Chapter 02 – Computer Evolution and Performance

Week – 02

10- 14 September

Topics to Cover

- 2.1 – A Brief History of Computers
- The First, Second, Third and Later Generations of Computers
- The Von-Neuman Machine
- The IAS Computer
- 2.2 – Designing for Performance
- 2.3 – Multicore Processors
- 2.4 – The Evolution of the Intel x86 Architecture
- 2.6 - Performance Assessment

2.1 A Brief History of Computers

- **The First Generation : Vacuum Tubes**
- Electronic Numerical Integrator And Calculator (ENIAC)
- Started 1943 (During World War II)
- Eckert and Mauchly were the Developers
- University of Pennsylvania, Professors
- Trajectory tables for weapons, without tables they were useless
- Finished 1946
  - Too late for war effort
- Used until 1955

ENIAC - Details

- Decimal (not binary)
- 20 accumulators of 10 digits each
- 18,000 vacuum tubes
- 30 tons weight
- 15,000 square feet area
- 140 kW power consumption
- Programmed manually by switches (Drawback)
- 5,000 additions per second

Commercial Computers (UNIVAC)

- 1947 - Eckert-Mauchly Computer Corporation
- UNIVAC I (Universal Automatic Computer)
- US Bureau of Census 1950 calculations
- Became part of Sperry-Rand Corporation
- Late 1950s - UNIVAC II
  - Faster
  - More memory

UNIVAC Series (Investment Plan)

- The UNIVAC II, which had greater memory capacity and higher performance than the UNIVAC I, was made.
- It illustrated several trends that have remained characteristic of the computer industry.
- Advances in technology allowed to build larger and powerful systems.
- Second, each company tries to make its new machines *backward compatible* with the older machines.
- This means that the program written for the older machines can be executed on the new machine.
- This strategy is adopted to retain the customer base; that is when a customer decides to buy a new machine, he buys from same series.

IBM Machines

- Punched-card processing equipment
- 1953 - the 701 Series
  - IBM’s first stored program computer
  - Scientific calculations
- 1955 - the 702 Series
  - Business applications
- Lead to 700/7000 series

What is a Computer Program?

- A **computer program** is a collection of instructions (written in logical order) that performs a specific task when executed by a computer.
- A computer executes the program's instructions in a central processing unit (CPU).
- A computer program is usually written by a computer programmer in a programming language e.g. Assembly language or C++.
- From the program in its human-readable form of source code, a compiler can derive machine code.
- Machine code—a form consisting of instructions that the computer can directly execute.

The Von Neumann Machine

- The task of entering and altering programs for the ENIAC was tedious.
- Von-Neuman gave the idea of a **stored-program concept** and his new stored-program computer is referred to as the IAS computer.
- **Stored-program concept**: A program could be represented in a form suitable for storing in the memory alongside the data.(same memory)
- Then, a computer could get its instructions by reading them from memory, and a program could be set or altered by setting the values of a portion of memory.
- The IAS (Institute of Advanced Studies) computer, is the prototype of all subsequent general-purpose computers.

The IAS Computer (See Fig. Next Slide)

- Four main components of the IAS computer are:

1) Main memory 2) ALU 3) Control unit 4) Input/Output (I/O)

1. A **main memory**, which stores both data and instructions.
2. An **arithmetic and logic unit (ALU)** capable of operating on binary data.
3. A **control unit**, which interprets the instructions in memory and causes them to be executed.
4. **Input/Output (I/O)** equipment operated by the control unit.

Structure of the IAS Computer

- Stored Program concept.
- Main memory stores programs and data.
- ALU operates on binary data.
- Control unit interprets instructions from memory and executes them.
- Input and output equipment is operated by control unit.

