idea behind developing MCM technology is to decrease the average spacing between ICs in an electronic system.

An MCM is a chip package that contains several bare chips mounted close together on a substrate (base) and interconnected by conductors in that base. The short tracks between the chips increase performance and eliminate much of the noise that external tracks between individual chip packages can pick up.

MCMs are classified by substrate, which include the following types:

MCM-L(Memory Chip Module - Laminated): The substrate is a multi-layer laminated printed circuit board (PCB).

MCM-Ceramics: The substrate is built on ceramic,such as low temperature co-fired ceramic.

MCM-D: The ICs are deposited on the base substrate using Thin Film technology.

Gates, Memory Cells, Chips, and Multichip Modules

The basic architecture of an MCM shown in Figure 1.13:

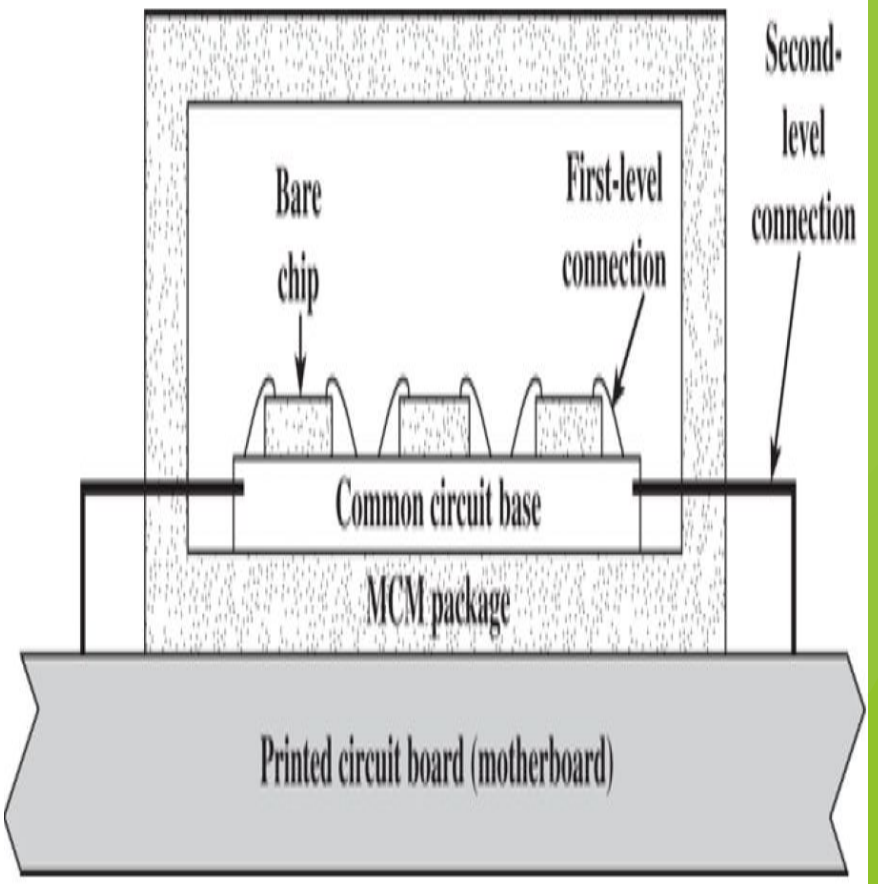
Integrated circuits: Bare chips mounted on/in the surface of the substrate.

Level-1 interconnections: Connections between chips through paths in the substrate.

Substrate: The common base that provides all the signal interconnections and the mechanical support for all chips.

MCM package: Provides a degree of protection to the circuits in addition to heat removal and interconnections.

Level-2 interconnections: Provides the necessary interface to the printed circuit board on which the MCM is mounted.



