COMPUTER ORGANIZATION & ARCHITECTURE

EIE 411

Course Lecturer: Engr Banji Adedayo. Reg COREN.

The characteristics of different computers vary considerably from category to category. Computers for data processing activities have different features than those with scientific features. Even computers configured within the same application area have variations in design.

Computer architecture is the science of integrating those components to achieve a level of functionality and performance. It is logical organization or designs of the hardware that make up the computer system. The internal organization of a digital system is defined by the sequence of micro operations it performs on the data stored in its registers. **The internal structure** of a MICRO-PROCESSOR is called its architecture and includes the number lay out and functionality of registers, memory cell, decoders, controllers and clocks.

HISTORY OF COMPUTER HARDWARE

The first use of the word 'Computer' was recorded in 1613, referring to a person who carried out calculation or computation.

A brief History:

Computer as we all know 2day had its beginning with 19th century English Mathematics Professor named Chales Babage. He designed the analytical engine and it was this design that the basic frame work of the computer of today are based on.

1st Generation 1937-1946

The first electronic digital computer was built by Dr John V. Atanasoff & Berry Cliford (ABC). In 1943 an electronic computer named colossus was built for military.

1946 – The first general purpose digital computer- the Electronic Numerical Integrator and computer (ENIAC) was built. This computer weighed 30 tons and had 18,000 vacuum tubes which were used for processing. When this computer was turned on for the first time, lights dim in section of Philadelphia. Computers of this generation could only perform single task, and they had no Operating system.

2nd Generation 1947-1962 use transistors instead of vacuum tubes. In 1951 the Universal Automatic Computer (UNIVAC) as the first set of commercial computers were introduced to the public. In 1953- International Business Machine (IBM 650 & 700) made their mark, computer programming languages were developed, computer had memory and OS. Storage media such as tape and disk were in use, also were printers for output.

3rd Generation 1963 to date – The invention of IC, computers became smaller, more powerful, more reliable and able to run many programs at the same time.

In 1980 MS-DOS was born

In 1981 IMB introduced the Personal Computer (PC) for homes and office.

In 1985 Apple gave us the Macintosh computers with its icon driven interface and 90's gave us Window Operating System.

The melelium computers in the year2000 series computers laptops with built-in-video camera, palmtops, flat screen, Apple tablets, hp slate computer, Lenovo hybrid note book and many 'computer on the go' such as Blackberry with touch-sensitive screen, ability to recognize voice commands, and dictation of texts, a built-in-video camera.

As a result of the various improvements to the development of the computer, we have seen the computer being used in all areas of life be it: telecoms, space technology, marine, education, health, business and as artificial intelligence (Robotics).

COMPONENTS OF COMPUTER

While structure of computers could be likened to the anatomy and physiology of the computer, like the classification of humans as consisting of body and soul, computer could be subject to like that namely:

Computer Hardware: this refers as electronic and mechanical parts. Hardware includes input devices, output devices, system unit, storage unit and communication devices.

Input devices- it enables us enter data into computer eg keyboard, mouse, scanner, microphone, digital camera, pc video camera.

Output devices- It gives information to the users eg Monitor (VDU), printer, speaker.

The system Unit- it contains all the electronic components which is the internal hardware.

They are: power supply Unit, mother board, RAM, ROM, CD Drive, hard Disk, the monitor (VDU), keyboard, mouse.

Optional Peripherals: stabilizers/surge protectors, UPS.

The assembly, maintenance of **system unit** is handled by **system Engineer**.

Computer Software- Computer will not work without software. Software also call programs are instructions that tell computer what to do and how to do it.

Three main Software: 1. System software (Also called OS) eg Unix, MS-DOS, Window NT, Sun OS, etc 2. Application Software- It allows specific task eg word processing, excel, power point, e-mail application, internet application (Internet explorer, fire force)

3. utility software eg antivirus, Zip, Maintenance software (tele-diagnosis) etc.

The prophesy on the death of PC

The two words that will kill the dominating world of computing by Bill gate is **"CLOUD COMPUTING".** The days of paying for costly software upgrades are numbered. The PC will soon be obsolete.

IT consulting firm IDC reports that every dollar a company spends on a Microsoft products results in an additional $8 of IT expenses.

The two words CLOUD COMPUTING scare the hell out of Bill Gates. Bill Gate has to say: "the next sea change is upon us". Thanks to the thousands of miles of fiber-optic cable laid during the late 1990's and middle of 2000's in Nigeria, the speed of computer networks has finally caught up to the speed of the computer processors.

Suddenly computers that were once incompatible and isolated are now linked in a giant network or "Cloud". As a result, computing is fast becoming a utility in much the same that electricity did. Think back a few years –any time you wanted to type a letter, create a spread sheet, edit a photo or drawing or play a game, you had to go to the store, and install it on your computer. Presently in Nigeria every computer users in Nigeria will be struggling to hook to the "CLOUD".

GENERAL COMPUTER ARCHITECTURE

A typical computer system can be divided into 3 sections namely:

(a) Control section: It executes instruction and processes data
(b) Memory section: It stores data instructions

(c) Input and output section: handles communications between the computer and outside world.

Also, these sections are interconnected to one another by tiny wires which constitute the conducting paths called buses.

Buses are divided into three:

- Control bus
- Data bus
- Address bus

Control Bus: used to send control signals generated in timing and control circuits of microprocessor to the memory and the input and output units (I/O). Control bus is uni-direction. eg It carries read or write signals to memory and i/o parts

Data Bus: Used to transfer machine instructions and data from the memory to the microprocessor; and data associated with I/O transfers only. Data bus is bi-directional in nature as distinct from control and address busses (one direction). It consists of tracts that carry 8 bits or 1 byte word simultaneously.

Address Bus: used to transfer the address of the location in memory or the I/O part involved in data transfer.

N.B: Some of these buses may be physically the same, a single bus may carry data in different directions or addresses at times. As a result, a bus for more than one purpose is shared or multiplexed. Shown below is the block diagram of typical computer system.

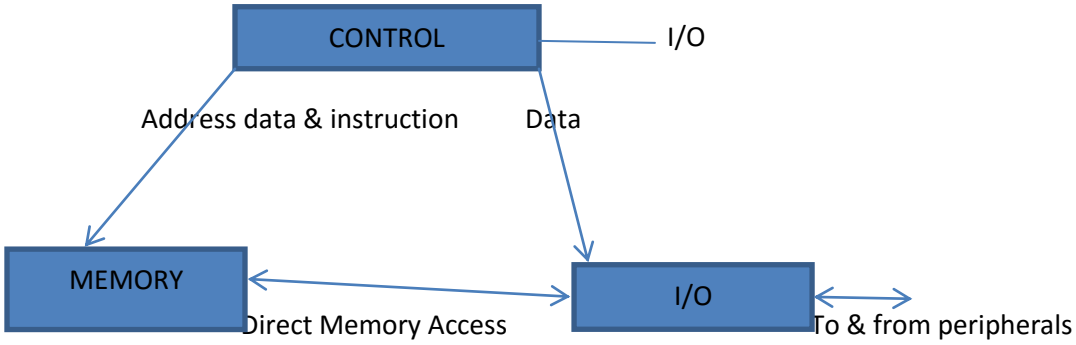


Fig 1a

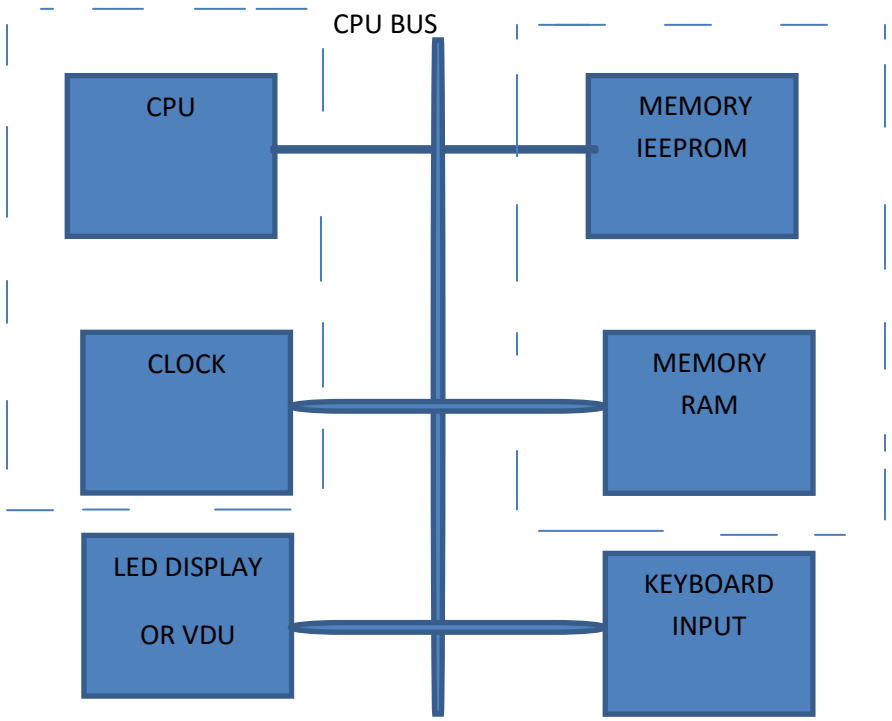
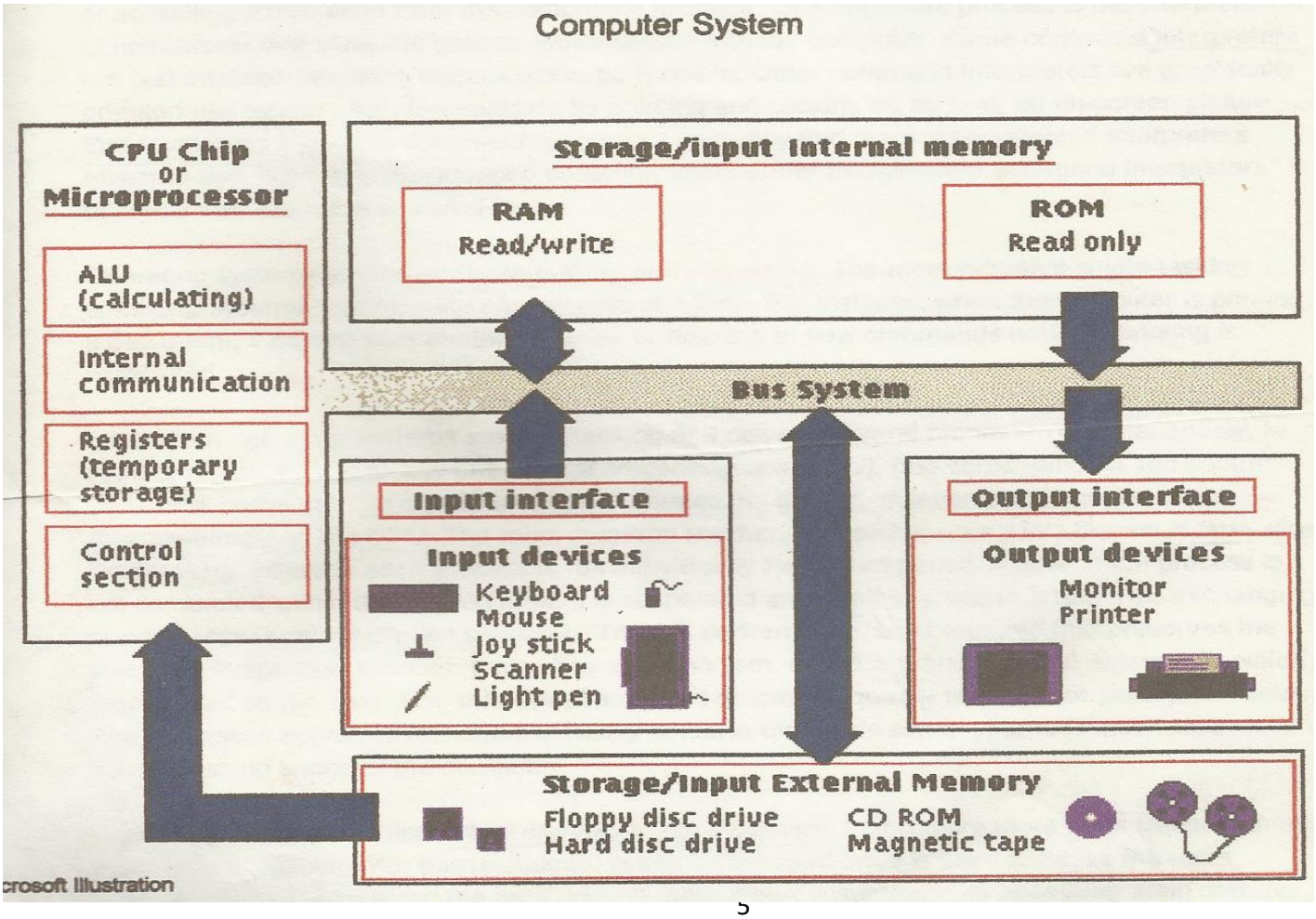


Fig 1b

Fig1a &b Shows a block diagram for a generic computer system.



A computer is a machine that computes, processes and store information. A **micro-computer** is a computer that has been integrated into a very large scale integrated circuit (VLSIC). There are all sorts of micro-computers, but one important class of micro-computer is Micro-controller. A **micro-controller** is a micro-computer that has been specially designed to interact in a timely manner with different types of devices in the outside world.

The Fig1b shows that a computer consists of a **CENTRAL PROCESSING UNIT (CPU**), a **CLOCK**, **MEMORY** and **PERIPHERAL** or **I/O** devices. All of these subsystems communicate over a CPU bus.

INPUT/OUTPUT SECTION

It handles the transfer of data between the computer and external devices or peripherals. The transfer involves status and control signal as well as data. The input/output must reconcile time difference between the computer and the peripherals, format the data properly, handle status and control signals and supply the required voltage level, irregular transfer may be handled with interrupts control signals that receives immediate attention of the control section and cause the suspension of its activity.

A typical I/P operation proceeds as follows:

- The peripherals signal the control section that new data is available. The I/O section must format the signal properly and hold it until the control accepts it.
- It sends data to the control section. The I/O section must hold the data until control section is ready to read it.
- The control section reads the data. The I/O section must have a decoding mechanism that select a particular part of the instruction. After reading the data, the control signal is deactivated and the peripheral is acknowledged for more data to be sent.

Output operations are much like input operations.

- The peripherals indicates to the control sections that it is ready to accept data
- The control section then sends the data along with a signal (a strobe) that indicates to the peripheral that the data is available.
- The I/O section formats the data and control signal properly; and hold the data long enough for the peripheral to use it.

- O/P data must be held longer than I/P data, since mechanical devices, lighted displays and human observers respond much more slowly than computers.

FUNCTIONS OF THE I/O SECTION

1. The I/O section must perform many simple interfacing tasks. It must place signals in the proper form for both the control section and the peripheral to understand.
2. The I/O section may perform some functions that the control section could perform such as the following: (I). conversion of data between serial (one bit at a time) and parallel (more than one bit at a time) form (ii) The insertion or detection of special patterns that mark the beginning or end of data transfer iii) The calculation and checking of error detection codes, such as parity.
3. Also, the I/O section of a computer may be programmed or may even contain a processor so that it can handle some of the processing task.

THE CONTROL SECTION.

In any typical computer system, control section is vested with overall responsibility and ability to coordinate the activity of the machine. It is the" brain" where "every thought" and synchronization of all other sections in the machine is initiated. Besides, other functions being performed are highlighted as follows:

- The control section **(CPU or Processor)** processes the data
- It fetches instructions from memory, decodes and executes them.
- It generate timing or control signals
- It transfer data to and from the memory and I/O sections
- It performs arithmetic and logical functions
- It recognizes external signals

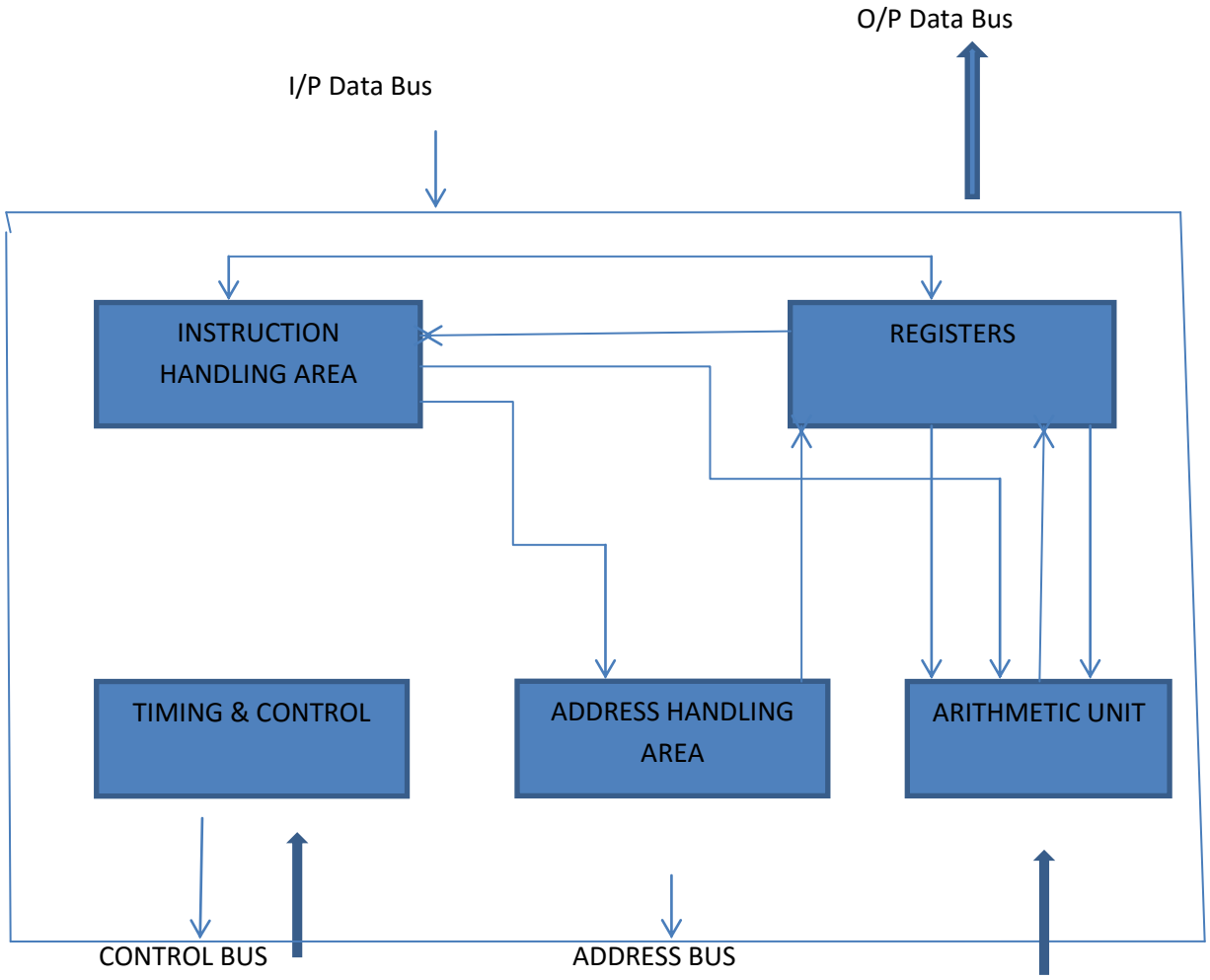


Fig 2 A typical example of control section

During each instructions cycle, the control section performs many tasks:

1. It places the address of the instruction on the memory address bus
2. It takes the instruction from the input data and memory address bus
3. It fetches the address and data required by the instruction. The addresses and data may be in memory or in any of the registers.
4. It performs the operation specified by the instruction code. The operation may be an arithmetical or logical function, a data transfer or management functions.
5. It looks for control signals, such as interrupts and provides appropriate responses that are required.
6. It provides status, control and timing signals that the memory and I/O section can use.