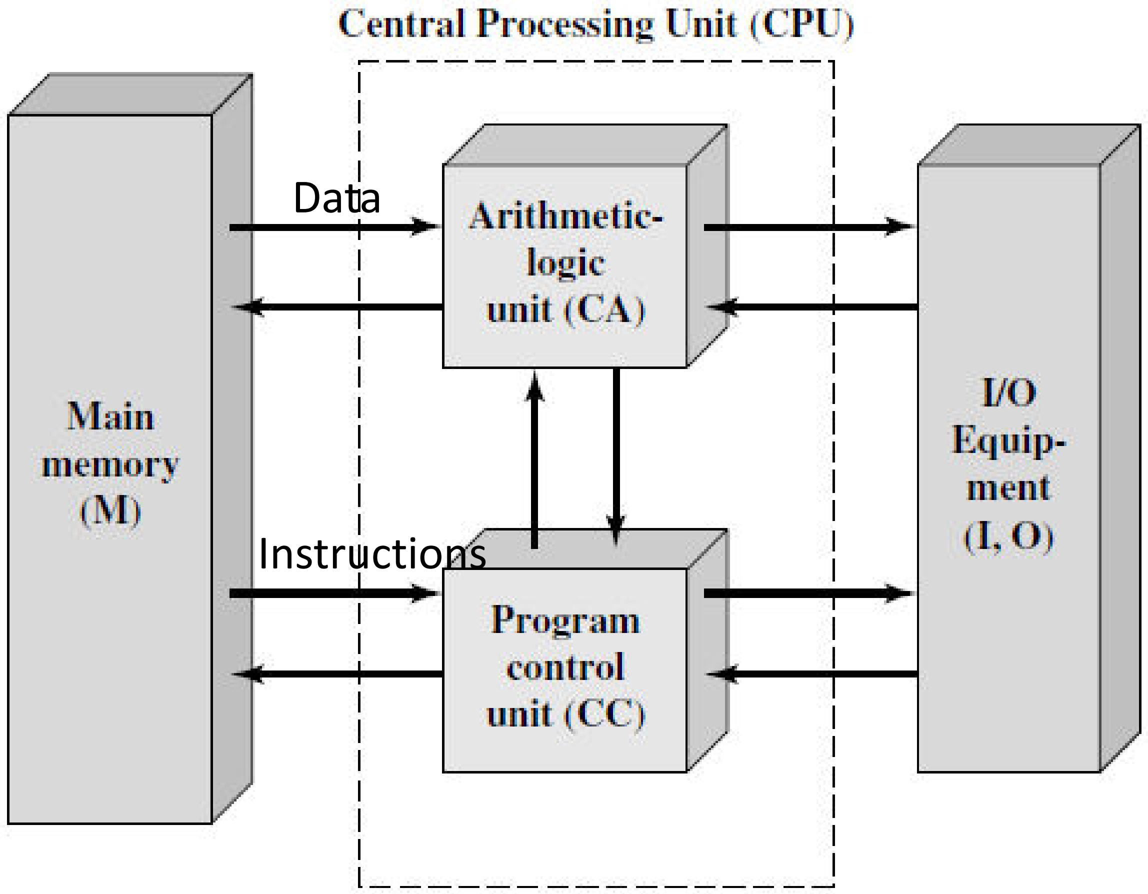

Central Processing Unit (CPU)
Main memory (M)
Data
Instructions
Arithmetic-logic unit (CA)
Program control unit (CC)
I/O Equipment (I, O)

Figure 2.1 Structure of the IAS Computer

IAS Structural Components

- Since the device is primarily a computer, it will have to perform the elementary operation of arithmetic most frequently.
- These are addition, subtraction, multiplication, and division.
- This requires a **central arithmetic (ALU)** part of the device to exist.
- The logical control of the device, that is, the proper sequencing of its operations, can be most efficiently carried out by a **central control (CU)** organ.
- Any device that is to carry out long and complicated sequences of operations (specifically of calculations) must have a **memory** unit.

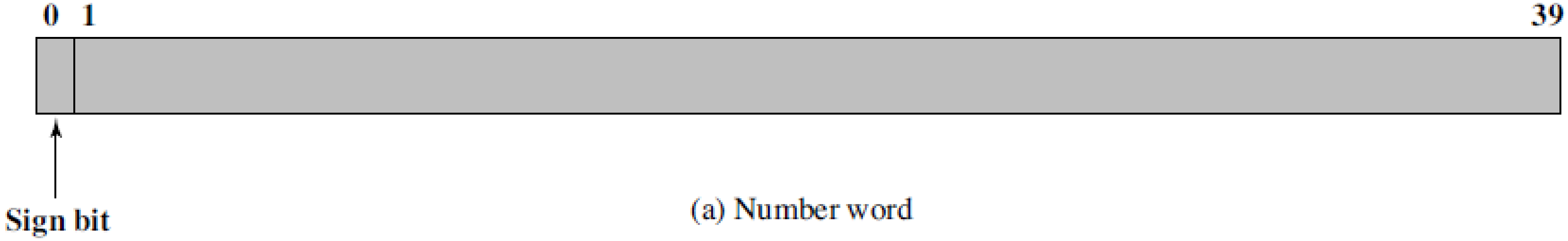
IAS Structural Components

- The device must have organs to transfer information from the outside medium into its specific parts CPU and memory.
- These organs form its **input**.