Evolution of the Intel x86 Architecture

≠Í¼ ð ╠┤ ┼ ¼Ż ╚ ◘ïï ðď'¼┤ ł·ŻĐ ╒ ┼ ¼┘ ┼ ¼ ╒¼Ż ╚ ╚í¼ ┼ ¼ ╒ ╠┤t' ╚ ╒ ðď ¹¼ðÒ¹¼ ╒ ðď ¹¼ ╒ł·ĐŻ ¼ď'ð┤ ╚ ðŻððŹ┘ t'¼◘ ł·Ż ╒ ╚┤ ╠┤ð ╚ł·ðŻ ╒¼ ╚ ððŹ┘ ╠┤╚¼┤ ╒ Ą8b•8 ╒ţø!Ż Òť ╚¼┤ ŻÒ ╚ł·—½ Ò┘ ┘ ┼ ðÒðí ╚ð┘ ┼ ðð¼ ╒ ╒ð┤ ¹¼ ╒ł·ĐŻ ł· ╒ ╚í¼ ┼ ¼¹ ╠┤ð¼¹ ł·Ż ╒ ╚┤ ╠┤ð ╚ł·ðŻ ╒¼ ╚ððŹ┘ ╠┤╚¼┤ Ąüb•8ţø

	(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

Evolution of the Intel x86 Architecture

(b) 1980s Processors				
	80286	386TM DX	386TM SX	486TM DX CPU
Introduced	1982	1985	1988	1989
Clock speeds	6–12.5 MHz	16–33 MHz	16–33 MHz	25–50 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

Evolution of the Intel x86 Architecture

	(c) 1990s Processors			
	486TM SX	Pentium	Pentium Pro	Pentium II
Introduced	1991	1993	1995	1997
Clock speeds	16–33 MHz	60–166 MHz,	150–200 MHz	200–300 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

Evolution of the Intel x86 Architecture

	(d) Recent Processors				
	Pentium III	Pentium 4	Core 2 Duo	Core i7 EE 4960X	Core i9-7900X
Introduced	1999	2000	2006	2013	2017
Clock speeds	450–660 MHz	1.3–1.8 GHz	1.06–1.2 GHz	4 GHz	4.3 GHz
Bus width	64 bits	64 bits	64 bits	64 bits	64 bits
Number of transistors	9.5 million	42 million	167 million	1.86 billion	7.2 billion
Feature size (nm)	250	180	65	22	14
Addressable memory	64 GB	64 GB	64 GB	64 GB	128 GB
Virtual memory	64 TB	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	14 MB L3
Number of cores	1	1	2	6	10

Evolution of the Intel x86 Architecture

some of the highlights of the evolution of the Intel product line:

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.

Evolution of the Intel x86 Architecture

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: 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: incorporated Intel MMX technology, which is designed specifically to process video, audio, and graphics data efficiently.

MMX is a processor supplementary capability that is supported on IA-32 processors by Intel and other vendors as of 1997. AMD also added MMX instruction set in its K6 processor.

Evolution of the Intel x86 Architecture

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: includes additional floating-point and other enhancements for multimedia.

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.

Embedded Systems

Embedded systems are tightly coupled to their environment. This can give rise to real-time constraints imposed by the need to interact with the environment. Constraints such as speeds of motion, precision of measurement, time durations etc.

Examples include home security systems, washing machines, lighting systems, thermostats, printers, various automotive systems such as transmission control, cruise control, fuel injection, anti-lock brakes, and suspension systems and numerous types of sensors and actuators in automated systems.

Deeply embedded systems often have wireless capability and appear in networked configurations, such as networks of sensors deployed over a large area (e.g., factory, agricultural field).

Embedded Operating System: Example:TinyOS widely used in wireless sensor networks.

Embedded Systems

Figure 1.14 shows in general terms an embedded system organization.