The control section or CPU is the centre of the computer operations because of its ability to both process data and direct the entire computer.

MEMORY SECTION

It consists of storage units which comprises either magnetic cores or semiconductors cells. The units are of binary constituents that have two states i.e value one or zero. Memory is organized sequentially into bytes and words, each of which has a unique address. Its contents are rightly accessed by virtue of their unique addresses. Based on its designs, memory could either be accessed randomly or sequentially. Different locations in random memory are accessed at virtually the same amount of time while sequentially memory locations are accessed at distinctly different times.

Access time is the time it takes to find the correct memory location and obtaining its content. It affects the speed of the computer since the computer must obtain its instructions and most of the data from the memory. Memory access speed in most computers ranges between 100ns (nano-second) to several microseconds.

Memory sections are often subdivided into units called pages. The entire memory may involve billions of words, whereas a page contains 256 and 4k words. Most memory accesses are accomplished by first accessing the required particular page and then accessing the required location on that page.

Advantage: The computer can reach several locations on the same page. The control section transfers data to and from the memory as follows:

1. The control section sends an address to the memory
2. The control section sends signal (read/write) to the memory to indicate the direction of transfer of the signal.
3. The control section waits until the transfer has completed. The delay precedes the actual data.

REGISTERS

These are memories that reside in the control section aside the arithmetic and logic unit, address decoding unit, instruction handling unit and; timing and control unit. Like other form of memories, registers consist of flip-flops and its capacity is in bytes. A 4-bit register is made up of 4 flip-flops and hold 4-bits of information. Registers can be used in various ways as counters, shift registers, increment and decrement counters with parallel load. These are temporary storage devices used in the internal storage of intermediate data in the central processing unit (CPU). They are advantageous under program control because the CPU can obtain data from them without memory access. However, those that are not under program control allow the control section to save on them for later use.

They have **internal buses** by which they are interconnected together. Registers are expensive because of complexity and difficulty involved in producing it especially at the manufacturing stage, hence limiting the number found in a computer.

There are different types of registers:

- Program counter (PC)
- Instruction Registers (IR)
- Memory Address Registers (MAR)
- Accumulators
- General purpose Registers
- Condition Code Registers
- Stack Pointers
- Index Registers

N.B: A processor is 64/32 bits. It means the device has 64bit registers and 32 bit data bus.

The size of a processor indicates the maximum no of bits which the µP can communicate to the" outside world" in a single operation or simply put as the width of processor data bus eg Hybrid sizes such as 16/8-bit, 32/16-bit, 64/32-bit etc designates a processor by internal register size and data bus. In Intel 8088 µP, the heart of the IBM PC, is a 16/8-bit device with 16-bit registers and an 8-bit data bus. Engr Adedayo's PC has Intel (R) B960 processor (64/32-bit). Here the outside world refers to the various peripherals devise attached to the µP eg memory, hard-disk, display VDU, printer, keyboard etc

A computer (micro or otherwise) is ther4 a µP with needed peripheral devices to make it a functional useful whole.

Program Counter (PC): Contains the address of the memory location that contains the next instructions. The instruction cycle begins with the CPU placing the contents of the PC on the address bus, the CPU then fetches the first word of the instruction from the memory. The CPU also increments the content of the PC so that the next instruction cycle will fetch the next sequential address in memory. The CPU fetches the instructions sequentially until there is a branch or jump instruction that changes the content of the PC.

INSTRUCTION REGISTER (IR): Holds instruction until it is decoded. The bit length of the instruction register is the length of the basic instruction word that the computer would accommodate. The IR cannot rightly be accessed by the programmer because it is under the total power of the section. Some computers have two IRs so that they can fetch and save one while executing the previous one (Pipe lining process)

Memory Address Registers (MAR): Holds the address of data in memory. The address may be part of the instructions or may be provided separately from the instructions or the program eg LOAD ACCUMULATOR FROM MEMORY loads the contents of desired memory locations and places it in accumulator.

Accumulators: These are temporary storage used during calculations. In arithmetic operation, the accumulators always constitute one of the operands. The accumulators may be used in logical operations, shift and rotation operations. They are used temporarily to store intermediate results in both arithmetic and logical operations. Accumulators are the most widely used registers in computer systems. A computer with one accumulator is slower in operation than that with two or more accumulator. Example is A*B+C*D

SINGLE ACCUMULATOR	**TWO ACCUMULATOR**
Load Accumulator with A	Load accumulator1 with A
Multiply by B	Multiply by B
Store in Temp Register1	Load Accumulator 2 with C
Load Accumulator with C	Multiply by D
Multiply by D, store in temp Reg2	Add the Accumulators
Add temp storage	

General Purpose Registers: Perform a variety of functions. They could be made to serve as temporary storage for data or addresses. They could be made to perform the functions of the program counters or accumulators etc.

Index Registers: They are normally used for addressing. The contents of the Index Registers are added to the memory address which an instruction would use. The sum is the actual address of the data or effective address. The same instruction can be used to handle different addresses by changing the contents of index register. Index registers allow easy transfer of data from one place to another. Also, data are easily accessed in array memory.

Condition Code or Status Registers: These are indicators (flags) that represent the state or pre-condition inside the computer system. The flags are the basis of decision making. Different computers have different number and types of flags based on their area of application and

technology. Among the common flags that exist are carry, zero, overflow, sign, parity, interrupt enable etc. Newer computers have several flags.

The common ones are explained as follows:

CARRY-1: If the last operation generated a carry from the significant bit. The carry flag can retain a single bit of information and the carry will then be handled from one word to the next as in multiple arithmetic.

ZERO-1: If the result of arithmetic operation produces zero its flag bit indicates 1

OVERFLOW-1: If the last operation produced a twos complement overflow. The overflow bit determines if the result of an arithmetic operation has exceeded the capacity of a word.

SIGN-1: If the most significant bit of the result of the operation was negative, then the sign flag indicates 1. It is useful in arithmetic in examining the bits within a word.

PARITY-1: If the number one bits in the last operation was even (even parity), bit 1 is indicated otherwise or it indicates bit 0 (odd parity). It is useful in xter manipulation and communications.

Parity- a redundant bit added to others to detect a bit inaccuracy

HALF CARRY-1: If the last operation generated a carry from the lower half word.

INTERRUPT enable-1: If an interrupt is allowed, otherwise 0 if not.

Stack Pointer: It's a register that contains the address of the top of the stack.

ARITHMETIC UNITS

The arithmetic part of a control section varies from a simple adder to a complex unit that performs many arithmetic and logical functions. Modern computers have binary adder to produce a binary sum and carry bit. Subtraction operation is being performed by using two's complement form while multiplication and division are performed by repeated additions and subtraction. Extra circuitry can form other status such as a zero indicator.

Single-Chip Arithmetic Logic units (ALUs) consist of a binary address and other logical circuitry.

ALU performs both arithmetic and logical operations such as:

-Addition -Subtraction

-Division -Multiplication

Logical operations such as:

-Logical (inclusive OR –complement)

-Increment (add) - decrement (subtract)

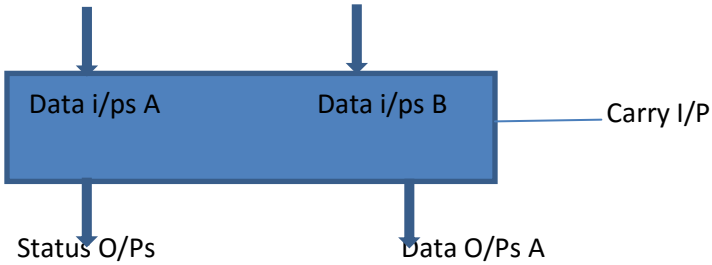


Fig3 An Arithmetic-Logic Unit

Instruction Handling Areas

The CPU must translate the instruction it obtains from memory into control signals that produce the desired actions. For example with instruction: Add R1 and R2 and place the result in R3. If the computer is organized as-----

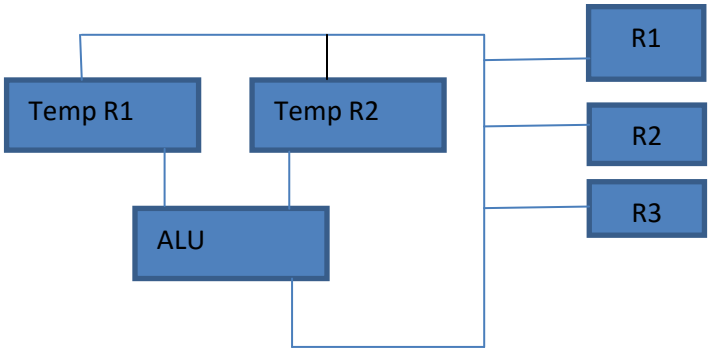


Fig 4.

From Fig 4 above, then (1) The contents of R1 are placed in Temp R1

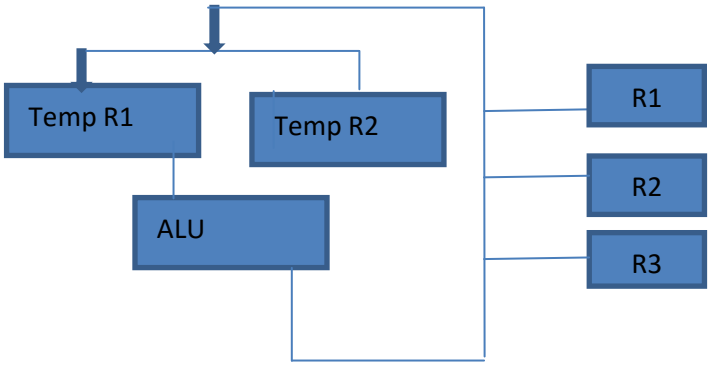
(2) The contents of R2 are placed in Temp R2

(3) The addition is performed and the result is placed in R3

The CPU must also obtain the instruction from memory, place it in the instruction register and prepare to fetch the next instruction.

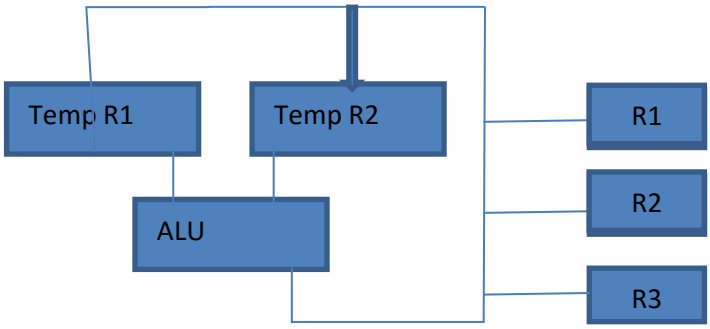
The CPU will hence perform an instruction cycle as follows:

- The CPU places the program counter (PC) on the memory address bus in order to fetch instruction.
- The CPU increments the PC so as to be ready to fetch the next instruction.
- The CPU takes the instruction from the memory data input/output bus and places it in Instruction Register
- The CPU decodes and executes the instruction.



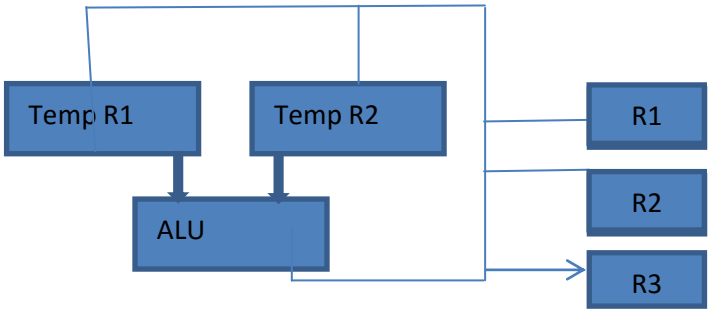
(a) content R1 to temporary Register 1

Fig 5 (a)



(b) content R2 to temporary Register 2

Fig 5 (b)



(c) From ALU to Register 3

Fig 5 (c)

The figures 5 show the demonstration of execution of the instruction ADD R1 & R2 and place the result in R3

THE MEMORY

The central memory units store programs currently being executed by the computer and firmware. 'Firmwares' are permanently stored programs which are necessary for the user to operate the computer systems. In microcomputers, two types of memory devices are employed .

ROM (Read Only Memory): is a permanent storage which can only be read i.e program instruction can only be copied from ROM memory. No information can be written into ROM. Program instructions are 'burnt' into its store during manufacturing or possibly by the user.

It is employed to store permanently required program, the firmware. For example, monitor programs which control the operation of the microcomputer system and allow users to run applications, to input and output data, to examine and monitor main memory and dedicated programs such as the control of industrial design.

ROM is nonvolatile i.e it does not loose information when there is no power supply. Among several uses of ROM are program storage function tables, clock generating, clock conversion table, character generating tables, mathematical or unit conversion, linearization tables, random functions generators, division and multiplication tables, instruction or diagnostic decoder, pattern generators, diagnostic messages, and teaching aids.

RAM (Read/write Access Memory)

It is normally called Read/Write memory; it is designed to have information written into it and read out of it. In a RAM, each word is accessed or retrieved in a given amount of time. It is accessed randomly in the same amount of time i.e. the access time is independent of the position of the word in a memory. It can be used for storing application programs and for storing intermediate result during program execution. Information stored in RAM can be read or modified. RAM is volatile because data stored in it is lost if there is no power supply.

DRAM: Dynamic RAM

DDR-DRAM: Double Data Rate Dynamic RAM- A type of DRAM that transmit data twice clock cycle.

DDR2: Double data rate level2

DDR3: Double data rate level 3

DDR2/3 Supports faster bus speed.

Advantages of ROMS & RAMS are:

For ROMs

- ROMs are non- volatile
- ROMs are cheaper than RAMs
- The contents of ROM are always known and can be verified
- ROMs are easy to test
- ROMs are more reliable than RAMs because their circuitry is simpler
- ROMs are static and do not require refresh
- ROMs are easier to interface than RAMs
- ROMs cannot be accidentally changed.

For RAMs

- RAMs can be updated or corrected
- RAMS can serve as temporary data storage
- RAMS do not require lead time like ROMs
- RAMs do not require programming equipment like PROMs

Programmable Read Only Memory (PROMs): The only difference between PROM & ROM is that users and manufacturers determine their contents.

Erasable PROM (EPROM): Its contents are lost through exposure to high intensity short wave ultra-violet light between 10 and 20 minutes after being removed from the circuit board.

Electrically Alterable ROMs (EAROMs) can be reprogrammed electrically without removal from the circuit board unlike **EPROMs**. Its contents can be changed in a matter of milliseconds. It has the features of being nonvolatile or nondestructive read out.

Specific features of Microprocessor Architecture

This is concerned with the registers, arithmetic units and instruction decoding mechanisms of microprocessors.

Microprocessors registers:- It differs from those of larger computers for the following reasons

1. Limited chip size: a single microprocessor therefore cannot save addresses or data in program memory
2. Use of read only program memory
3. Limited read/write memory
4. Short word length: A memory address may occupy several data words
5. Interrupt driven operations

The following are the architectural features of microprocessors:

They have several general purpose registers. For example, the fair-child F-8 has 64 GPR intel 8080 has 6 etc expect a few like Motorola 6800 which has none.

Almost every microprocessor has a single accumulator

Almost all microprocessors have a stack for saving subroutine return address.