- The device must have organs to transfer from its specific parts CPU and memory into the outside medium.
- These organs form its **output**.

The Von-Neuman IAS Arcthitecture

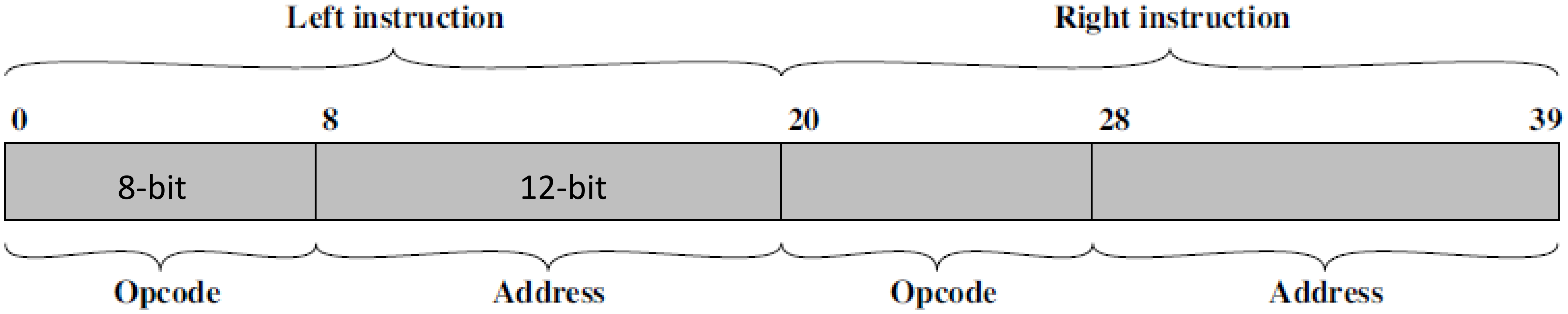
- All of today’s computers have this same general structure and function and are thus referred to as **von Neumann machines.**
- The main memory of the IAS consisted of 1000 storage locations, called **words**.
- Each word has a width of **40 binary digits (bits).** (1000 x 40 bit words)
- Both data and instructions are stored there.
- Numbers are represented in binary form, and each instruction is a binary code.
- Each number is represented by a <u>sign bit</u> and a 39-bit value.



(a) Number word

IAS Memory Formats

- A word may also contain two 20-bit instructions.
- Each instruction consisted 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).



The ‘Registers’ for ALU and Control Unit

- The control unit operates the IAS by **fetching** instructions from memory and executing them one at a time.
- Both the control unit and the ALU contain storage locations in CPU, called *registers*, defined as follows:

ALU Registers:
Accumulator (AC)
Multiplier Quotient (MQ)
Memory Buffer Register (MBR)

Program Control Unit Registers:
Memory Address Register (MAR)
Instruction Register (IR)
Instruction Buffer Register (IBR)
Program Counter (PC)