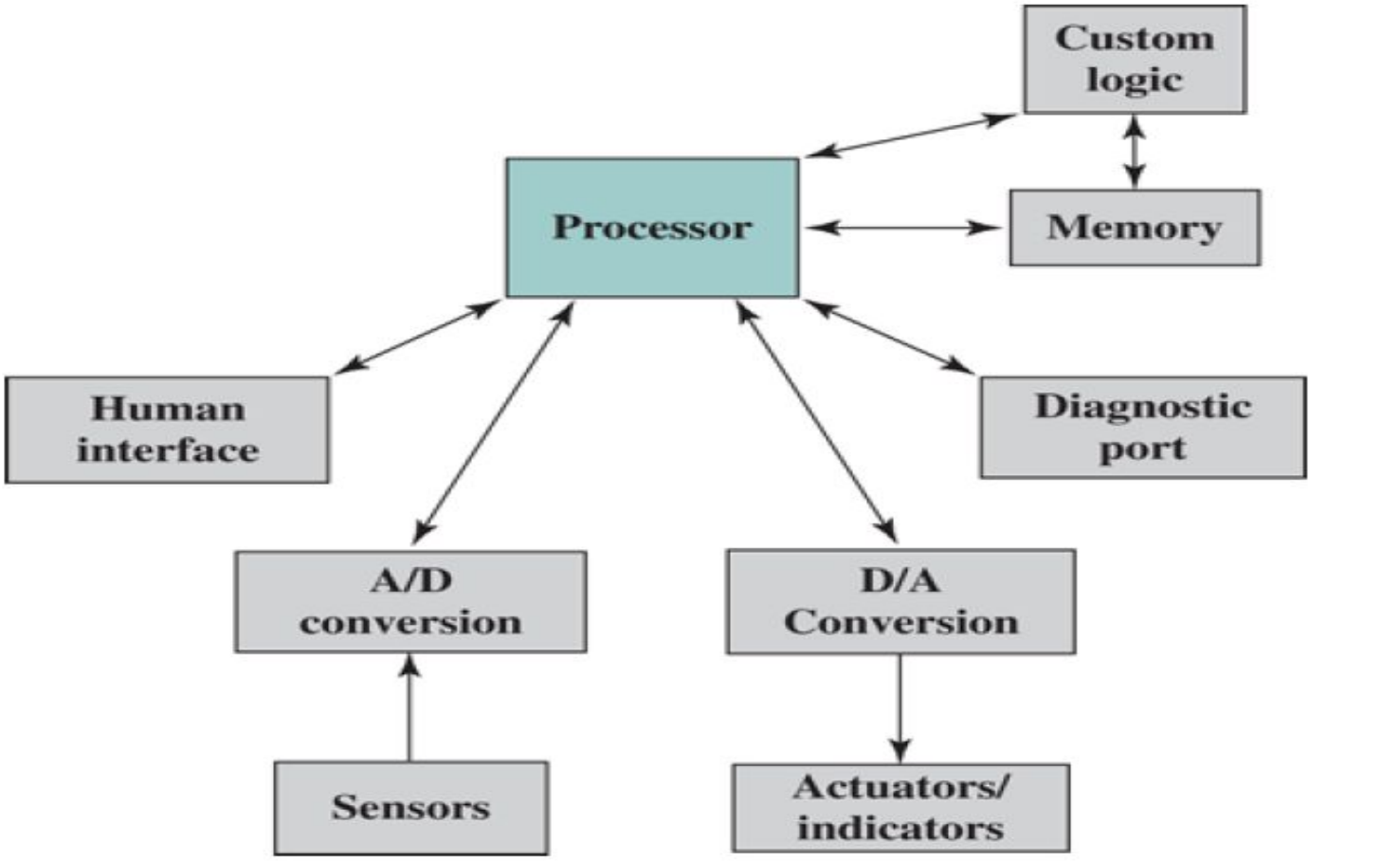


Figure 1.14 Possible Organization of an Embedded System

Embedded Systems

In addition to the processor and memory, there are a number of elements that differ from the typical desktop or laptop computer:

1.There may be a variety of interfaces that enable the system to measure, manipulate, and interact with external environment. Embedded systems interact with external world through sensors and actuators, and are reactive systems; **a reactive system is in continual interaction with the environment and executes at a pace determined by that environment.**

2.Human interface may be as simple as a flashing light or as complicated as real-time robotic vision. In many cases, there is no human interface.