They have special features which involve the use of different set of registers during the interrupt service e.g. Intel 4040 and signetic2650 have this feature. . Microprocessors with only a few registers eg (Motorola6800) can respond to interrupt quickly since they have little to save.

The short words of length of microprocessors make the handling of addresses difficult.

Registers may alleviate this problem in the following ways:

Varied register length: often some registers are longer than the normal word length of the processor particularly program counters memory, address registers, stack pointers, index register and return address; stacks. Special instruction may be available to load and manipulate their contents.

Most microprocessors have an arithmetic unit with a simple bus structure. Generally, one operand is an accumulator and the operand is a temporary register; the result is sent to the accumulator.

Many microprocessors have special read only memory or their circuit to perform Binary Coded Decimal addition and BCD subtraction.

Question: (a) Explain what do you understand by Computer Architecture and Micro-processor Architecture.

(b) (i)What is Register? (ii) Name four registers you know. (iii) Where is Accumulator normally located in a computer system? Describe the function of Accumulator.

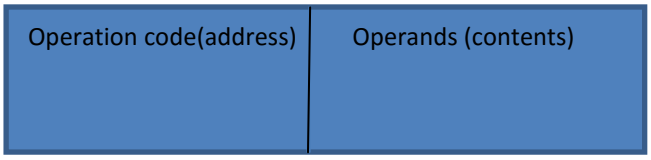
(c) Give Reason or reasons why registers are always very few in computer systems?

(c) Show by display or demonstration the execution of the instructions in the ALU: ADD R1 and R2 and place the result in R3

INSTRUCTION CODES

The internal organization of a digital system is defined by the sequence of micro operations it performs on the data stored in its registers. The internal structure of a microprocessor is called its architecture and includes the number lay out and functionality of registers, memory cell, decoders, controllers and clocks. A digital general purpose computer capable of executing various micro-operations and in addition must be instructed on the sequence of operations it must perform by means of a program. Each program instruction specifies the operation and the operands. A program instruction is just a binary code that specifies a sequence of micro-operation. Instruction codes and data are stored in memory. The processor reads each instruction from the memory and places it in the Instruction Register(IR). It then interprets the binary code of the instruction and proceeds to execute the instruction by issuing a sequence of control functions.

Each instruction is usually divided into parts or fields, called the instruction format.



The operation code (op code) is a group of bits that defines such operations as add, subtract, multiply, shift, complement etc. The relation between an operation and a micro-operation is that an operation is part of the instructions stored in computer memory. It is the binary code that tells the computer to perform a specific operation. The processor reads the instruction from memory and interprets the op code bits. It then issues a sequence of command functions (micro-operation) needed for the hardware implementation of the specified operation. For every operation, there is a sequence of micro-operation. Because it specifies a set of micro-operations.
The operand(s) specifies the data or location of the data in memory or processor register.
There are many variations for arranging the binary code of instruction (i.e different instruction formats) and each computer has its own particular instruction code format. Instruction format are governed by computer designers who specify the architecture of the computer.
Typical 4-address Machine

i. Op code (Op)
ii. Address of 1st operand (#1)
iii. Address of 2nd operand (#2)
iv. Address of Result (R)
v. Address of next instruction (N)

A machine with this type of instruction is referred to as a 4-address machine. The first type of machine designed was the 4-address machines. Because of wastage of storage, the 4-address machine became obsolete and the 3-address machine, 2-address, and finally zero-address machine were developed.

(2) 

3-address machine instruction format

(3) 

2-address machine instruction format

(4) 

1-address machine

(5) 

Zero-address machine (Stack machine)

THREE ADDRESS INSTRUCTIONS:

Computers with 3-address instruction formats can use each address field to specify either a processor register or a memory operand. The program in assembly language that evaluates **X= (A+B) (C+D)** is shown below:

ADD R1 ,A, B R1 ← M (A) + M (B)
ADD R2, C, D R2 ← M(C) + M (D)
MUL X, R1, R2 M (X) ← R1 X R2

The symbol **M (A)** denotes the operand at memory address symbolizes by A.
The advantage of the 3-address format is that it results in short programs when evaluating arithmetic expression. The disadvantage is that the binary-coded instructions require too many bits to specify three addresses. An example of a computer using three-address format is the Cyber 70 computer.

TWO-ADDRESS INSTRUCTIONS:

Two address instructions are most common in commercial computers. Here each address field can specify either a processor register or a memory word

The program to evaluate X= A+B) * (C+D) is as follows:

MOV	R1, A	R1 ← M(A)
ADD	R1, B	R1 ← R1 + M(B)
MOV	R2, C	R2 ← M(C)
ADD	R2, D	R2 ← R2 + M (D)
MUL	R1,R2	R1 ← R1 * R2
MOV	X, R1,R2	M(X) ← R1

The MOV instruction moves or transfers the operands to and from memory, and processor registers. The first symbol listed in an instruction is assumed to be both a source of the operation is transferred.

An example of a commercial computer that uses two-address instruction is PDP-11 (a mini computer)

ONE-ADDRESS INSTRUCTIONS:

One-address instructions use an implied accumulator (AC) register for all data manipulation. For multiplication and division, there is a need for second **register** in the following, however we will neglect the second register and assume that the AC contains the result of all operations. The programme to evaluate: X = (A+B) * (C+D) is

LOAD A	AC ←	M(A)
ADD B	AC ←	AC + M(B)
STORE T	M (T) ←	AC

LOAD C	AC ←	M(C)
ADD D	AC ←	AC + M (D)
MUL T	AC ←	AC * M (T)
STOR X	M (X) ←	AC

All operations are done between the AC register and a memory operand. In the address of a temporary memory location required for storing the immediate result. An example of a computer that uses one address instruction is PDP-8

ZERO-ADDRESS INSTRUCTIONS

A stack organized computer does not use an address field for the instructions ADD and MUL. The PUSH and POP instructions however need an address field to specify the operand that communicates with the stack. The following program shows how X = (A + B) * (C + D) will be written for a stack-organized computer. (TOS means Top Of Stack)

PUSH A	TOS ←	A
PUSH B	TOS ←	B
ADD	TOS ←	(A + B)
PUSH C	TOS ←	C
PUSH D	TOS ←	D
ADD	TOS ←	(C + D)
MUL	TOS ←	(C+D) (A+B)
POP X	MX ←	TOS

In order to evaluate arithmetic operations in a stack computer, it is necessary to convert the expression into reverse polish notations. The name zero-address is given to this type of computer because of the absence of an address field in the computerization instructions. An example of a stack-organizes computer is Burroughs-6700.

COMPUTER ARCHITECTURE

There are two main types of architecture

4,3,2 & 1 address machine form the Neumann architecture	Zero address machine form the stack machine

The most general representation of a digital computer consists of four basic units

i) I/O
ii) Memory
iii) Arithmetic & Logic Unit (ALU)
iv) Control

All general purpose computers form microcomputers to large scale multi-processing systems can be described in terms of these basic units. The particular design and interaction between these four units is called the architecture of the machine.

Computer architecture encompasses both hardware and software structures. One describes the interaction between static hardware design and the other describes the dynamic software processing.

As stated above, the two most practical architecture are:

i. Von Neumann architecture
ii. Stack architecture

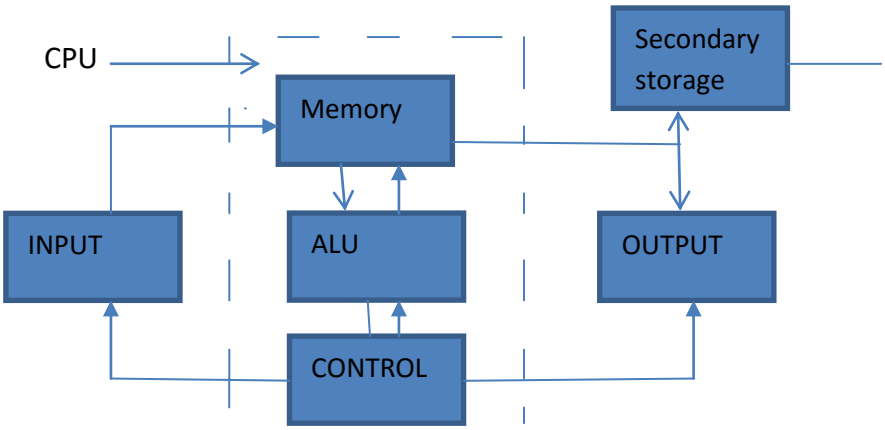


Fig6 Bloch diag. of a General purpose computer

VON NEUMANN'S ARCHITECTURE

This is the most commonly used in digital computers. It specifies the type of interaction between the control and the memory. The control section operates on the basis of a sequence of instructions called a program which is stored in memory. Each instruction consists of two parts: an op-code and operand.

The Von Neumann computer operates by executing a sequence of instruction that correspond to the operation to be performed by the system. Since there is a lot of interaction between the control unit and memory, special hardware structures are provided to speed up the data and instructions transfers,

These include:-

Instruction Register (IR)

Programme Counter (PC)

Instruction Decoder (ID)

Accumulator (AC)

Arithmetic & Logic Unit (ALU)

Major State Register (MSR)

Flag Register (G)

These are depicted in Fig 7 below as CLASSICAL DIGITAL COMPUTER ORGANIZATION.

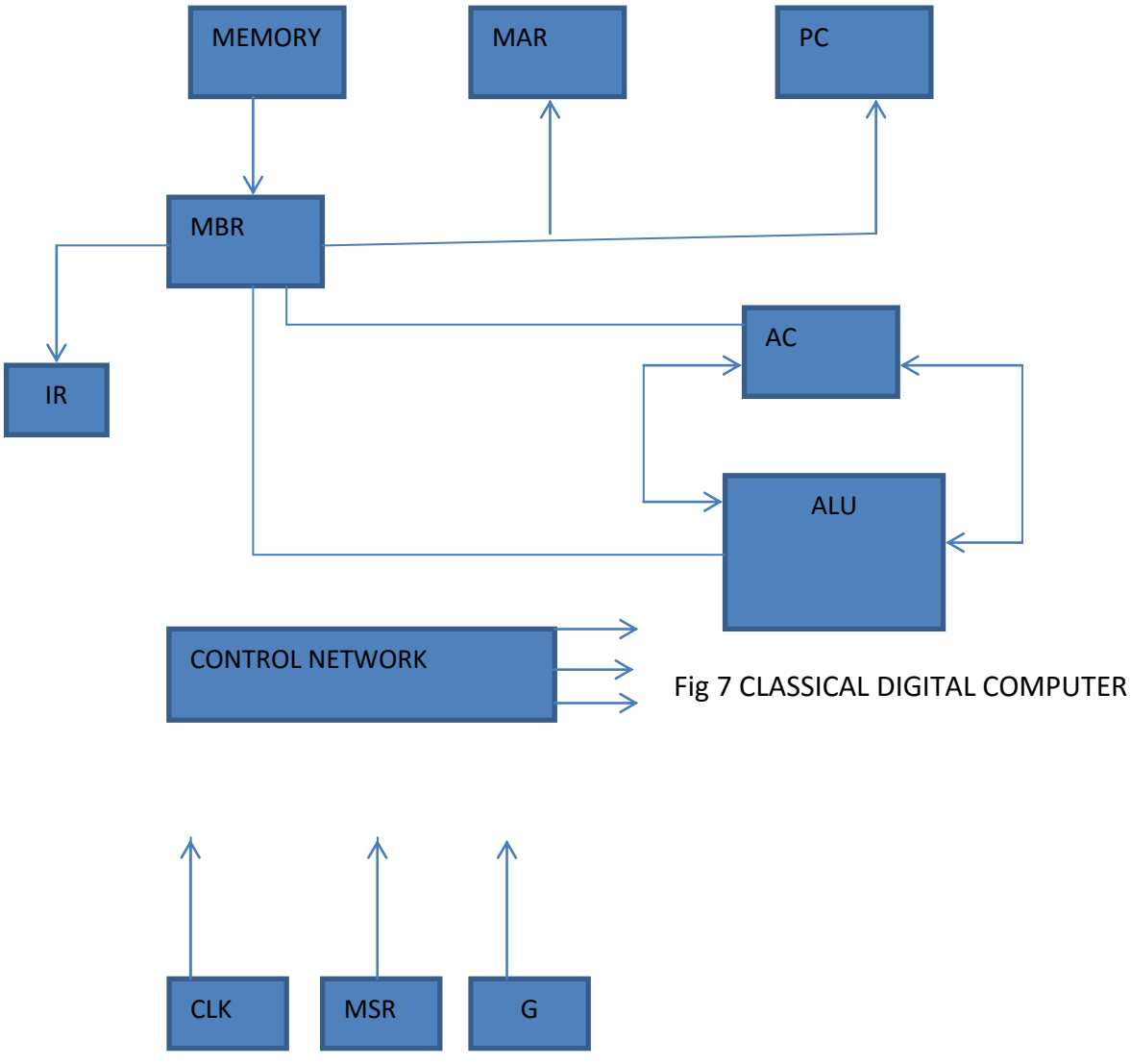


Fig 7 CLASSICAL DIGITAL COMPUTER

The IR stores the next instruction to be executed by the computer. The Program Counter (PC) stores the location in memory of the next subsequent instructions to be executed by the computer. The Accumulator (AC) stores data that are to be processed or accepts new data after being processed. The ALU performs the specified arithmetic and logic calculation on predetermined data indicated by the instruction.

These hardware structures are connected to each other by means of one or more buses. A bus is an electrical connection between several points which may act as sources of links for signals. There are 3 types of buses, namely:

- Data bus
- Address bus
- Control bus

A Bus is a data high way between CPU and other peripherals eg memory & i/o devices.

The bus connects these registers and the ALU with the memory and the i/o devices. The control unit performs the basic supervision and synchronization of all other units of the computer. Computer systems are synchronized on the basis of a standard clock signal that is provided throughout the system.

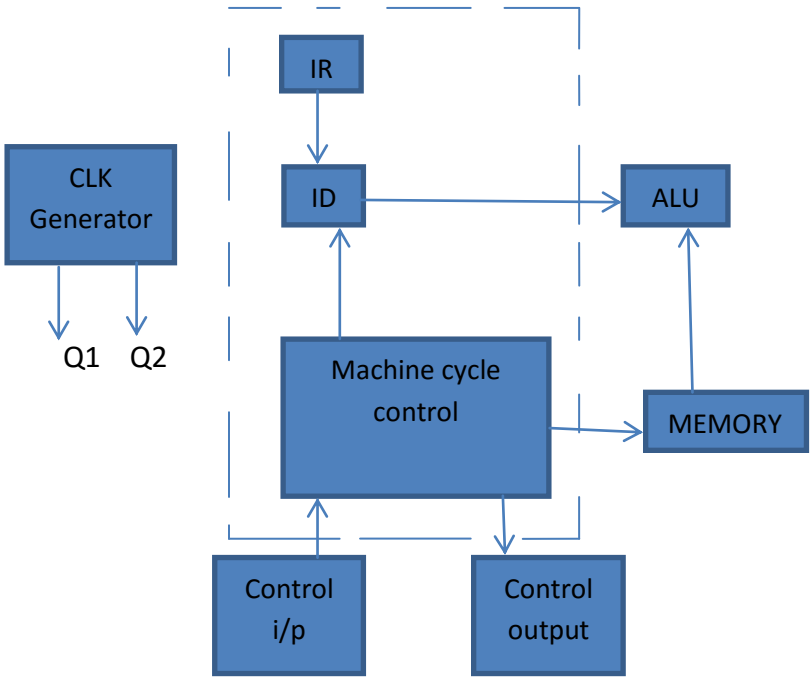


Fig 8 Von Neumann architecture

A clock generator is shown having output Q1 & Q2. The machine cycle control unit, based on the instruction indications from, the instruction decoder (ID), provides external indications of the type of instructions being executed through the control outputs, as well as the memory. This other system component can be synchronized with operation of the ALU.

Computer architecture is the same as computer system organization. Computer organization refers to the ways in which the various physical components of a computer system are connected together.

STACK ARCHITECTURE

The architecture of a computer may also be organized around the user language in which it is programmed. There are numerous variations of such architecture or "compactible architecture". Presently, the only one that is already commercially available is the stack architecture . It is called stack architecture because it uses Last-In-First-Out (LIFO) stacks for handling system information. A storage location to only one entry ie one entry is used for entry and exit eg pop & push.