Structure of IAS

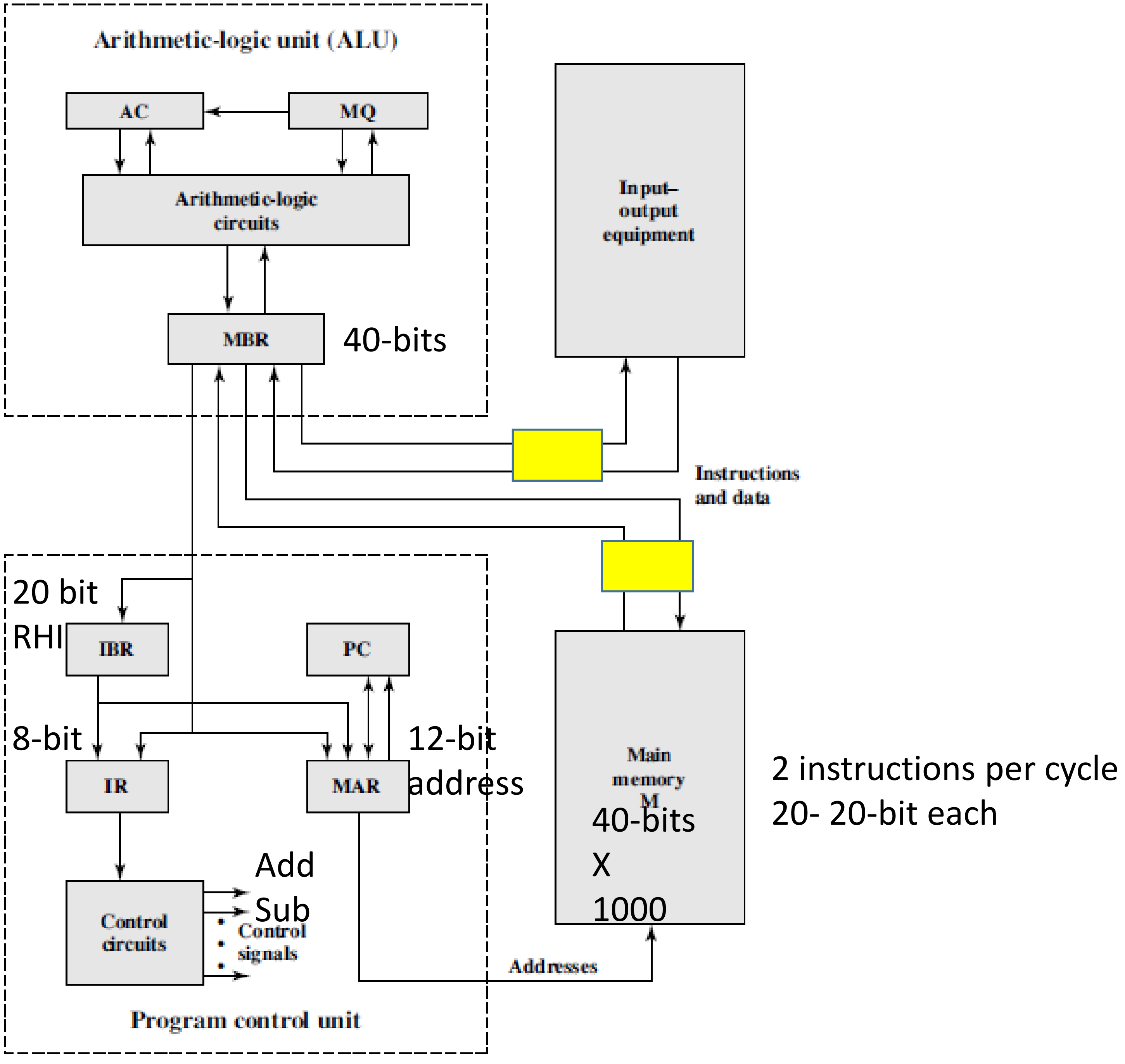


Figure 2.3 Expanded Structure of IAS Computer

ALU Registers

- **Memory Buffer Register (MBR):** contains a 40-bit word to be stored in memory or sent to the I/O unit, or is used to receive a word from memory or from the I/O unit.

- **Accumulator (AC) and Multiplier Quotient (MQ):** are 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.

Program Control Unit Registers

- **Memory Address Register (MAR):** is a 12-bit register that 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 by the 'control circuits' e.g. add, multiply.
- **Instruction Buffer Register (IBR):** is employed to hold temporarily the right-hand instruction of 20-bit from a word in Memory. While the left hand instruction goes to IR and MAR (20-bit).
- **Program Counter (PC):** contains the address of the next instruction pair to be fetched from memory.

(Half)

What is an ‘Instruction Cycle’?

- An **instruction cycle** (sometimes called a fetch–decode–execute cycle) is the basic operational process of a computer.
- Each ‘instruction cycle’ consists of two sub cycles,

  1) Fetch cycle 2) Execute cycle

- **Instruction cycle** is the process by which a computer retrieves a program **instruction** from its memory called **fetch cycle.**
- Then it determines what actions the **instruction** dictates, and carries out those actions called **execute cycle.**

Steps of ‘Instruction Cycle’: Fetch-Decode-Execute

- To execute a program, the program code is copied from secondary storage into the main memory.
- In a program, each **machine code** instruction takes up a slot in the main memory.
- These slots (or memory locations) each have a **unique memory address**.
- The CPU's **program counter** is set to the memory location where the first instruction in the program has been stored and execution begins.
- PC register tells the CPU the order of instructions, program is running.
- When a program is being executed, the CPU repeats this cycle again.