3.The diagnostic port may be used for diagnosing the system that is being controlled—not just for diagnosing the computer.

4.Special-purpose field programmable (FPGA), application-specific (ASIC), or even nondigital hardware may be used to increase performance or reliability.

5.Software often has a fixed function and is specific to the application.

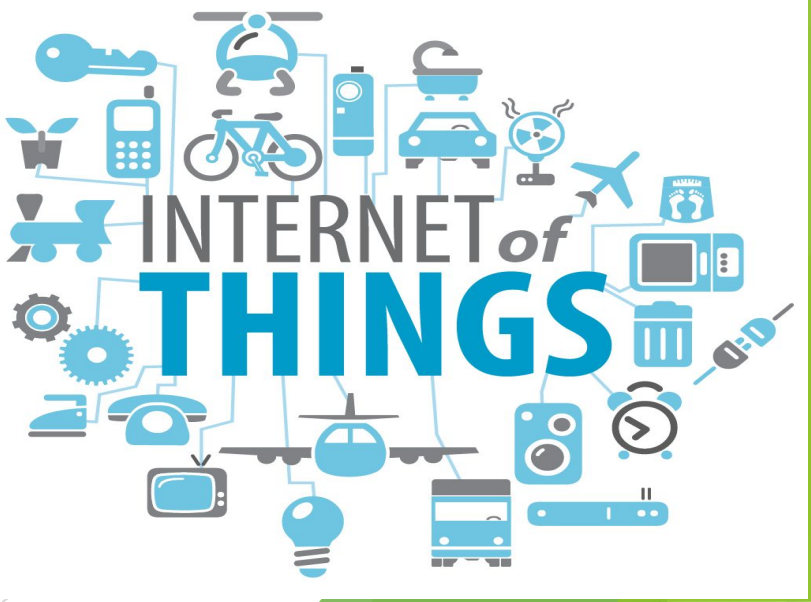
6.Efficiency is of paramount importance for embedded systems. They are optimized for energy, code size, execution time, weight and dimensions, and cost.

Internet of Things

Internet of things (IoT) is a term that refers to the expanding interconnection of smart devices, ranging from appliances to tiny sensors.

Embedding of short-range mobile transceivers into a wide array of gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves.

IoT devices are low bandwidth, low-repetition data-capture, and low-bandwidth data-usage appliances that communicate with each other and provide data via user interfaces. Embedded appliances, such as high-resolution video security cameras, video VoIP phones etc.



Internet of Things

Internet has gone through roughly four generations of deployment culminating in the IoT:

1. Information technology (IT): PCs, servers, routers, firewalls, and so on, bought as IT devices by enterprise IT people and primarily using wired connectivity.

2. Operational technology (OT): Machines/appliances with embedded IT built by non-IT companies, such as medical machinery, SCADA (supervisory control and data acquisition), process control, and kiosks, bought as appliances by enterprise OT people and primarily using wired connectivity.

3. Personal technology: Smartphones, tablets, and eBook readers bought as IT devices by consumers (employees) exclusively using wireless connectivity and often multiple forms of wireless connectivity.

4. Sensor/actuator technology: Single-purpose devices bought by consumers, IT, and OT people exclusively using wireless connectivity, generally of a single form, as part of larger systems

Application Processors vs Dedicated Processors

Application processors: are defined by the processor's ability to execute complex operating systems, such as Linux, Android, and Chrome. The application processor is general-purpose in nature.

A good example of the use of an embedded application processor is the smartphone. The embedded system is designed to support numerous apps and perform a wide variety of functions.

Dedicated processor: is dedicated to one or a small number of specific tasks required by the host device. Because such an embedded system is dedicated to a specific task or tasks, the processor and associated components can be engineered to reduce size and cost.

Microprocessor vs Microcontroller

Microprocessor chips: included registers, an ALU, and some sort of control unit or instruction processing logic. As transistor density increased, it became possible to increase the complexity of the instruction set architecture, and ultimately to add memory and more than one processor.