Stack Architecture (SA) was developed to increase the effectiveness of computation by providing a means for the direct execution instruction. A stack is temporary storage facility, either a separate buffer memory, or a specific portion of the main memory that operates to store information in sequence. As new information is entered into the stack, other information in the stack is "moved down" in the stack. As information is removed from the stack, the information is "removed up" in reverse order. This is the last information placed in the top of the stack is the first information out. The concept of Last-In-First-Out is graphically illustrated in the figure below, where push refers to the process of entering information into the stack is graphically illustrated in the figure below, where push refers to the process of entering information into the stack while "pop" refers to removing information from the stack

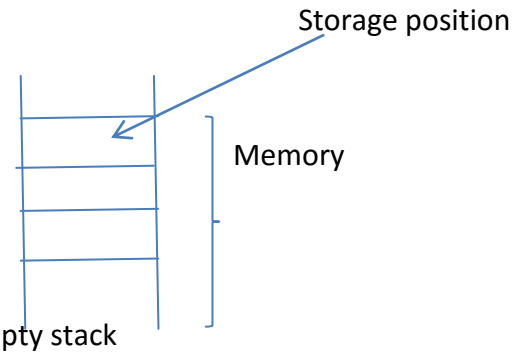
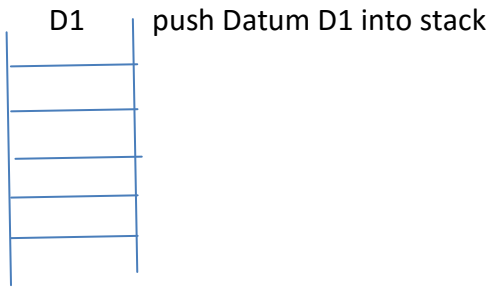
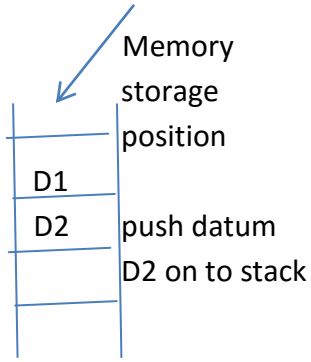


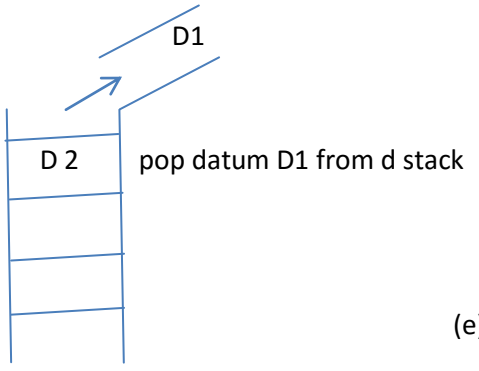
Fig 9 (a) Empty stack



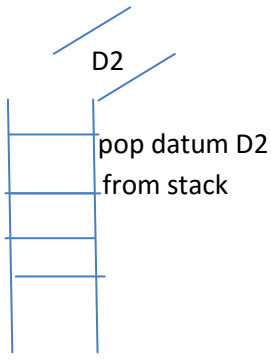
(b)



(c)



(d)



(e)

A register, the stack pointer (SP) always points to the address top word in the stack ie address of the last item inserted in the stack.

In reality, nothing is physically pushed or popped in a computer stack. The operation of push and pop are simulated by incrementing the stack pointer or decrementing the stack pointer register. The organization of a 64-word memory stack is shown below.

The Organization of a 64-word memory stack is shown below

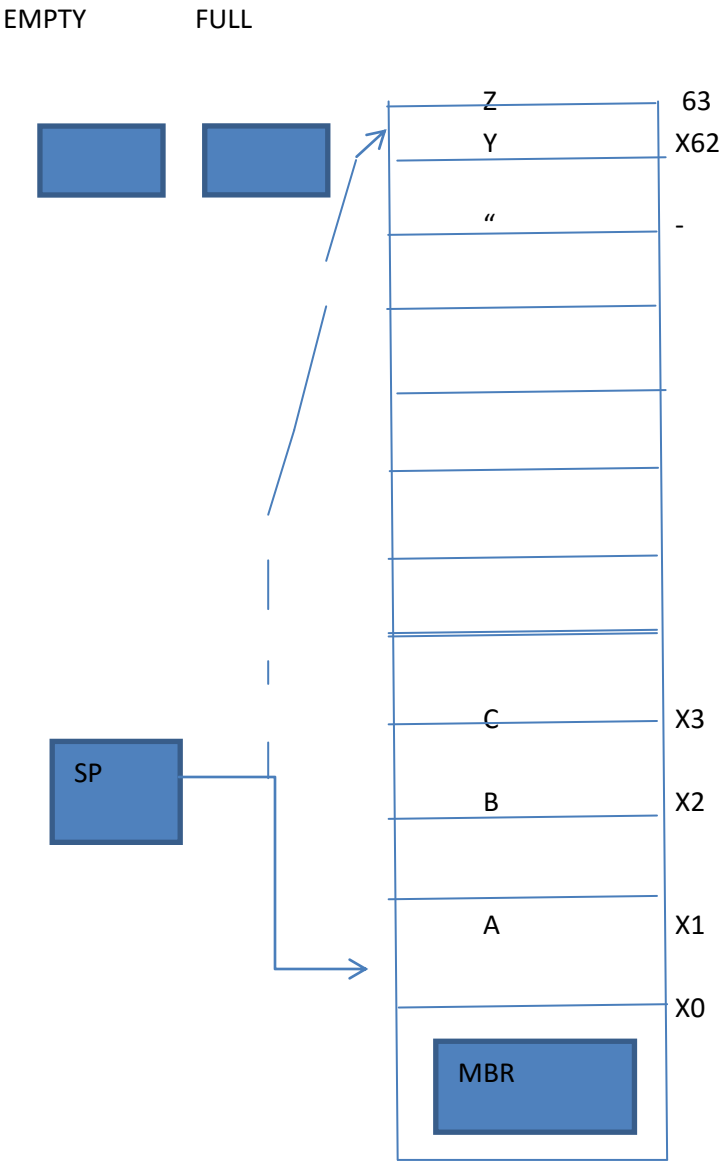


Fig 10. 64 bits memory stack.

In a 64-word memory stack, the SP contains 64 bits since 26=64. The one-bit register FULL and empty are set when the stack is full or empty of item respectively. The MBR (Memory Buffer Register) holds the data to be written into memory stack.

Initially, SP is cleared and empty is set, so that SP points to the word at 0 and the stack is marked EMPTY. If the stack is not full (if FULL= 0), a new item is inserted by a push operation. The push operation is implemented with the following micro-operations.

SP ← SP + 1 : Increment stack pointer

M (SP) ← MBR : write item top of stack

If (SP=0 then (FULL= 1) : Check if stack is full

EMPTY ← 0 : Mark the stack not empty

NOTE: The condition of SP=0 is reached if the last item was in location 63 and by increasing SP, the item is stored in location 0. Once an item is stored in location zero, there are no more empty registers in the stack hence is cleared. A new item is deleted from the stack if the stack is not empty (if EMPTY= 0)

The POP operation consists of the following sequence of micro-operations.

MBR ← M (SP) : Read item from stack top

SP ← M (SP) :

SP ← SP-1 : Decrement SP

If (SP=0) then (EMPTY ← 1) : Check 1s stack is empty

FULL ← 0 : Mark stack not full

Note: If the POP operation reads the item from location 0 and the SP is decremented, SP change to all 1's which is equivalent to binary 63.

The two operations for either PUSH or POP stack are:

1) An access to memory through SP
2) Updating SP

STACK PROCESSOR

The stack processors are used more efficiently to execute higher level language code (eg assembly language). A stack is a special linear list in which insertion and deletion operations are restricted to occur only at one end, refer to as the stack's top.

A linear list is a data structure comprised of an ordered set of elements. The no of element in the list can vary. The two operations that take place in the stack is PUSH (insert) or POP (delete).

APPLICATION OF STACKS

Stacks are used extensively by compilers, operating systems, and application programs in solving variety of problems.

The three uses of stacks are as follows:

1. Matching parenthesis
2. Recursion
3. Postfix notation.

CONTROL UNIT

The function of the control unit in a digital system is to initial sequence of micro-operations. To execute instructions, the CPU must have some means of generating appropriate control signals to initiate the next step in the correct sequence. When the control functions are generated by hardware using conventional logic design techniques, the control unit is said to be hardwired . An alternative is micro-programmed control (software).

SYNCHRONIZE AND ASYNCHRONIZE PROCESSORS

The total time required to execute an instruction depends on the instructions itself. The fetch face is the same for all instructions but the execute phase may require one or more steps. Also the time for each step of the fetch is not necessarily the same as each is just being indicated. The time for a memory transfer will be relatively large compared to the time for an operation on an internal registers. Also operations involving simple register transfers such as those which

occur in executing a BRANCH instruction will take less time than those operations on registers requiring Arithmetic or logic operation.

The timing for each step can be achieved in one of two ways below:

SYNCHRONIZED PROCESSOR

Has an internal processor clock. This is an electronic circuit that generate electronic pulses at regular and accurate intervals of time and usually based on an **crystal control oscillator** for access and stability. Such step must commence operation on a clock pulse and although each transfer step always requires the same amount of time each time it is performed, some steps may require more than one clock period to complete. This lead to a relatively simple processor construction but the disadvantage of it is that not all steps need the same amount of time. Some operations cannot commence until the next clock pulse, even though the preceding step is complete.

ASYNCHRONIZED PROCESSOR

An **asynchronized processor** is one where initial of the next steps takes place immediately the previous step is completed. This will prevent idling the processor as it waits for the next of clock pulse and consequently will result in an increase in the speed of the processor. However, this is tempered by the fact that extra logic circuit make it more expensive (because of the logic circuit to detect the end of a step, will usually be greater than simple clock) but also the fact that the end of an event has to be detected will take sometimes and reduce the time saved in synchronized processor.

Asynchronized processor of the CPU is however gradually faster but more complex and costly than synchronized operation because more hardware is required.

INTER CONNECTION METHODS (IN THE CPU)

The CPU is made up of a number of component parts. These components parts can be interconnected in several ways. The way the components are connected can have a significant effect on the hardware, instruction that can be provided and at the speed of CPU. Consequently, it is one the architectural features that can distinguish one machine from the other.

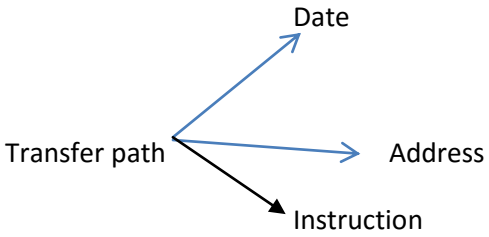


Fig 11 Transfer path.

CPU connection can be implemented in one of the 3ways either

1. Using point-to-point connection (PP)
2. A common bus
3. A multiple bus system

N.B: System interconnection scheme is different from CPU interconnection scheme.

1. POINT TO POINT CONNECTION (PP)

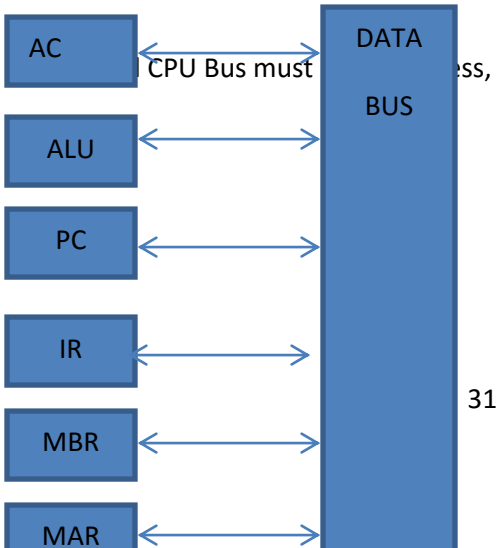
If a computer is to achieve a measurable speed of operation it must be organized in a parallel version. This means that in order for a component to handle a full-word (say 16 bits) of data at a time, the data transfer between components must be done in parallel. (This means there should be 16 busy lines) which implies that a considerable number of wires are needed for the necessary connection.

In the case of point-to-point bus system, every information transfer path required is provided with a dedicated bus. In fig 11 above, each data flow path would be implemented in an individual bus.

ADVANTAGES: Many transfers would be taken place simultaneously thereby tending to lead to a fast CPU.

On studying the transfer necessary for the operation of a machine instruction, it can be seen that a lot of transfers cannot logically take place simultaneously even though physically it is possible. Consequently a full point-to-point bus system for internal CPU organization is almost never used. At the other extreme there would be a common bus system.

COMMON BUS SYSTEM (CBS)



CPU Bus must
ess, data &
Control

Fig12 COMMON BUS SYSTEM

Implemented with a common bus system, the CPU whose data flow paths are shown in fig 11 would be as shown in fig 12.

All data transfers take place along a common bus to enable information to be transferred from one register to another. There need to be some logic pulse (switches on or off) which enable the required register to be actively connected to the bus at the appropriate time.

ADVANTAGE : Inexpensive

DISADVANTAGE: Only one transfer can take place at once. There4, no need for concurrent operation. Consequently this type of computer can be very slow. The most usual form of internal CPU Bus system on many machines is the MULTIPLE BUS SYSTEM.

MULTIPLE BUS SYSTEMS (MBS)

MBS is a compromise between the 2 extremes just described. Here there is more than one bus to which a number of component are attached. Sometimes, some registers are connected by point-to-point bus, if necessary. Often the data flow requirements are implemented with 2 buses- An address bus and a data bus.

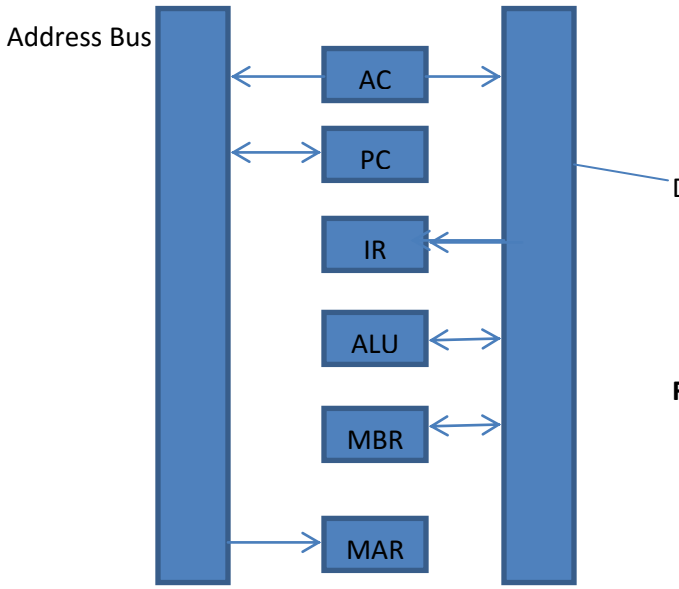


Fig 13 MULTIPLE BUS SYSTEM

For the machine described in Fig 13 implemented as a multiple bus machine. The structure might be as in fig 12. This system allows more than one transfer simultaneously. Although allowing one transfer on each bus at any time.

A 2-BUS STRUCTURE

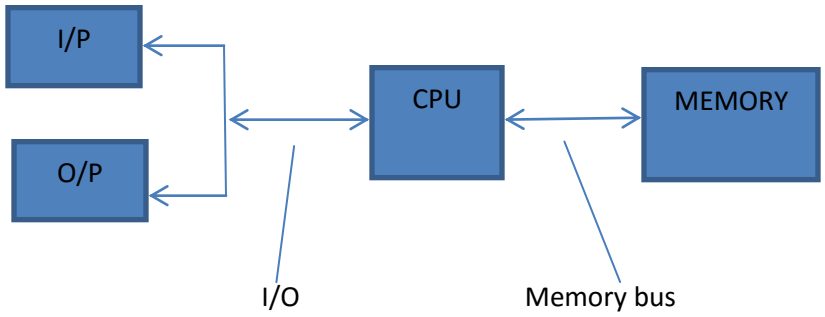


Fig 14 Shows the simplest form of a 2-bus structure computer . Here the CPU interacts the memory via memory Bus. i/o functions are handled by means of an I/O bus, so that data passes through the CPU enroute to the memory. In such a configuration, the I/O transfers are usually under direct control of CPU. It initiates the transfers and monitors the progress unit completion. A commonly used term to described this kind of operation is programmed I/O.

The presence of I/O processor acts as an I/O for CPU in transferring data from and to memory and to CPU. This aids the CPU IBM 360/270 has this structure.

CONTROL UNIT

To execute instructions, the CPU must have some means of generating appropriate control signal to initiate the next step in the correct sequence of events. The various ways that have been developed fall into one of 2 categories:

1 Hardwired control
2 Micro-programming control

HARDWIRED CONTROL

Consider the sequence of steps involved in the fetch/wait execute cycle as discussed previously. Each step will require a time period, defined by the clock. If the processor is a synchronous one. The particular action at each step is defined by the state of in built clock system.