The Operation of the IAS (Fetch Cycle)

- The IAS operates by repetitively performing an **instruction cycle**.
- Each ‘instruction cycle’ consists of two sub cycles, **fetch cycle** and **execute cycle.**
- 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.

The Operation of the IAS (Execute Cycle)

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

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

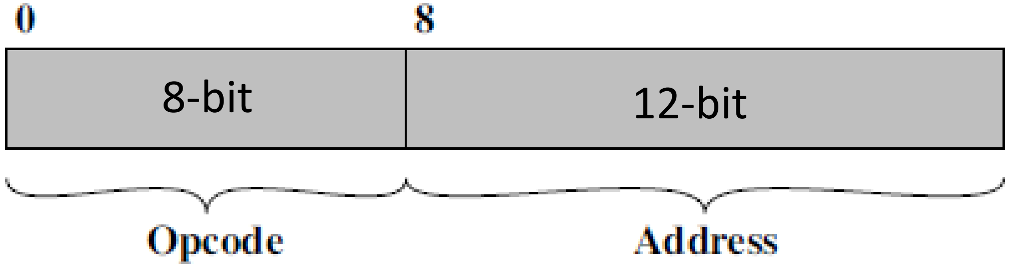
- For Example: The opcode: 00000101 is interpreted by the ‘control circuit’ as ADD M(X), and it Adds M(X) to AC; and puts the result in AC.

Functions Performed by IAS Instructions

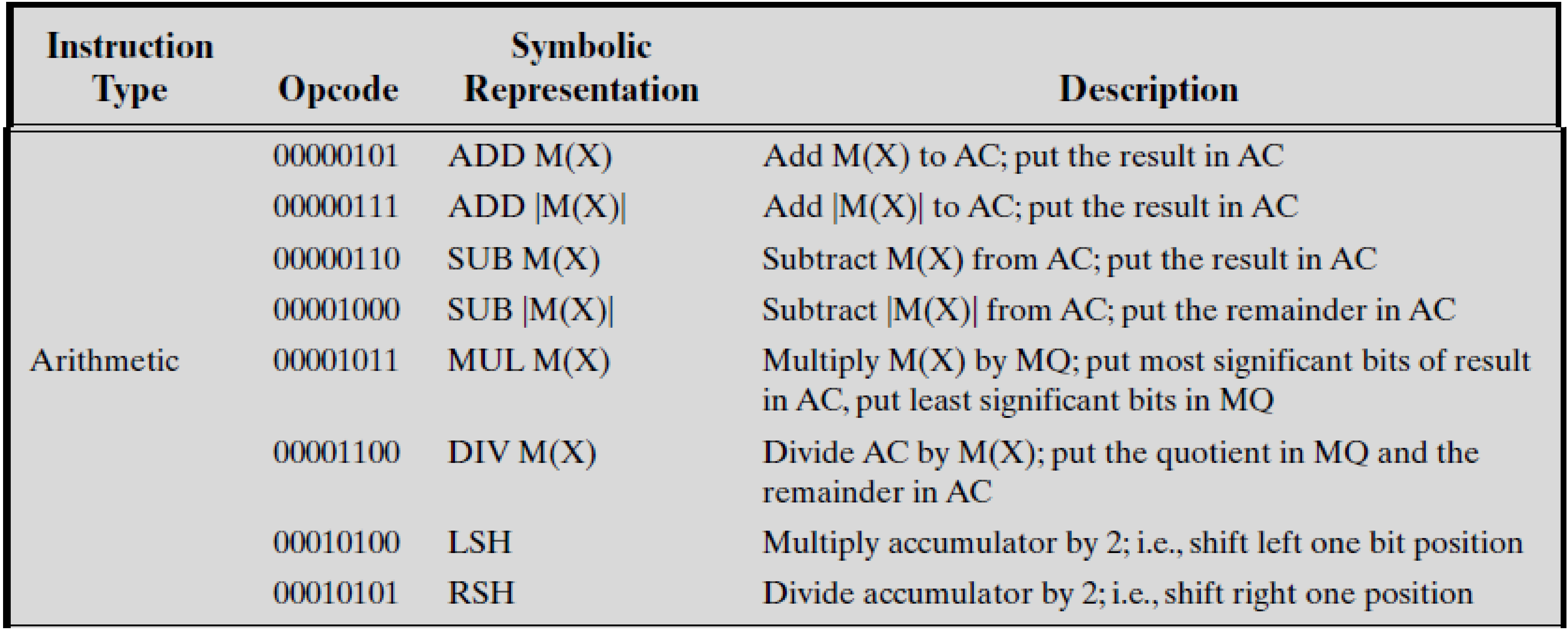
- The IAS computer had a total of 21 instructions.
- These instructions can be grouped as follows:
- **Data transfer:** Move data between memory and ALU registers.
- **Unconditional branch:** Normally, the control unit executes instruction in sequence from memory. This sequence can be changed by a branch 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 address to be computed in the ALU and then inserted into instructions stored in memory.