Microcontroller chip: is a single chip that contains the processor, non-volatile memory for the program (ROM), volatile memory for input and output (RAM), a clock, and an I/O control unit. The processor portion of the microcontroller has a much lower silicon area than other microprocessors and much higher energy efficiency.

Microprocessor vs Microcontroller

Used for the smaller, less expensive microcontrollers, they are used as dedicated processors for specific tasks.

For example, microcontrollers are heavily utilized in automation processes. By providing simple reactions to input, they can control machinery, turn fans on and off, open and close valves, and so forth.

Microcontrollers come in a range of physical sizes and processing power. Processors range from 4-bit to 32-bit architectures.

Microcontrollers tend to be much slower than microprocessors, typically operating in the MHz range rather than the GHz speeds of microprocessors.

Microcontroller is programmed for a specific task, embedded in its device, and executes as and when required.

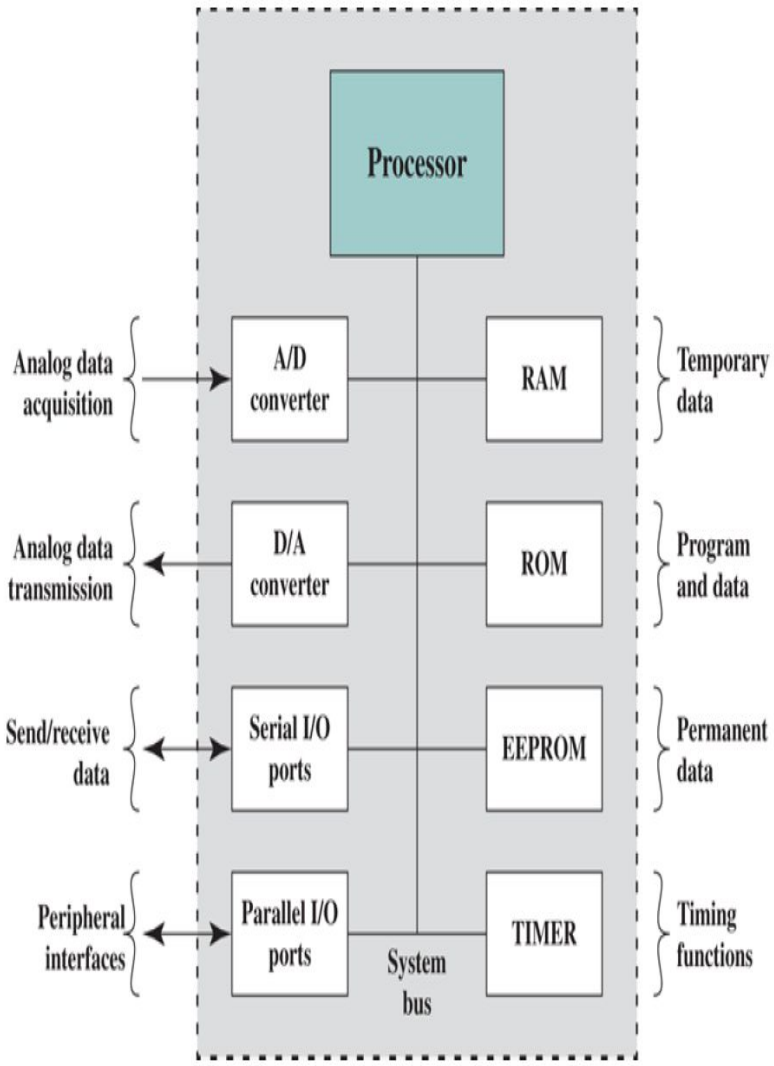


Figure 1.15 Typical Microcontroller Chip Elements

ARM Achitecture

ARM chips are high-speed processors that are known for their small die size and low power requirements. They are used in smartphones and other handheld devices, including game systems, as well as a large variety of consumer products. ARM chips are the processors in Apple's popular iPod and iPhone devices, and are used in virtually all Android smartphones as well.