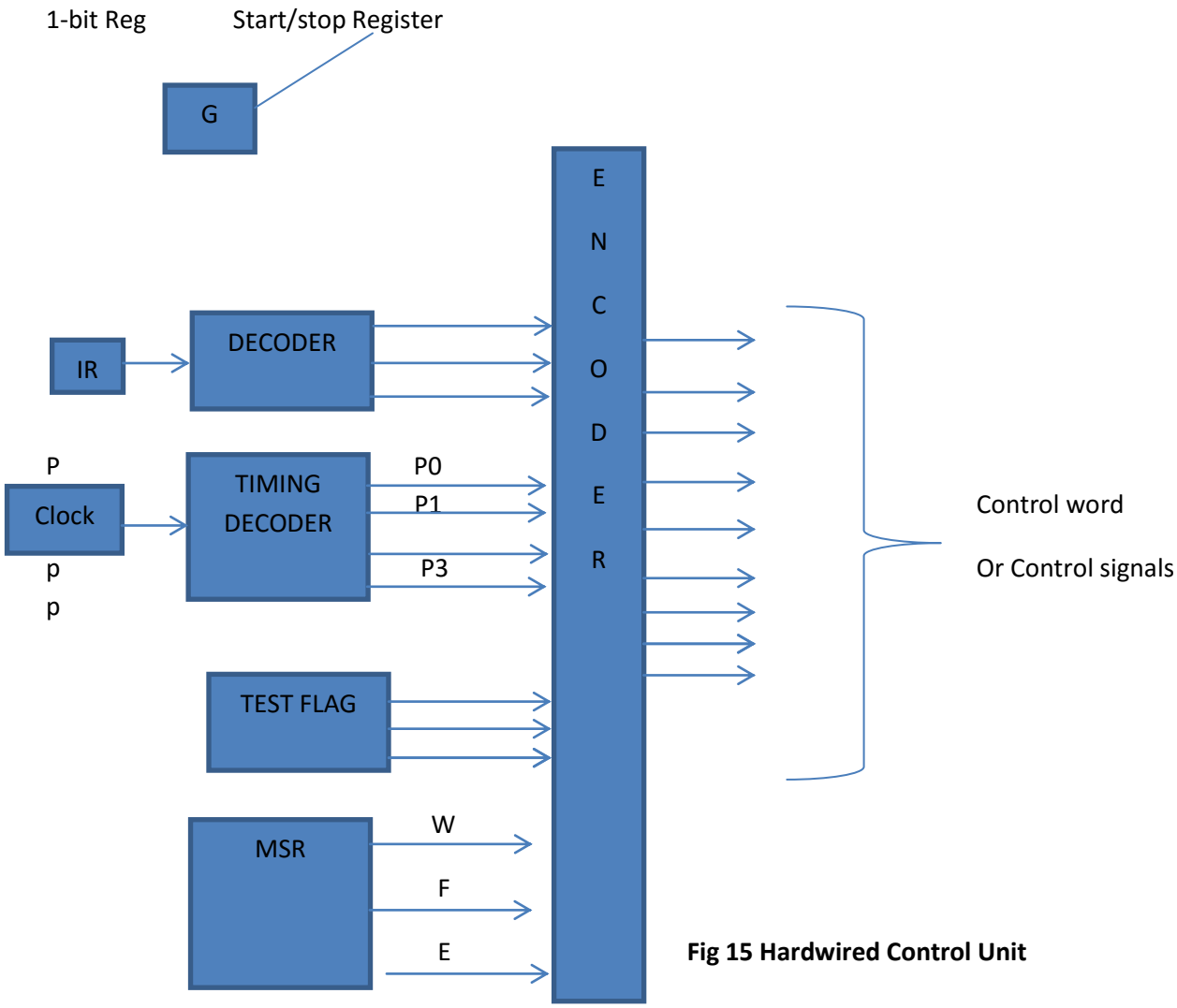


Fig 15 Hardwired Control Unit

W = Wait, F = Fetch, E = Execute, IR = Instruction Register MSR = Major State Register

(a).The output of the decoder. This defines the machine instruction to be executed
(b).The clock relative to the time period at the start of this machine instruction.
(c).Any test flags set the ALU
(d).The output of the MSR: wait, fetch, execute.
(e).The output of the G (GO/Wait) register

The particular operation carried out by the CPU is controlled by a complex logic circuit known as ENCODER (The encoder is a sequential cct). This is illustrated in Fig 15, it has input, a series of lines from the decoder, clock, ALU, test flag, MSR and the G-Register. Only 1 line from the decoder will have a signal on it indicating which step period is the

current one. Only those lines from test flags set will have signals on them. Only 1 line from the MSR will have signals on it indicating the current state of the machine.
Each output line corresponds to a particular control signal. A combination of which will result in one of the transfer steps that the CPU is capable of performing.
All registers are connected to a bus system to allow information to pass between appropriate registers.
Connections to the bus systems are through gates (A gate is normally inhibited but can be enabled by the application of a control signal)
Thus to transfer into or from one register to another, the output gate of one register and the output of the other register will both need to be enabled requiring 2 control signals

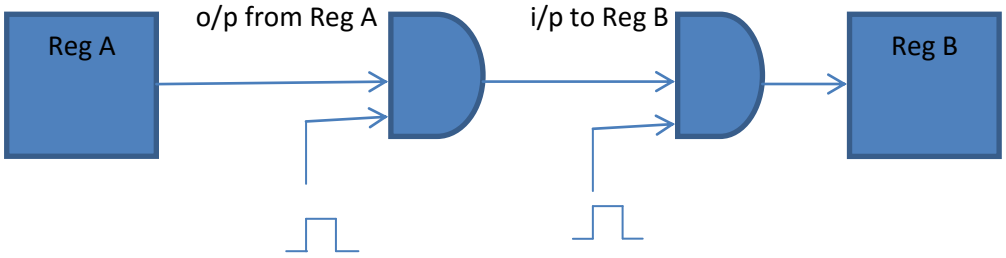


Fig 16

According to the combination of input lines to the encoder, with signals on them, only the appropriate output lines will generate control signals which cause that particular step to be performed.

MICRO PROGRAMMED CONTROL

An alternative way of generating the appropriate control signal at the appropriate time is by a software technique known as micro-program and control.

The execution of an instruction is similar to that of a program, each is made up of more elementary steps which together accomplish the objective of the instruction. Each machine instruction may be implemented using a process very similar to programming

- Within the executing control that which can switch on and off of a data path in the CPU by sending pulses on various control lines. Line 1 for example controls the path of PC to MAR, line 2 causes a memory read when activated by a pulse. To execute any particular instruction, it is only necessary to send pulses. Sequentially to the relevant control lines while the control lines not involved in the particular operation receive nothing.
- This method of implementing the effect of machine instruction by constructing a string of control pulses

is called micro-instruction. Each line is called a micro-instruction or micro-program control and a whole instruction set is implemented by writing a series of micro-programs and storing therein a control store, micro-program memory inside CPU.

Simply put, a control unit whose micro-operation steps are stored in a memory is called a micro-instruction programmed control unit. Each control word of memory is called a micro-instruction and a sequence of words is called a micro-program.

N.B: Micro-programs are usually stored in a ROM since control micro-program do not need to change the use of ROM makes the control words permanent during production.

When a particular instruction is to be executed, the execution control unit goes to the relevant input-output-control of the control store, reads out a sequence of micro instructions and sends 1's and 0's and on the data traffic lines word by word. No complex control unit needs to be constructed for the execution control unit or the instruction decoder as the intelligence has already been written in the micro-program software.

Example

Fetch state operation (Micro instructions need for fetch routine)

F.P0 : MAR ← PC

F.P1 : MBR ← (MAR): PC + 1

F. P2 : IR ← MBR (OP)

MAR ← MBR (AD)

F.P3 MSR ← 2

ADD Operation

E. ADD P1 : MBR ← M (MAR)

E.ADD P2 : AC ← AC + MBR

E.ADD P3 : G = 1 → MSR (MAR) 1 → Fetch state

G = 0 → MSR (MAR) 0 ← Wait

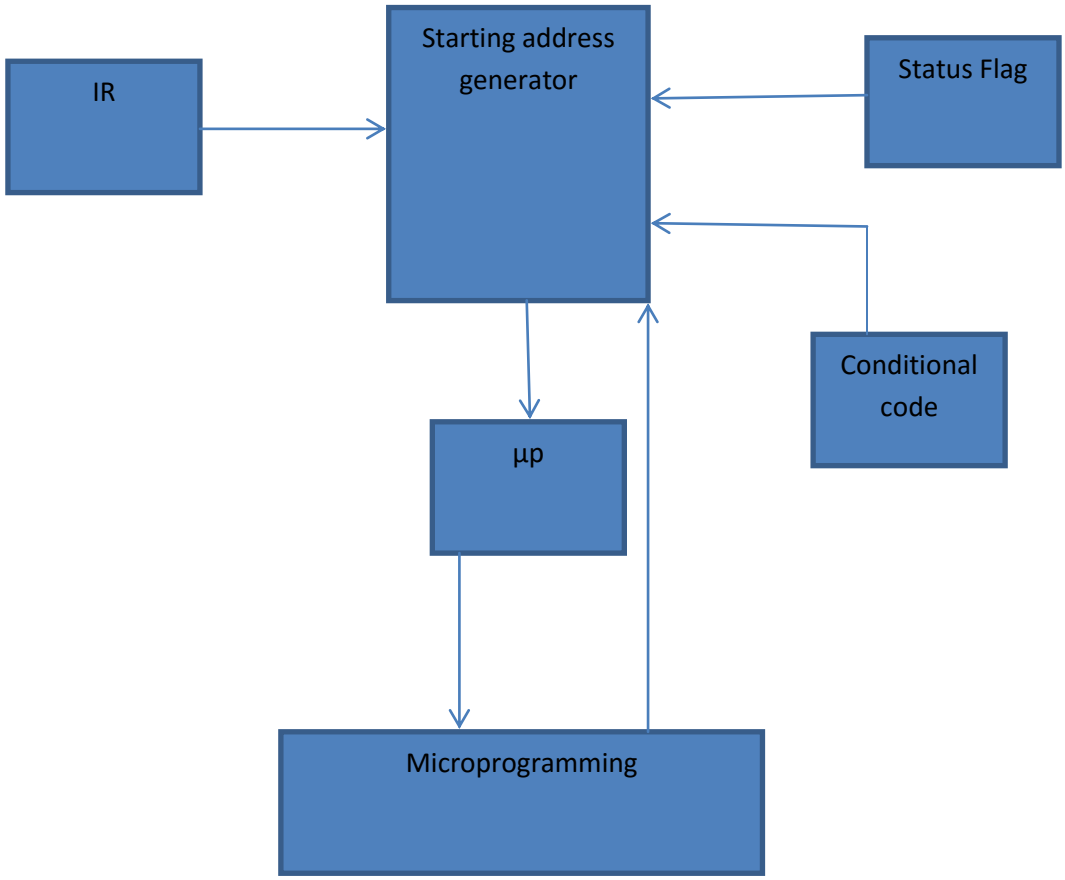


FIG 17 Organization of micro-programmed control unit to enable conditional branching in the micro-program.

µPC = Microsoft program counter

CW = control Word

CM = Control Memory

MM = Main Memory

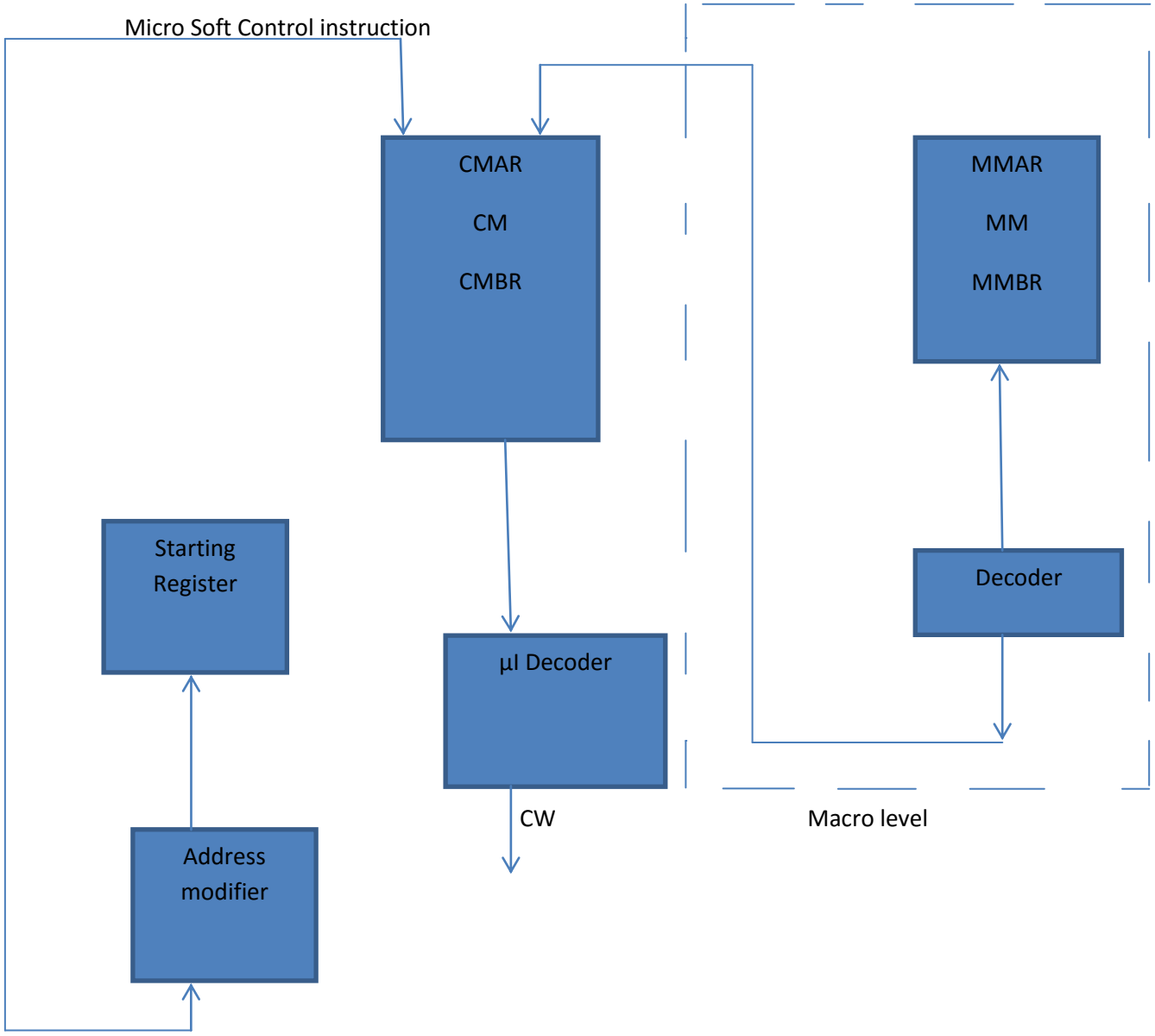


FIG 18 Micro-Instruction programming

ADVANTAGES & DISADVANTAGES OF µPROGRAMING

The use of microprogramming to implement execution control provide certain advantage over the direct construction of h/wired control circuit, largely in the greater flexibility of the approach.

Specifically:

1.Both complex and simple instruction sets may be implemented without H/W differences. A complex instruction Squire root (srt) simply requires larger control store (ROM ic) and more elaborate micro programs
2.Errors in the micro-codes are easier to correct than errors in H/wired design
3.CPU can be upgraded by micro-codes modification rather than hardware replacement
4.A range of CPU performance can be obtained from same or very similar h/w either by using control store or different speeds or by adopting different micro-instruction word size
5.It can emulate other machine
6.Micro-programming can extend the useful life of a computer. When new ideas about machines or instruction set are developed, the target machine(s) in use can be modified to include them by simply changing their micro-program instead of throwing out the old machine by a new one.
7.µ-programmed computers are easier to design and are conceptually simpler than the H/W wired ones.
8.Maintenance is easier.

MODERN COMPUTER ORGANISATION

The classical organization in its specific form is primarily the result of the minimum h/wired requirement in the early days of computer development. In recent years the technological advancement of semi-conductor and magnetics has drastically reduced the cost of digital electronics computer henceforth some wastefulness of h/wired component is perfectly justified in order to :

1. Improve the overall machine performance in speed precisions and versatility

2. Render the design and construction of the machine simple and systematic and thus economical

3. Make users programming and system programming easier and cheaper to prepare per group.

Thus over the years, the basic classical organization has modified in a number of ways to achieve these objectives some of the prominent features of modern design are:

1. Modular design
2. Micro-programmed control
3. Memory hierarchy

MODULAR DESIGN

The modern design employs a number of standardized functional units linked by several data buses to realize a systematic and flexible organization, the functional units may be arranged to operate.

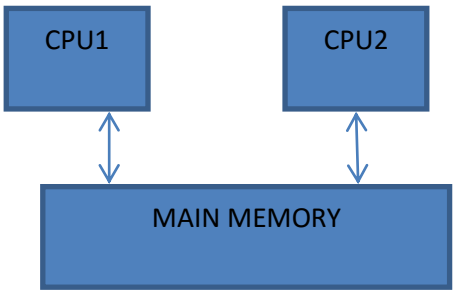
1.Simultaneously (parallel operation)
2.Cascade (pipeline operation)

The overall aim is to increase the effective computing speed by allowing several instructions to be executed at the same time.

1.Parallel computer Memory Architecture

Von Neumann Architecture

For over 40yrs, virtually all computers have followed a common machine model known as the Von Neumann computer which was named after the Hungarian mathematician John Von Neumann. A von Neumann computer uses the stored-program concept. The CPU executes a stored program that specifies a sequence of read and writes operations on the memory. Its basic design includes the following: Memory is used to store both program and data instructions; program instructions are coded data which tell the computer what to do. Data is simply information to be used by the program. The Central Processing Unit (CPU) gets instructions and /or data from memory, decodes the instructions and then sequentially performs them. The simplest way of doing this is to have 2 or more independent CPU's sharing common memory as shown in fig 19 a & b below



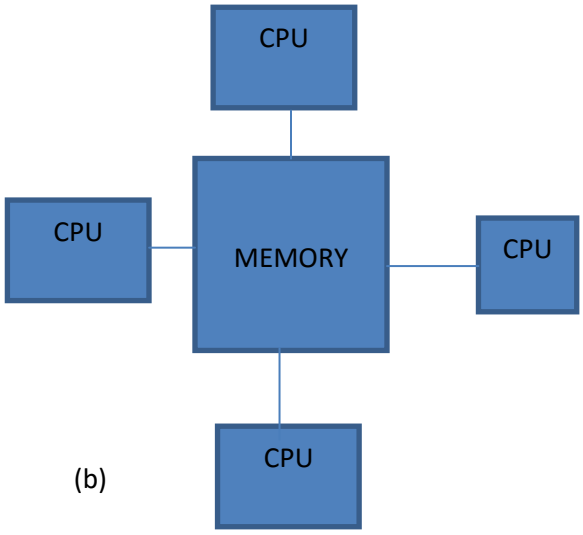


Fig 19 A Multiprocessor system (a} (b)

Chart of shared memory

Usually each CPU has its own program to run with no interaction between CPUs. A computer with (a) multiple processors is called Multiprocessor system or MULTIPLE INSTRUCTION, MULTIPLE DATA STREAM COMPUTER (MIMD Computer)

(b) Some problems require carrying out the same computation on many sets of data (SIMD) eg weather prediction programme used by metrological instruction might read hourly temperature measurements taken from 1000 weather stations and then compute the daily average at each station by performing exactly the same computation on each set of 24-hrly readings since the same program is used on each data set, a processor with one program counter and one instruction decoder built- in-arithmetic units and n-register sets could carry out computations on n-data sets simultaneously

.