The IAS Instructions

- The ‘opcode’ portion (first 8 bits) specify which of the 21 instructions is to be executed. The ‘address’ portion (remaining 12 bits) specifies which of the 1000 memory locations is to be involved in the execution of the instruction.
- All of the 21 IAS instructions are categorized as follows:
- **Data transfer:** LOAD, STORE.
- **Unconditional branch**: JUMP.
- **Conditional branch**: JUMP+Condition.
- **Arithmetic**: ADD, SUB, MUL, DIV, LSH, RSH.
- **Address modify**: STOR



The IAS Instruction Set (Arithmetic)



Instruction Type	Opcode	Symbolic Representation	Description
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

Harvard Architecture

- Under pure von Neumann architecture the CPU can be either reading an instruction or reading/writing data from/to the memory. Both cannot occur at the same time since the instructions and data use the same bus system. In a computer using the Harvard architecture, the CPU can both read an instruction and perform a data memory access at the same time, even without a cache. A Harvard architecture computer can thus be faster for a given circuit complexity because instruction fetches and data access do not contend for a single memory pathway .

2.1 A Brief History of Computers

- The Second Generation : Transistors
- Replaced vacuum tubes (they generate heat, were bulky, unreliable)
- Invented 1947 at Bell Labs
- William Shockley et al.
- Smaller
- Cheaper
- Less heat dissipation
- Solid State device
- Made from Silicon (Sand)

Improvements in the Second Generation

- ‘Computer generations’ are classified based on the fundamental hardware technology employed e.g. transistors.
- Each new generation is characterized by greater processing performance, larger memory capacity, and smaller size than the previous one.
- Also the second generation saw the introduction of more complex arithmetic and logic units and control signals, the use of high level programming languages, and the provision of *system software* with the computer.
- ‘System software’ provide the ability to load programs, move data to peripherals, and to perform common computations, similar to Windows.

Transistor Based Computers

- Second generation machines
- NCR & RCA initially produced small transistor machines
- IBM 7000 series came later
- DEC (Digital Equipment Corporation) - 1957
    - Produced its first computer PDP-1
- This computer and this company began the mini-computer phenomenon that became so prominent in the third generation.

The Third Generation: Integrated Circuits (ICs)