ARM's partners shipped 16.7 billion ARM-based chips in 2016. ARM is probably the most widely used embedded processor architecture and indeed the most widely used processor architecture.

Instruction Set Architecture: ARM instruction set is highly regular, designed for efficient implementation of the processor and efficient execution.

All instructions are 32 bits long and follow a regular format. This makes the ARM ISA suitable for implementation over a wide range of products. Augmenting the basic ARM ISA is the Thumb instruction set, which is a re-encoded subset of the ARM instruction set.

Thumb is designed to increase the performance of ARM implementations that use a 16-bit or narrower memory data bus, and to allow better code density than provided by the ARM instruction set.

ARM Achitecture

ARM Holdings licenses a number of specialized microprocessors and related technologies, but the bulk of their product line is the Cortex family of microprocessor architectures. There are three Cortex architectures, labeled with the initials A, R, and M.

CORTEX-A: Cortex-A series of processors are application processors, intended for mobile devices such as smartphones and eBook readers, as well as consumer devices such as digital TV and home gateways (e.g., DSL and cable Internet modems).

These processors run at higher clock frequency (over 1 GHz), and support a memory management unit (MMU), which is required for full feature OS such as Linux, Android, MS Windows, and mobile OS.

MMU: is a hardware module that supports virtual memory and paging by translating virtual addresses into physical addresses.

The two architectures use both the ARM and Thumb-2 instruction. Some of the processors in this series are 32-bit machines and others are 64-bit machines.

https://developer.arm.com/documentation/den0013/d/Introduction-to-Assembly-Language/The-ARM-instruction-sets

ARM Achitecture

CORTEX R: Cortex-R is designed to support real-time applications, in which the timing of events needs to be controlled with rapid response to events. They can run at a fairly high clock frequency (e.g., 2 MHz to 4 MHz) and have very low response latency. It includes enhancements both to the instruction set and to the processor organization to support deeply embedded real-time devices.

Most of these processors do not have MMU; the limited data requirements and the limited number of simultaneous processes eliminates the need for elaborate hardware and software support for virtual memory. Cortex-R does have a Memory Protection Unit (MPU), cache, and other memory features designed for industrial applications.

MPU:MPU is a hardware module that prohibits one program in memory from accidentally accessing memory assigned to another active program. Using various methods, a protective boundary is created around the program, and instructions within the program are prohibited from referencing data outside of that boundary.

Examples of embedded systems that would use the Cortex-R are automotive braking systems, mass storage controllers, networking and printing devices.

ARM Achitecture

CORTEX-M: Cortex-M series processors have been developed primarily for the microcontroller domain where the need for fast, highly deterministic interrupt management is coupled with the desire for extremely low gate count and lowest possible power consumption.

As with the Cortex-R series, the Cortex-M architecture has an MPU but no MMU. The Cortex-M uses only the Thumb-2 instruction set. The market for the Cortex-M includes IoT devices, wireless sensor/actuator networks used in factories and other enterprises, automotive body electronics, and so on.

There are currently seven versions of the Cortex-M series:

Cortex-M0: Designed for 8- and 16-bit applications, this model emphasizes low cost, ultra low power, and simplicity. It is optimized for small silicon die size (starting from 12k gates) and use in the lowest cost chips.

Cortex-M0+: An enhanced version of the M0 that is more energy efficient.

ARM Achitecture

Cortex-M3: Designed for 16- and 32-bit applications, this model emphasizes performance and energy efficiency. It also has comprehensive debug and trace features to enable software developers to develop their applications quickly.

Cortex-M4: This model provides all the features of the Cortex-M3, with additional instructions to support digital signal processing tasks.

Cortex-M7: Provides higher performance than the M4. It is still primarily a 32-bit machine but uses 64-bit wide instruction and data buses.

Cortex-M23: This model is similar to the M0+, and adds integer divide instructions and some security features.

Cortex-M33: This model is similar to the M4, and adds some security features.