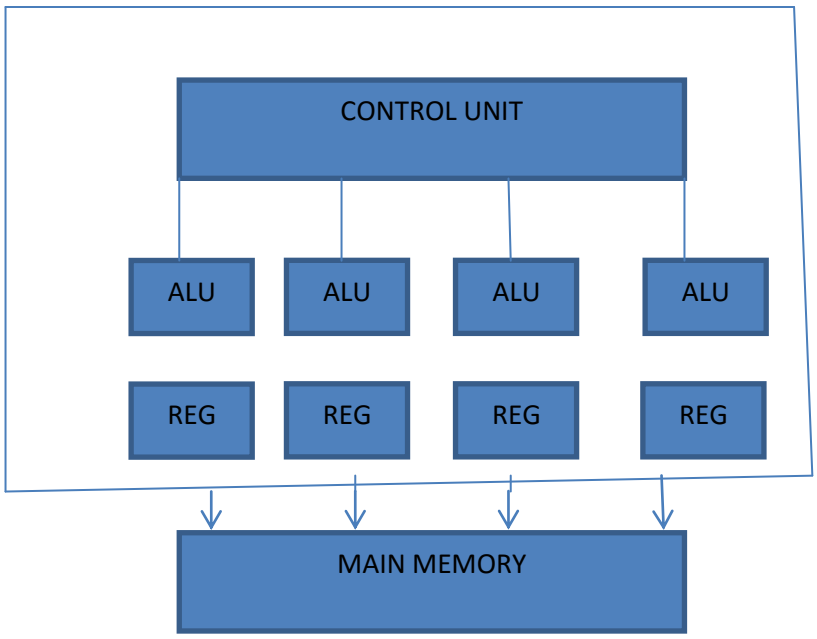


FIG 20 ARRAY PROESSOR (SIMD COMPUTER)

Fig 20 shows an array of processors and its configuration is sometimes called Single-Instruction-Multiple-Data stream (SIMD) processor.

An example of this type of organizations is Illiac 1V designed at the University of Illinois and constructed by Borough Corporation (a US company) . The Illiac iv consisted of 4 control units each of which operates on 64 data sets simultaneously. It could perform 4 different computation at the same time, each computation being carried out on 64 sets making a total 256 calculations in parallel.

Flynn's Classical Taxonomy

There are different ways to classify parallel computers. One of the more widely used classifications, in use since 1966, is called Flynn's Taxonomy. Flynn's taxonomy distinguishes multi-processor computer architectures according to how they can be classified along the two independent dimensions of Instruction and Data. Each of these dimensions can have only one of two possible states : Single or Multiple.

The matrix below defines the 4 possible classifications according to Flynn.

Flynn's Classical Taxonomy

S I S D Single Instruction, Single Data	S I M D Single Instruction, Multiple Data
M I S D Multiple Instruction, Single Data	M I M D Multiple Instruction, Multiple Data

Single Instruction, Single Data (SISD):

This pertains to a serial (non-parallel) computer to which only one instruction stream is being acted on by the CPU during any one clock cycle and only one data stream is being used as input during any one clock cycle. The execution is deterministic.
This is the oldest and until recently, the most prevalent form of computer

Examples: Most PCs, single CPU workstations and mainframes or server.

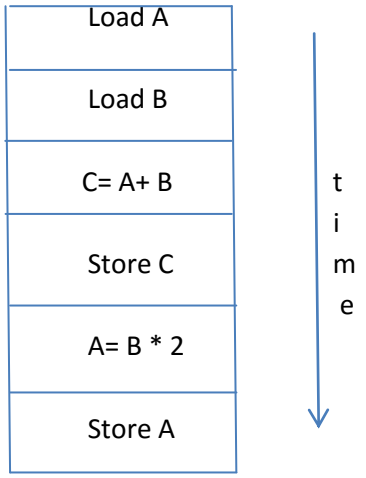


Fig 21 Chart of Single Instruction, Single Data (SISD)

Single Instruction, Multiple Data (SIMD):

This pertains to a type of parallel computer in which all processing units execute the same instruction at any given clock cycle and each processing unit can operate on a different data element. This type of machine typically has an instruction dispatcher, a very high-bandwidth internal network, and a very large array of very small capacity instruction units. It is best suited for specialized problems, characterized by high degree of regularity, such as image processing. It displays a synchronous (lockstep) and

deterministic execution. Two varieties: Processor Arrays and Vector pipelines eg Fig 19 & 20.

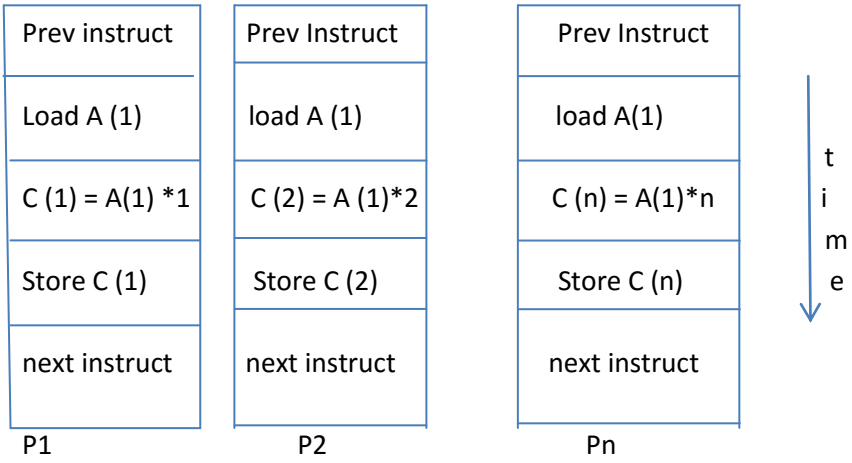


Fig 22 Schematic of Single Instruction, Multiple Data (SIMD)

Multiple Instructions, Multiple Data (MIMD):

Currently, this is the most common type of parallel computer. Most modern computers fall into this category in which every processor may be executing a different instruction stream and every processor may be working with a different data stream. Execution can be synchronous or asynchronous, deterministic or non-deterministic.

Examples: Most current super computers, network parallel computer "grids" and multi-processor SMP computers-including some types of pcs.

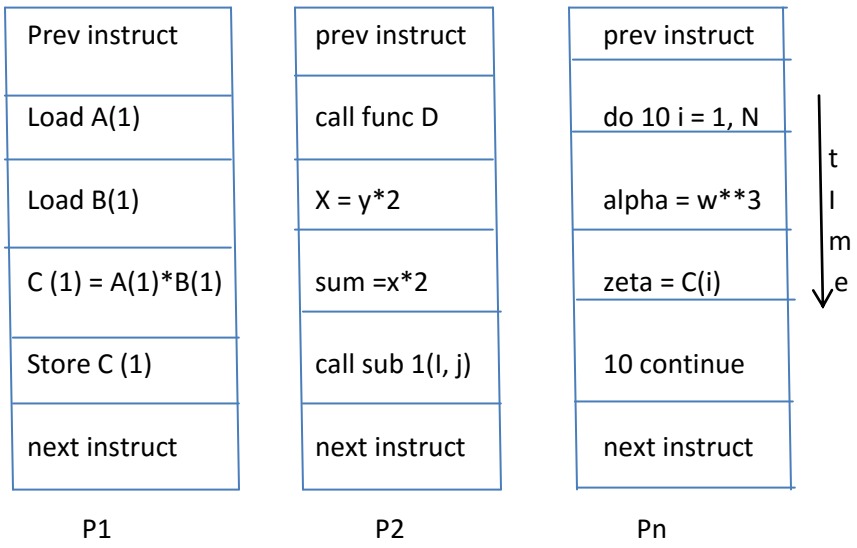


Fig 23 Representative of Multiple Instruction, Multiple Data (MIMD)

Examples: Processor Arrays: Connection Machine CM-2, Maspar MP-1, MP-2

Vector Pipelines: IBM 9000, CraC90, Fugtsu VP, NECSX-2, Hitachi S820

Multiple Instructions, Single Data (MISD): This entails a single data stream being fed into multiple processing unit operates on the data independently via independent instruction streams.

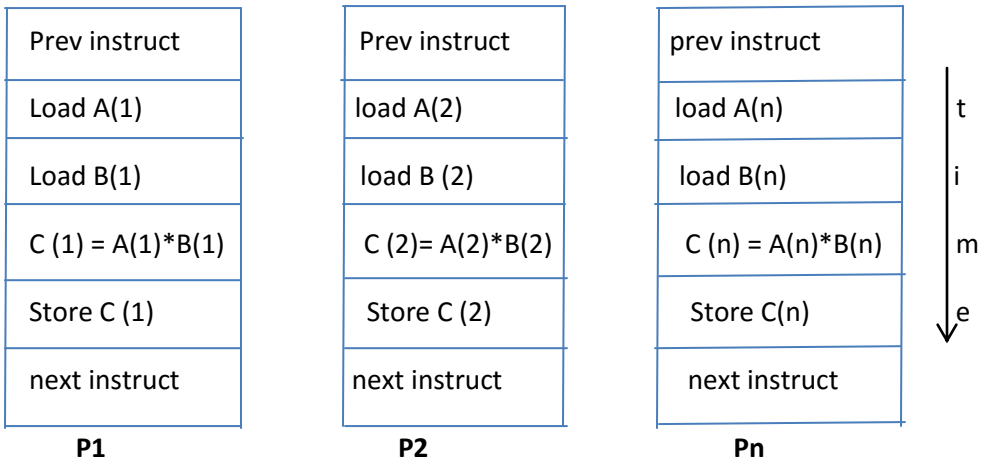


Fig 24 Schematic of Multiple Instruction, Single Data (MISD)

Few actual examples of this class of computer have ever existed. One is the experimental Carnegie-Mellon Computer (1971). Some conceivable uses might be multiple frequency filters operating on a single signal stream and multiple cryptography algorithms attempting to crack a single coded message.

Note: Shared memory machine as shown in fig 19b can be divided into two main classes based upon memory access time: UMA and NUMA

UNIFORM MEMORY ACCESS (UMA): Most commonly represented today by Symmetric Multiprocessor (SMP) machines. Identical processors with equal access and access times to memory. Sometimes called CC-UMA - Cache Coherent UMA. Cache coherent means if one processor updates a location in shared memory, all the other processors know about the update. Cache coherency is accomplished at the hardware level.

NON-UNIFORM MEMORY ACCESS (NUMA): Often made by physically linking two or more SMPs. One SMP can directly access memory of another SMP. Not all processors have equal time to all memories; memory access across link is slower. If cache coherency is maintained, then it may also be called CC-NUMA-Cache Coherent NUMA.

Advantages of Shared Memory

Global address space provides a user-friendly programming perspective to memory. Data sharing between tasks is both fast and uniform due to the proximity of memory to CPUs.

Disadvantages: Primary disadvantage is the lack of scalability between memory and CPUs. Adding more CPUs can geometrically increases traffic on the shared memory- CPU path, and for Cache Coherent Systems, geometrically increase traffic associated with cache/memory management. Programmer responsibilities for synchronization constructs and assure "correct" access of global memory. It becomes increasingly difficult and expensive to design and produce shared memory machines with ever increasing number of processors.

DISTRIBUTED MEMORY

The general characteristics of Distributed memory systems like shared memory systems, varied widely but share a common xtics. Distributed memory systems require a communication network to connect inter-processor memory. Processors have their own local memory. Memory address in one processor do not map to another processor , so there is no concept of global address space across all processors. Because each processor has its own local memory, it operates independently. Changes it makes to its local memory have no effect on the memory of other processors. Hence, the concept of cache coherency does not apply. When a processor needs access to data in another processor, it is usually the task of the programmer to explicitly define how and when data is communicated. Synchronization between tasks is likewise the programmer's responsibility. The network "fabric" used for data transfer varies widely, though it can be simple as Ethernet.

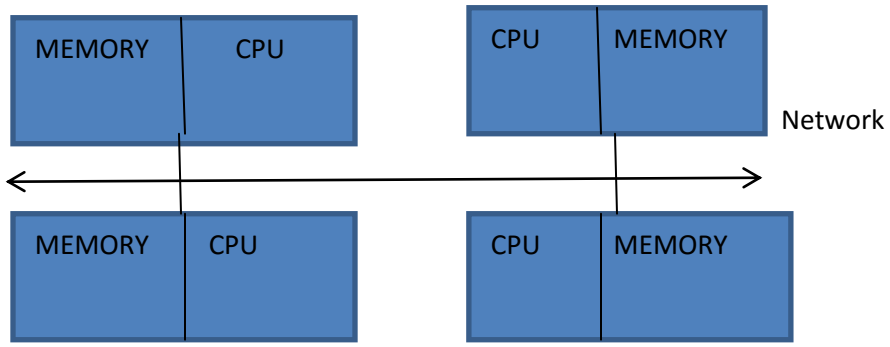
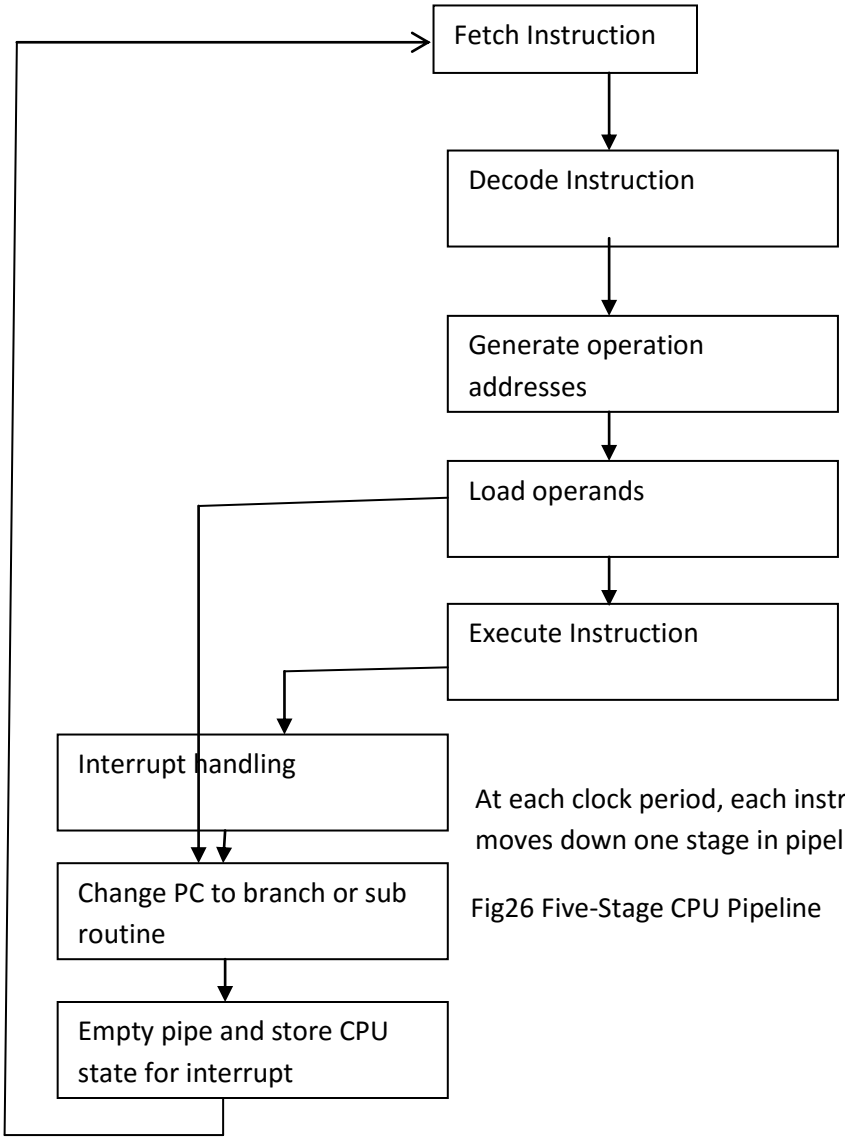


Fig 25 Schematic of Distributed Memory

Advantages: Memory is scalable with number of processors. Increase the number of processors and the size of memory increase proportionately. Each processor can rapidly access its own memory without interference and without the overhead incurred with trying to maintain cache coherency.
COST EFFECTIVENESS: Can use commodity, off-the-self processors and networking.

PIPELINED COMPUTERS
Pipeline is one of the techniques employed to greatly improve CPU speed of operation. Pipelining is the addition of parallel elements to the computer's arithmetic and control element so that several instructions can be worked on in parallel, increasing the throughput of the computer. The basic idea in pipeline is to begin carrying out new instructions b/4 execution of an old one is completed.
Pipelining is an experience occurred that typically specially in register character charts for addition, subtraction and other operations; where each step involved is outlined in the micro-program lists, which show micro instructions that must perform an instruction.

The possible list of steps to be carried out for an instruction is highlighted as follows:-
1. Fetch the instruction: Read the instruction word from memory
2. Decode the instruction. Identify the instructions based on the Op code and determine what control signals are required and when.
3. Generate operand address. Determine the address of the operand to be used.
4. Fetch operands. Access operands to be used.
5. Execute instruction to perform the required operation (addition, subtraction etc),



At each clock period, each instruction moves down one stage in pipeline

Fig26 Five-Stage CPU Pipeline

INTERRUPTS: Each block contains all the logic to perform the operation indicated.

The figure above shows the block diagram of a pipeline for the operations outlined.