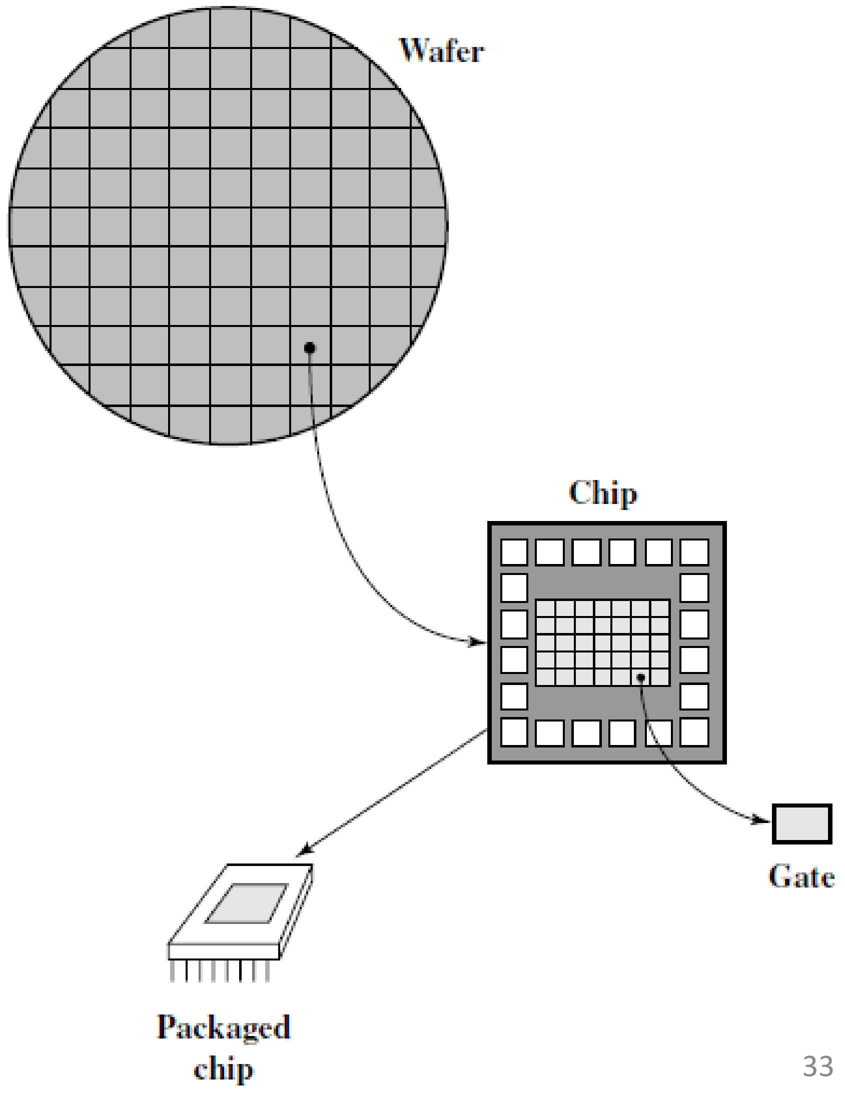
- A single transistor is a discrete component.
- Throughout the 1950 these discrete components were manufactured separately, packaged in their own containers, and soldered together onto circuit boards, which were then installed in computers. The addition of a new transistor has to be soldered again on circuit board.
- The entire manufacturing process, from transistor to circuit board, was expensive and cumbersome.
- In 1958 came the achievement that revolutionized electronics and started the era of electronics: the invention of the **integrated circuit.**
- It is the **integrated circuit** that defines the third generation of computers.

Micro-Electronics

- Literally means “small electronics” has tremendously improved speed.
- There has been a consistent trend toward the reduction in size of digital electronic circuits thus leading to ‘microelectronics’.
- A computer is made up of (logic) gates, memory (storage) cells and interconnections.
- The **integrated circuit** uses the fact that such components, and paths can be fabricated onto a semiconductor such as Silicon wafer (Chip).
- To fabricate an entire circuit in a tiny piece of silicon, in this way many transistors can be produced at one time on a single wafer of silicon.

Relationship among Wafer, Chip, and Gate

- A thin **wafer** of silicon is divided into a matrix of small areas, a few millimetres square.
- The identical circuit pattern is fabricated in each area, and then wafer is broken up into **chips.**
- Each chip consists of many **gates** and/or **memory cells**.
- These chips can be connected on a PCB to produce complex circuits.



Generations of Computers

- Vacuum tubes - 1946-1957
- Transistors - 1958-1964
- Small scale integration - 1965 on
  - Up to 100 devices on a chip
- Medium scale integration - to 1971
  - 100-3,000 devices on a chip
- Large scale integration - 1971-1977
  - 3,000 - 100,000 devices on a chip
- Very large scale integration - 1978 -1991
  - 100,000 - 100,000,000 devices on a chip
- Ultra large scale integration – 1991 -
  - Over 100,000,000 devices on a chip

Typical Speed (operations per second)
Vaccum tubes 40,000
Transistors 200,000
Small- and medium-scale integration 1,000,000
Large-scale integration 10,000,000
Very-large-scale integration 100.000.000
Ultra-large-scale integration 1,000,000,000