ARM Cortex-M3 is best suited of all ARM models for general-purpose microcontroller use.It is used by a variety of manufacturers of microcontroller products. Initial microcontroller devices from lead partners already combine Cortex-M3 processor with flash, SRAM, and multiple peripherals.

ARM Achitecture

Figure 1.16 provides a block diagram of the EFM32 microcontroller from Silicon Labs with Cortex-M3 processor and core components.

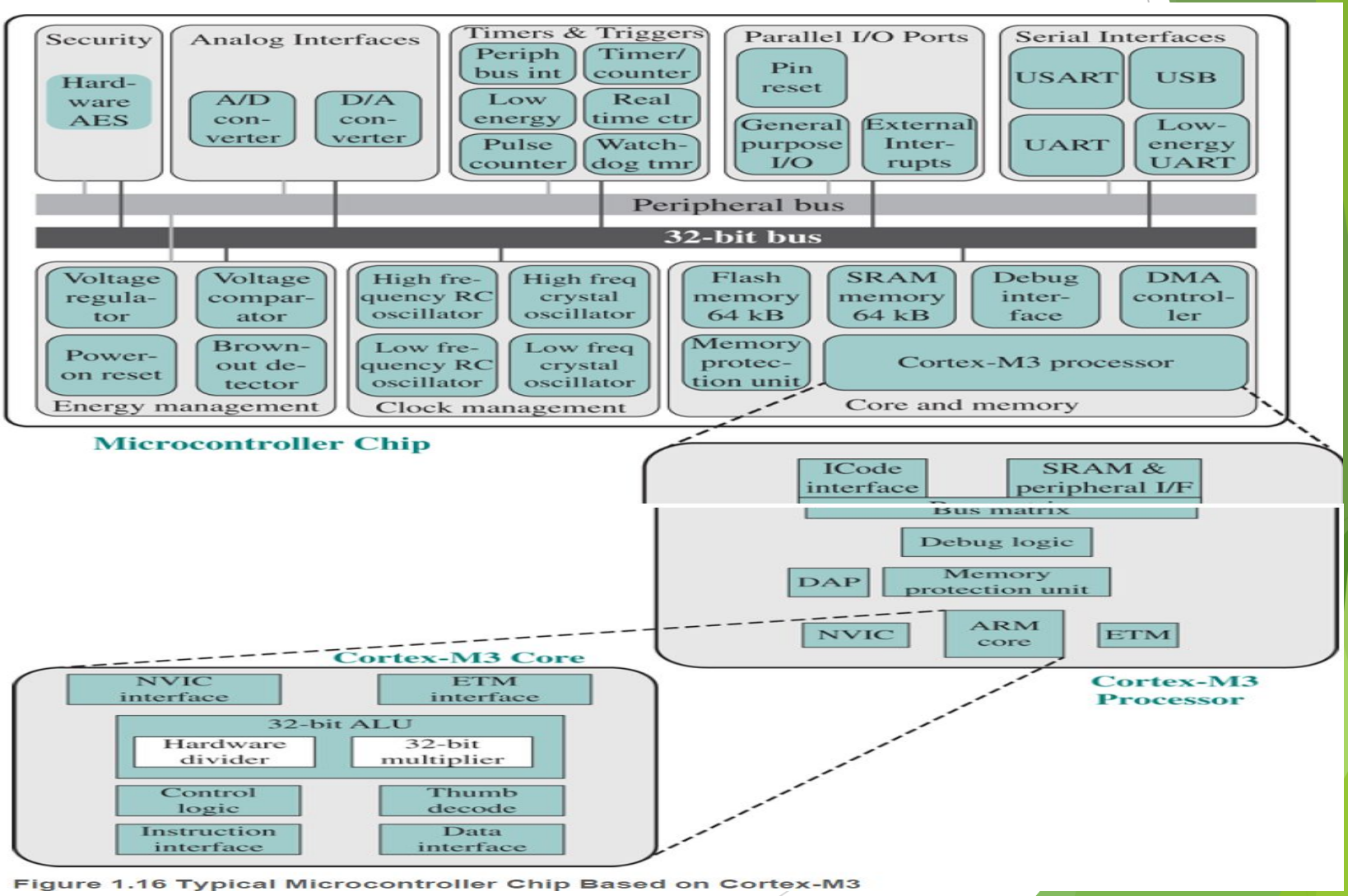


Figure 1.16 Typical Microcontroller Chip Based on Cortex-M3

ARM Achitecture

Figure 1.16 shows microcontroller built with the Cortex-M3.This microcontroller is used in a wide variety of devices, including energy, gas, and water metering; alarm and security systems; industrial automation devices; home automation devices; smart accessories; and health and fitness devices. The silicon chip consists of 10 main areas:

Core and memory: This region includes the Cortex-M3 processor, static RAM (SRAM) data memory, and flash memory for storing program instructions and nonvarying application data.

Flash memory is nonvolatile (data is not lost when power is shut off) and so is ideal for this purpose. Flash memory is a versatile form of memory used both in microcontrollers and as external memoryParallel I/O ports: Configurable for a variety of parallel I/O schemes.

SRAM stores variable data. This area also includes a debug interface, which makes it easy to reprogram and update the system in the field.Static RAM (SRAM) is a form of random-access memory used for cache memory; see Chapter 6.

Serial interfaces: Supports various serial I/O schemes.

ARM Achitecture

Analog interfaces: Analog-to-digital and digital-to-analog logic to support sensors and actuators.

Timers and triggers: Keeps track of timing and counts events, generates output waveforms, and triggers timed actions in other peripherals.

Clock management: Controls the clocks and oscillators on the chip. Multiple clocks and oscillators are used to minimize power consumption and provide short startup times.

Energy management: Manages the various low-energy modes of operation of the processor and peripherals to provide real-time management of the energy needs so as to minimize energy consumption.

Security: The chip includes a hardware implementation of the Advanced Encryption Standard (AES).

32-bit bus: Connects all of the components on the chip.