A new instruction word is read into the fetch instruction section each clock period. This instruction word is passed to the decode instruction section , during the next clock period while a new instruction is taken in sequence from memory.

The block shown is implemented separately using gates and flip-flops and results are passed from block to block over connections. Control signals are generated so that each instruction is properly executed as

it passes through the pipeline. All the details of the instruction decoded (op-code, operand values etc) must be passed along each stage of the pipeline.

However, instructions such as floating-point instructions, multiplication and division which require more than the normal number of instruction steps, are accommodated by adding special sections for these operations. Under this influence, the control sees that subsequent instructions are delayed until the operations are completed.

Other problems encountered are:

1.BRANCH instructions that cause the order of instructions taken from memory to vary from the “normal” sequence.

2.It arises when an instruction modifies the operand to be used by a following instruction. Branch instruction are accommodated by delaying the reading of new instructions until BRANCH instruction is executed. This normally slows down the system used. And the operand modification is dealt with, by simply delaying the execution of the following instructions until new values have been formed. As a result of enhanced speed and additional throughput, pipelines have been made a component of almost every new machine, including most microprocessors.

An interrupt refers to the temporary transfer of control from a currently running program to an interrupt service as a result of externally or intentionally generated request, control returns to the program that was read after the service routine has been executed (An interrupt procedure resembles a subroutine call).

As just mentioned, interrupts can be externally generated (by I/O devices for example) or internally generated by CPU itself). (An example of latter occurs when a DIVIDE instruction attempts to divide a number by 0) internally generated are sometimes called traps.

After an interrupt occurs, the CPU must return to its state as of the time the interrupt occurred. The state of the CPU includes the current contents of the program counter, the contents of all processor registers and the content of all the status registers. These must be stored b/4 the interrupt is serviced and restored by the interrupt service routine (or User) b/4 operation of the program can be resumed.

REDUCED INSTRUCTION SET COMPUTER (RISC) & COMPLEX INSTRUCTION SET COMPUTER ARCHITECTURE (CISC)

The target of RISC is to trade low complexity for high performance. We are targeting high performance processor architecture that is less cumbersome, scalable, flexible with low power consumption and at a reduced price.

The different processor architecture include: RISC, CISC, STACK, VLIW & TTA.

We have been dealing with various type of processor architecture since the introduction of integrated circuits (IC) for the manufacture of computer systems. Improvement in chip technology allow more integration of transistors on a single chip resulting in higher scale of integration from Small Scale Integration (SSI) to Very Large Scale Integration (VLSI). This advancement in technology allows designers to manufacture complex hardware modules on a single chip. For instance, a research in computer architecture has made it possible to integrate a **whole system on a chip (SOC),** meaning components that used to exist separately but connected together through wires or buses can now co-exist in one complex IC. This however requires complex instructions to operate. The effect of this development enables architects in the 70's to design computers based on the Complex Instruction Set Computers (CISC) philosophy in which a lot of computational tasks were performed by the hardware in order to reduce the cost of software development which was very expensive at the time.

A contrary design strategy to CISC is Reduced Instruction Set Computers (RISC) whose philosophy rather assigns more responsibility to the software thereby making the H/W perform several simple tasks. The study of low complexity design in computer architecture can better be understood through an overview of CISC & RISC design philosophies. However the factors, the three factors that contribute to the differences in design strategies of CISC & RISC are: **storage and memory, compilers and VLSI technology.**

Storage & Memory: In 1970s, computers used magnetic core memory to store program code. The core memory was then expensive and very slow. The later introduction of RAM did not bring down the high cost of memory. Secondary storage was also expensive and slow. Hence a reduction in code size translated directly to a reduction in the total system cost.

Compilers: The job of compilers was simply to translate high level languages into **assembly language** which will then be converted to machine by the assembler. Hence the compilation stage took long time with sub-optimal output. Compact and optimized code was only possible if coded in assembler.

VLSI: Going by today's standard, transistor densities were low in the 70's. With this limitation, just little functionality could fit into one chip. The CISC machine at that time had functional units split up across multiple chips. A single-chip implementation would have been ideal becoz the delay-power penalty on data transfers between chips resulted in a limited performance.

Development from CISC to RISC architecture gave rise to powerful language like C++ or C-sharp that reduce op-code, with this in mind, softwware costs would be brought under control by simplifying the programmers' and compilers' jobs.

The simple design of functional units allows RISC to make use of large number of general-purpose registers. Hence, RISC instructions consists of simple register-to-register operations and LOAD and STORE instructions can access memory leading to a one-cycle operation by every instruction.

High Level Language Computer Architecture (HLLC)'s ideology assisted for the following reasons:

1.Reduced the difficulty of writing compilers and

2.Reduce the total system cost.

The time taken to execute a programme is

T = N x CPI x t where N is the No of dynamically executed instruction. CPI is the No of cycles per instruction, and t is the clock cycle.

The performance in eqn-1 below exposes the performance side of CISC

Time/program = Instructions/program x Cycle/Instruction x time/cycle---- -- -- - eqn-1

A CISC machine tries to reduce the amount of time it takes to run a program by reducing the no of instructions per program. Example of CISC machine : VAX 11/780, Pentium 235 Instruction.

While CISC design philosophy was motivated by shifting more responsibilities to hardware, the RISC philosophy believes in shifting the burden of code optimization from the H/W to the compiler.

In eqn-1 above, RISC prefers reduction of the second term to the right of equality sign (and allowing the first term to increase slightly) which ultimately reduces the time/program

ARCHITECTURE OF RISC PROCESSOR

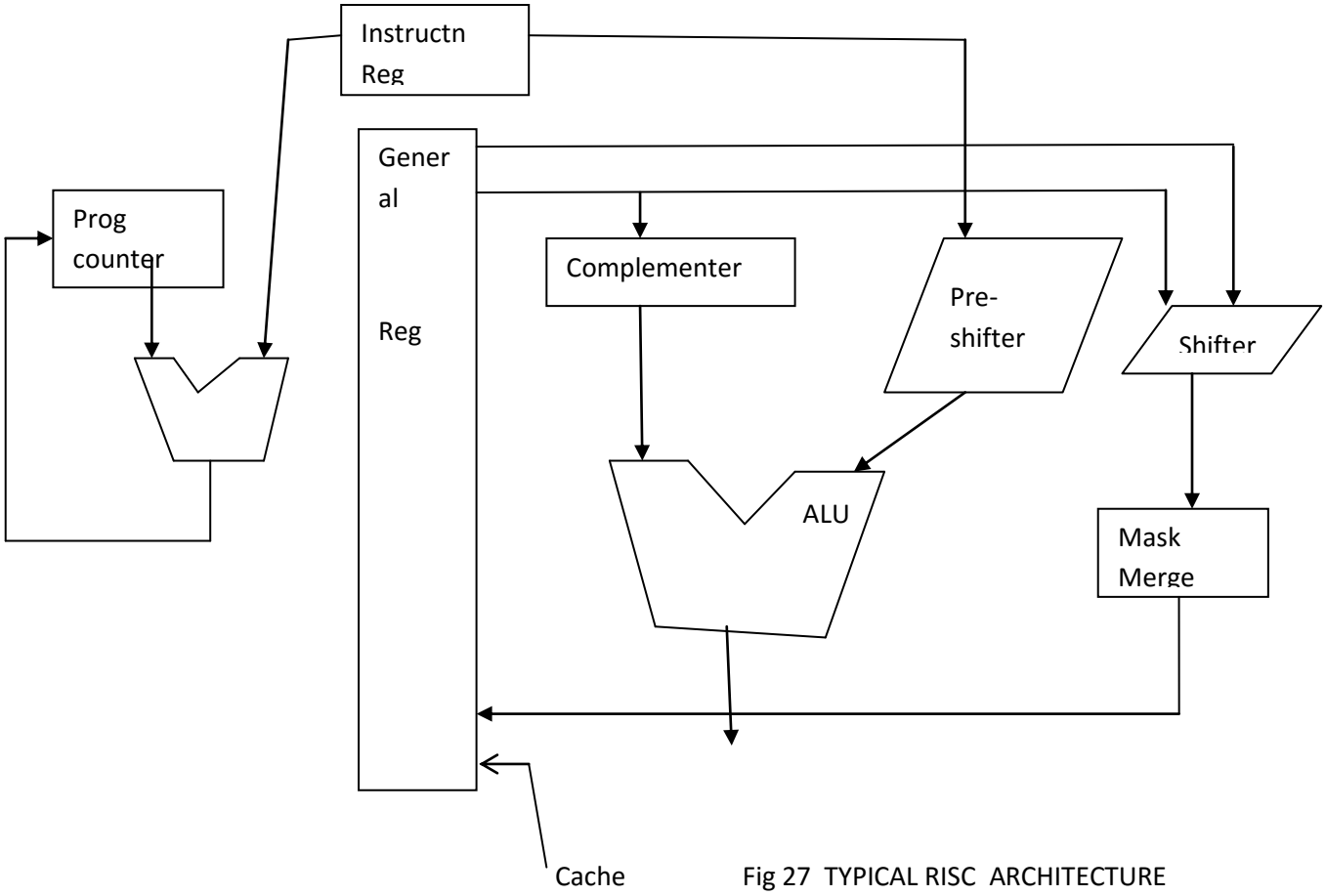


Fig 27 TYPICAL RISC ARCHITECTURE

CPU functional units are divided into the following functional groups as shown in fig27

1.Load and store: Load and store instructions transfer data between the memory and the coprocessors.

2. Two's complement arithmetic is performed on integers represented in two's complement notation. There are signed versions of add, subtract, multiply and divide.

Application of RISC processors

Example of RISC processors application: Sun SPARC, IBM Power PC, G3, MIPS Series, Alpha Series, Intel i 960.

TTA ----- Transport Triggered Architecture

VLIW----- Very Long Instruction Word.

EVOLUTION OF THE 8086 PROCESSOR FAMILY

These are micro-processor with multi-pins connections. Some of the pins will be useful as address pins (Registers), data pins (address pointers), power supply and earth. Some are left with or without any assigned function.

The processor came into being in early eighties and thus we have:

8086 has 40 connections pin, it requires 20 address pins, and 16 data pins and in addition to various other connectors. The CPU transfers data in different machine cycles using the same 16 pins AD0-AD15) ie Address pin 0-15.

8086, 80186, 80286, 80386, 80486: They all have data registers and address pointers. But they all have greater power because of the incorporation of I/O and system control H/W into the CPU's chip, with addition of several high level instruction and higher clock speed. The 186 has 68 pins, with most of the new pins use for I/L and control purposes.

The 8086/186 processors multiplex address/data pins, dia4 an instruction or data fetch cycle has to be divided into several sub-cycles. To write into memory, the CPU first places the address of the wanted memory on AD pins (Attached diagrams) and the memory sub-system must save the bits on the CPU, remove the address bit and places the data pins on the AD pins.

The 286 still has 68 pins, but uses them quite differently. There are now 24 address pins (A0-A23) and separate data pins (D0-D15) available for I/O and system control. Consequently additional chips for such purposes are required with 286 in configuring a complete computer, as in the case with 8086.

Ln contrast, the 186 does not require these and system configuration is simplified on the other hand it does not have Multi-tasking capabilities of the 286.

80386—This double the length of data and address registers so that each program can retain more data in CPU thereby increasing processing speed, and can also address large number of item, allowing the maximum program and provision is made for an off-chip memory **cache** to improve average memory access speed. The 386 has 132 pins in all. There are 32 address and 32 data pins that the number of system control pins is slightly large than of 286, bcoz more than ¼ of the pins are used by the power supply and earth connectors while several other pins are unused.

The 386 is more powerful than the 286 and much more powerful than 186.

It can be seen that the number of connector pins in a micro-program places a server limit on its power.

Other processors that follows are: Pentiums I, II. III & IV.

COMPUTER ORGANIZATION & ARCHITECTURE EIE 421

PRACTICE THE FOLLOWING QUESTIONS

1.(a) What do you understand by the phrase “size of a processor”?

(b) A processor is 64/32 bits. What is the meaning of this statement?

(c) In a computer hardware system what can you compare to (i) Human brain (ii) File cabinet

(iii) Writing table (iv) Road network within LMU.

(d) In data transfer within the bus system, what is the advantage of the parallel transmission over serial transmission?

2. (a) What do you understand by Von Neumann architecture?

(b) Mention the four possible classifications according to Flynn’s taxonomy.

(C) Highlight the merits and demerits of the Non-Uniform Memory Access ?

(d). With the aid of a well-labeled diagram, explain the economic importance of the distributed memory?

3. (a) Outline the various important features of RISCs on CISCs architecture?

(b) Explain the ways by which interrupts in any system can be made to occur?

(c) What are the problems encountered by the control unit for instructions such as floating-point instructions, multiplication and division operations to be executed in the CPU.

4 (a) Distinguish between Pipeline and Von Newmann based architecture?

(b) State the procedure involved for an instruction to be executed in a pipelined computer system?

5. (a) Write short notes on the following: (I) What are registers? (ii) Accumulators (iii) Program Counter (iv) Cache

(b) Write an assembly code using

(i) Three address instruction machine

(ii) Two address instruction machine

(iii) One address instructions

(iv) Zero address instruction

For X = (A + B) (C + B)

6. Computer as an electronic machine does not immerge from the blues, but rather immerged as a result of integration from different manufacturers. How does this integration done?

7. (a) What do you understand by instruction format?

(b) Explain various types of instruction formats existing in the CPU of hypothetical computer system?

(c) Describe in detail, various classifications of different busing structures?

8. (a) Super computer of today is a mini-computer computer of tomorrow. Discuss

(b) Provide the full meaning of the following acronyms and describe their modus operandi in computer system.

(i) EPROM

(ii) EAPROM

(iii) LIFO

(iv) RAM

(v) ROM

(vi) UMA

9. (a) What is operating system?

(b) What is software? Software are classified into 3 parts, name them and give at least one example each.

(c) Mention 3 functions of operating systems.

(d) Operating systems are either single-tasking or multi-tasking. What does this statement mean?

(e) Explain the difference: Mb and MB, Gb ad GB.

(f) What do you understand by the word super computer? A super-computer of today is a mini-computer of tomorrow, Explain with samples of micro-processors.

10 Give a brief family history of the three generations of computers. What do you think the next generation has in store for us?

11 (a) Write short notes on the following:

(ii) Accumulators (ii) Program counter (iii) Immediate addressing

(b) Evaluate the execution performance of the following expression in two computer systems with one accumulator and the other two accumulators.

A * (B + C * (D – E) + F)

ADDENDUM

SOFTWARE: A computer program that tells computer how to perform a task.

What is SERVER? It is a super computer that serve other computer on the network eg intranet/Internet.

What does it mean when I hear “The server is down?” Is server another type of computer?

When you hear that a server is down, it means a computer connected to that server can’t access the programs and files on that server.

SERVER- can be defined as a super-computer that serve other computers on a network eg intranet/internet.

SUPER COMPUTER Is ascribe super, if as at the time it is built, it is one of the fastest computers in the world.

Super computer can tackle tasks that would not be practical for other computers such as breaking codes and modeling world-wide weather system.

The basic part of a computer systems

VDU-Video Display Unit: It is a primary output. It displays what is typed from the keyboard or information upload on to memory or from hard drive.

Keyboard-It is primary input device. It is used to enter text and numbers.

System Unit: The case that holds the power supply, storage devices and the circuit boards. It houses the main circuit boards (Also call the mother board) which contains the processor (RAMs & ROMs)

Storage device (Hard disk derive)- It is used to store data on a computer or to share data among computer storage devices are both input and output devices. Data stored into it and are read from it.

Mouse- A common input device designed to move and select on-screen graphical objects and control.

Operating system- An operating (OS) is the master controller for all the activities that takes place within a computer system.

A bus is an electronic pathway that carries the electronic signals between the electronic parts of a computer system.

CACHE- Also called **RAM cache** or cache memory or Immediate addressing is a high-speed memory that a processor can access more rapidly than memory elsewhere on the mother board.

RAM - Read/Write Access Memory or Random Access Memory. It holds data temporarily in circuits. RAM is volatile, which means that it requires electrical power to hold the data. If the computer is off or if the power goes out, all data in RAM instantly and permanently disappears.

SDRAM- Synchronous Dynamic RAM

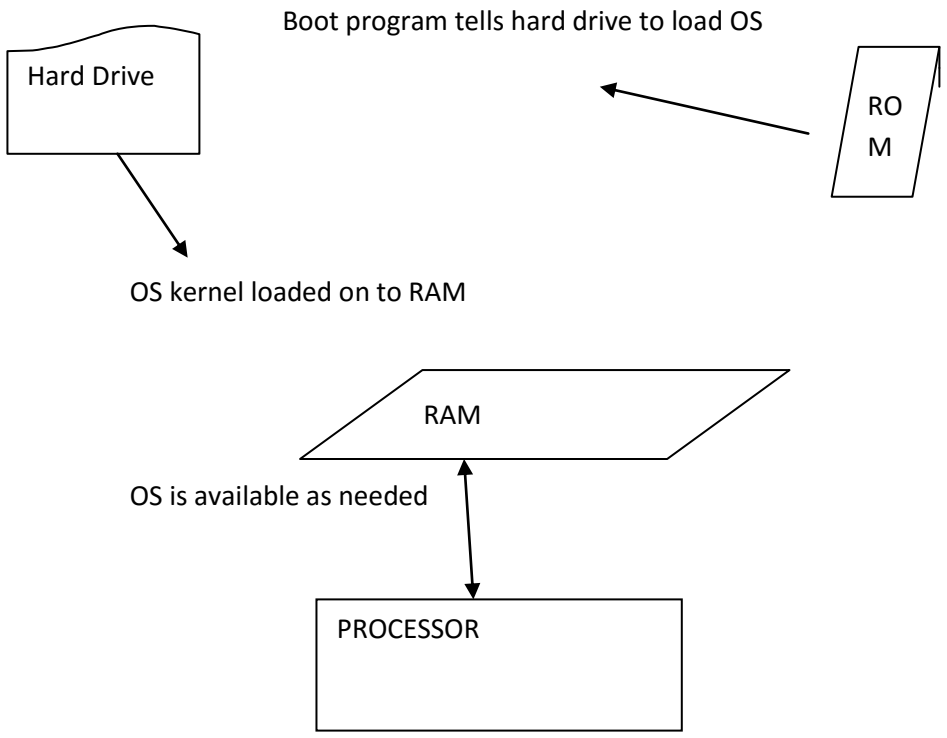
Virtual Memory: When RAM is full, the computer operation slows down. To solve this problem the OS uses an area of the Hard Drive to temporarily store data from RAM. This area of the hard drive is called virtual memory.