Moore’s Law

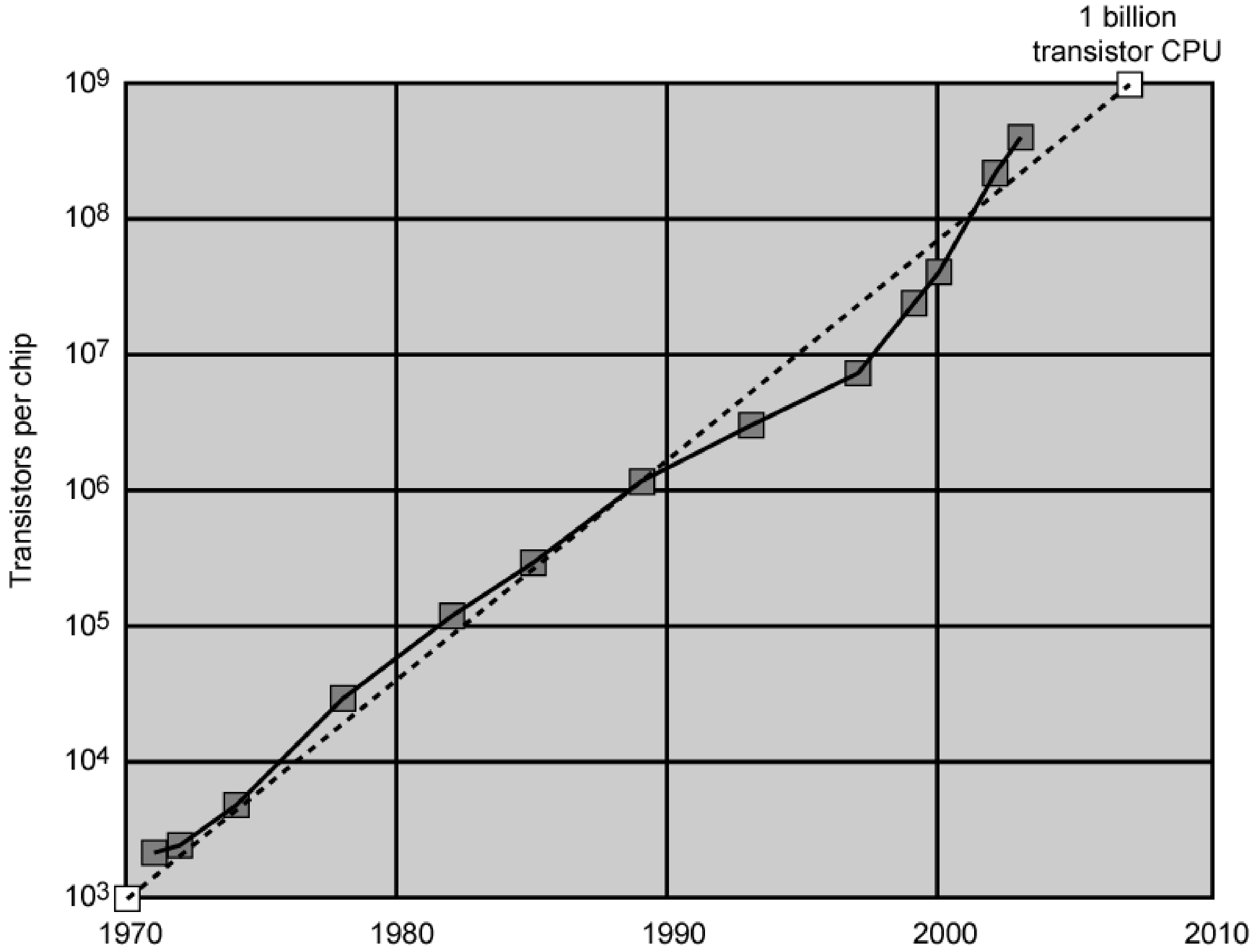
- Gordon Moore, co-founder of Intel, in 1965 presented his famous law.
- Moore observed that **the number of transistors that could be put on a single chip was doubling every year** (12 months).
- Furthermore he predicted that this pace would continue into the near future. Predicted increased density of components on a single chip.
- To the surprise of many, the pace continued year after year and decade after decade.
- The pace slowed to a **doubling every 18 months** in the 1970s but has sustained that rate ever since.

Consequences of Moore’s Law

- The consequences of Moore’s law are profound:

1. Cost of a chip has remained almost unchanged during this period of rapid growth in chip density.
2. Higher packing density means shorter electrical paths, giving higher performance, increased operating speed.
3. Smaller size gives increased flexibility. (more suitable to fit in home)
4. Reduced power and cooling requirements.
5. Fewer interconnections increases reliability.

Growth in CPU Transistor Count



Preparatory Questions (Week 2)

- **Q1. What is a ‘stored program’ computer?**
- **Q2. Define ‘ALU Registers’ of the IAS?**
- **Q3. Define ‘Program Control Unit Registers’ of the IAS?**
- **Q4. Define the structure of IAS computer in detail?**
- **Q5. Explain the consequences of Moore’s law?**

Next Class Topics

- Later Generations
- 2.2 Designing for Performance
- 2.3 Multicore Processors
- 2.4 The Evolution of Intel Architecture
- 2.6 Performance Assessment
- --