Peripheral bus: A network which lets the different peripheral modules communicate directly with each other without involving the processor. This supports timing-critical operation and reduces software overhead.

ARM Achitecture

watchdog timer (WDT) is a timer that monitors microcontroller (MCU) programs to see if they are out of control or have stopped operating.

USART (universal synchronous/asynchronous receiver/transmitter) is hardware that enables a device to communicate using serial protocols

UART (Universal Asynchronous Receiver/Transmitter) is the microchip with programming that controls a computer's interface to its attached serial devices.

Low Energy UART (LEUART) provides full UART communication running from a 32.768 kHz clock input.

Electronic oscillator is an electronic circuit that produces a periodic, oscillating or alternating current (AC) signal, usually a sine wave, square wave or a triangle wave.RC and crystal oscillator used for clock management.

Voltage comparator compares two input voltages and outputs a binary signal indicating which is larger.

ARM Achitecture

Voltage regulator is a circuit that creates and maintains a fixed output voltage, irrespective of changes to the input voltage or load conditions. Voltage regulators (VRs) keep the voltages from a power supply within a range that is compatible with the other electrical components.

Power-on reset (PoR) is a circuit that provides a predictable, regulated voltage to a microprocessor or microcontroller with the initial application of power. The PoR system ensures that the microprocessor or microcontroller will start in the same condition every time that it's powered up.

Brown Out Reset A “brown out” of a microcontroller is a temporary reduction in the power supply voltage below the level required for reliable operation.

Many microcontrollers have a protection circuit which detects when the supply voltage goes below this level and puts the device into a reset state to ensure proper startup when power returns. This action is called a “Brown Out Reset”.

A similar feature is called Low Voltage Detect (LVD) which is more complex and adds detection of multiple voltage levels and can produce an interrupt before a reset is triggered.

ARM Achitecture

External interrupts are caused by some external event or failure.

DMA Controller is a type of control unit that works as an interface for the data bus and the I/O Devices. DMA Controller has the work of transferring the data without the intervention of the processors