The data that has been moved from RAM to virtual memory is called a SWAP file.

ROM (Read-Only Memory)

It is a type of memory that holds the computers' start up routine and other basic information. ROM is housed in a single integrated circuit and is usually a large chip that is plugged into the mother board. The information in ROM is permanent and non-volatile ie the instructions are not lost even when the power is turned off. It carries out boots trap program. The boot process includes Power-On-Self Test (POST)

ROM AND THE BOOT PROCESS:



EEPROM – Electrically Erasable Programmable Read Only Memory.

The computer configuration settings are stored in EEPROM. It is non-volatile chip that requires no power to hold the data.

TERMS USED TO DESCRIBE DIGITAL DATA.

Term	Value	Term	Value
Bit	1 binary digit	Byte	8 bits
Kilo bit (Kb)	1,000 bits (Exactly 1,024)	Kilobyte (KB)	1,000 bytes (exactly 1,024)
Mega bit (Mb)	1 million bits	Mega bytes (MB)	1 million bytes
Giga bits (Gb)	1 billion bits	Giga bytes (GB)	1 billion bytes
Tera bytes (TB)	1 trillion bytes	Peta bytes (PB)	1,000 tera bytes (2^{50} bytes)
Exa bytes (EB)	1,000 peta bytes (2^{60} bytes)	Zetta byte (ZB)	1,000 exa bytes (2^{70} bytes)

An Intel core i 7 quad-core processor has a clock speed of 2.50Ghz. Calculate the time to take a processor to fetch a data from the memory.

CLOUD COMPUTING

Definition: Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources eg (n/work, servers, storage, application and services)that can be rapidly provisioned and released with minimal management effort or service provider interaction.

The cloud itself is a network of data centers, each composed of many thousands of computers working together that can perform the functions of software on a personal or business computer by providing users access to powerful applications, platform and services delivered over the internet.

Cloud computing is a type of computer that relies on sharing computing resources rather than having local servers or personal devices to handle applications.

Cloud computing means storing and accessing data and programs over the internet instead of your computer hard drives. The term 'Cloud' symbolizes the internet, which in itself is a "network of networks".

SOME NAMES OF CLOUD COMPUTING VENDORS

1.Financial force .com

2. Google

3. Microsoft

4. Cisco Systems

5. Oracle

6. IBM

7. AT & T

8. Blue Coat

9. Hewlet Parckard

10. Century Link Technology

The two key words that is killing the dominating world of computing by Bill Gate is "CLOUD COMPUTING". The days of paying for costly software upgrades are numbered. The PC will soon be obsolete. The two words CLOUD COMPUTING scare the hell out of Bill Gates and has this to say: :" The next sea change is upon us". The PC age is given way to a new era: "Utility age". Data

centers have become as vital to the function of society as power stations. With “Cloud Computing”, all the application soft-wares are stored at a giant data center somewhere in the “cloud”.

With era of existing fiber cables in Nigeria, bringing down ”data centers” will be made easy. As a result, computing is fast becoming a utility in much the same way that electricity did.

Design LLC incorporation, the giant company behind the mask has grown the servers up to 20 Million by 2014. APPLE tablets is the 1st ‘bride’ of cloud computing.

People who are still using PC- any time you wanted to type a letter, create a spread sheet, edit a photo, or drawing or play a game, you have to go to the store, install it on your computer. In short years to come, cloud computing will actually bury PC.

The first set of computers as ‘bride’ to Cloud Computing:

APPLE ---------- Tablet computer (i pad)

HP ---------- Slate computer (i pad)

S5 XX ----------Android tablet (cloud i pad)

KEY FEATURES OF CLOUD COMPUTING:

Agility- helps in rapid and inexpensive re-provisioning of resources

Location independence- resources can be accessed from anywhere and everywhere.

Multi-tenancy – resources are shared amongst a large pool of users

Reliability – dependable accessibility of resources and computable

Scalability- dynamic provisioning of data helps in avoiding various bottle neck scenarios

Maintenance - Users (companies/organizations) have less work in terms of resource upgrades and management which is the new paradigm will be handled by service providers of cloud computing.

The Cloud Computing

Architecture: 

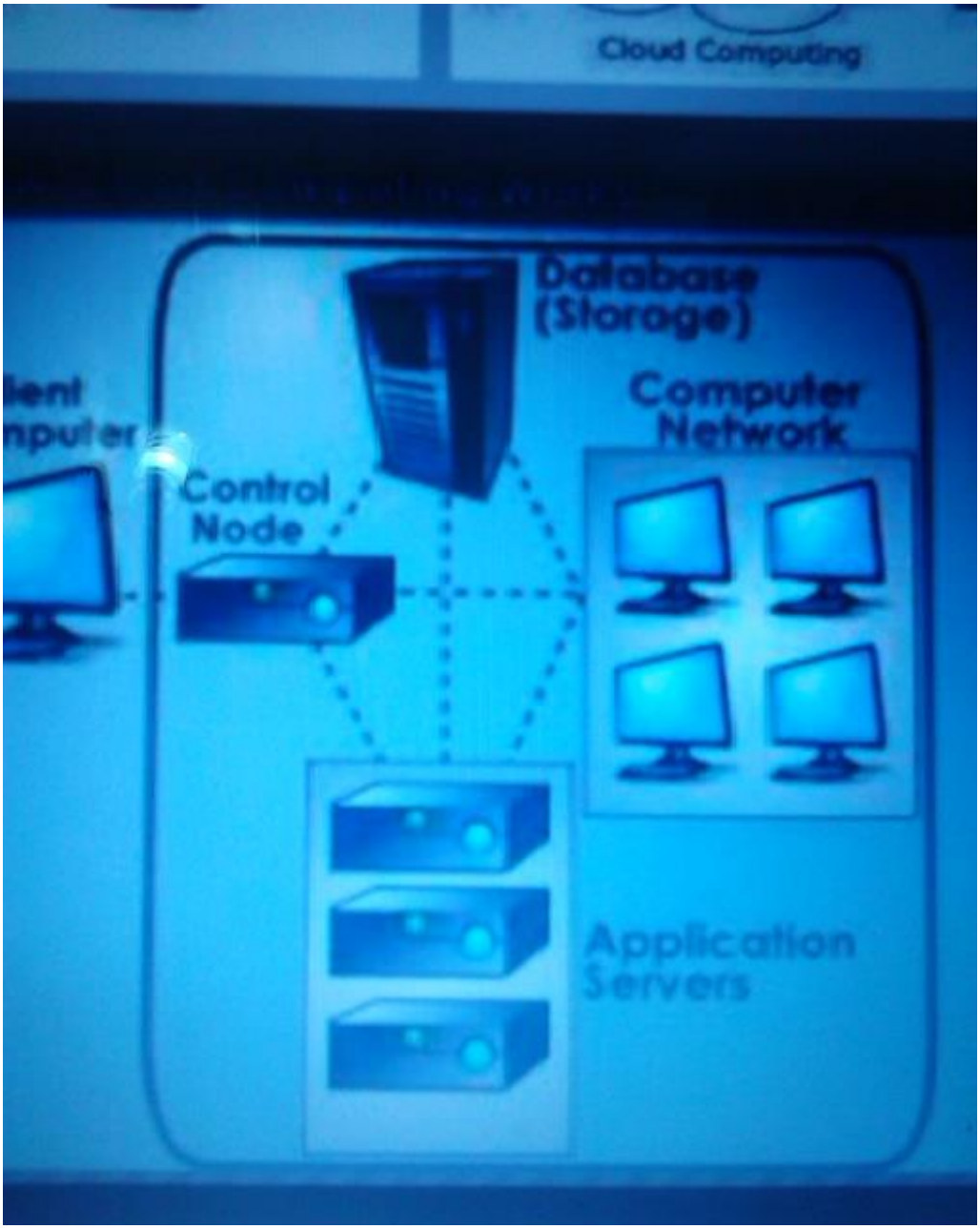


Fig Layered Architecture for a customized cloud service.

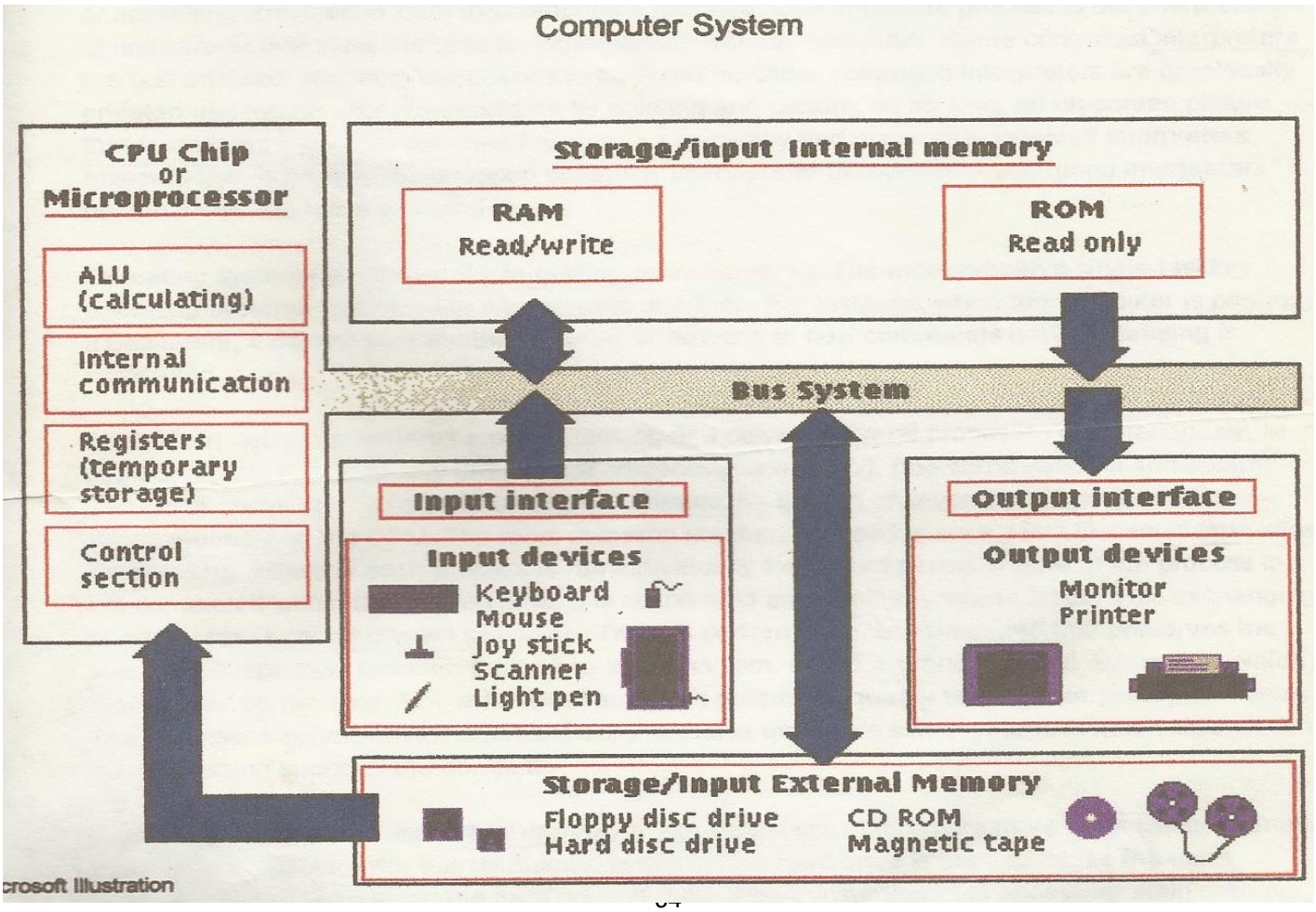
As with classical 7-layer OSI model of data networks, the layered model in cloud computing serves the same general purpose. Depending on the service requirement of an application, these layers are shuffled to create a customized architecture.

The layered architecture adheres to the principle of Service Oriented Architecture (SOA) which forms the core of the cloud computing paradigm. The components of a basic layered architecture are shown below in Fig

Given an acronym: C-SAPSI.

Clients
Services
Application
Platform
Storage
Infrastures

Fig Layered architecture for a customized cloud service.



Operating System: (OS), in computer science, the basic software that controls a computer. The operating system has three major functions: It coordinates and manipulates computer hardware, such as computer memory, printers, disks, keyboard, mouse, and monitor. It organizes files on a variety of storage media, such as floppy disk, hard drive, compact disc, and tape; and it manages hardware errors and the loss of data.

How IS OS Works

Operating systems control different system program, such as running a spreadsheet program or accessing information from the computer's memory. One important process is the interpretation of commands that allow the user to communicate with the computer. Some command interpreters are text oriented, requiring commands to be typed in. Other command interpreters are graphically oriented and let the user communicate by pointing and clicking on an icon, an on-screen picture that represents a specific command, Beginners generally find graphically oriented interpreters easier to use, but many experienced computer users prefer text-oriented command interpreters because they are more powerful

Operating Systems are either single-tasking or multitasking. The more primitive single-tasking operating systems can run only one process at a time. For instance, when the computer is printing a document, it cannot start another process or respond to a new commands until the printing is completed.

All modern operating systems are multitasking and can run several processes simultaneously. In most computers there is only one central processing unit (CPU), (the computational and

control unit of the computer), so a multitasking OS creates the illusion of several processes running virtual memory simultaneously on the CPU. The most common mechanism used to create this illusion is time-slice multitasking, whereby each process is run individually for a fixed period of time. If the process is not completed within the allotted time, it is suspended and another process is run. This exchanging of processes is called context switching. The OS performs the "book-keeping" that preserves the state of a suspended process. It also has a mechanism, called a scheduler, that determines which process will be run next. The scheduler runs short processes quickly to minimize perceptible delay. The processes appear to run simultaneously because the user's sense of time is much slower than the processing speed of the computer.

Operating systems can use virtual memory to run processes that require more main memory than is actually available. With this technique, space on the hard drive is used to mimic the extra memory needed. Accessing the hard drive is more time-consuming than accessing main memory, however, so performance of the computer slows.

Current Operating Systems

The operating systems commonly found on personal computers are UNIX, Macintosh OS, MS-DOS, OS/2, and Window '95. UNIX, developed in 1969 at AT & T Bell Laboratories, is a popular operating system among academic computer users. Its popularity is due in large part to the growth of the interconnected computer network known as the internet, the software for which initially was designed for computers that run UNIX. Variations of UNIX include sun OS (distributed by SUN Microsystems Inc.) Xenia (distributed by Microsoft Corporation), and Linux. UNIX and its clones support multitasking and multiple users. It file system provides a simple means of organizing disk files and lets users protect their files from other users. The commands in UNIX are not intuitive, however, and mastering the system is difficult.

DOS (Disk Operating System) and its successor, MS-DOS, are popular operating systems among users of personal computers. The file systems of DOS and MS-DOS are similar to that of UNIX, but they are single user and single tasking because they were developed before personal computers became relatively powerful. A multitasking variation is OS/2, initially developed by Microsoft Corporation and international Business Machines (IBM)

Few computer users run MS-DOS or OS/2 directly. They prefer versions of UNIX or windowing systems with graphical interfaces, such as Windows 95 or the Macintosh OS, which make computer technology more accessible. However, graphical systems generally have the disadvantages of requiring more hardware-such as faster CPUs, more memory, and higher-quality monitors-than command-Oriented operating systems.

Future Technologies

Operating systems continues to evolve. A recently developed type of OS called a distributed operating system is designed for a connected, but independent, collection of computers that share resources such as hard drives. In a distributed OS, a process can run on any computer in the network (Presumably a computer that is idle) to increase that process's performance. All basic OS functions-such as maintaining file systems, ensuring reasonable behavior, and recovering data in the event of a partial failure-become more complex in distributed systems.

Research is also being conducted that would be replace the keyboard with a means of using voice or handwriting for input. Currently these types of input are imprecise because people pronounce and write words very differently, making it difficult for a computer to recognize the same input from different users. However, advances in this field have led to systems that can recognize a small number of words spoken by variety of people. In addition, software has been developed that can be taught to recognize an individual's handwriting.

System, any collection of component elements that work together to perform a task. In computer science, Science is used in a variety of contexts. A Computer is a hardware system consisting of a microprocessor and allied chips and circuitry, plus an input device (keyboard, mouse, disk drive), an output device (monitor, disk drive), and any peripheral devices (printer, modem). Within this hardware system is an Operating System, often called system software, which is an essential set of programs that manage hardware and data files and work with application programs. External to the computer, system also refers to any collection or combination of programs, procedures, data, and equipment utilized in processing information: an accounting system, a billing system, a database management system.

Today's UNIX System.

What is an Operating System? An **operating system** (OS) is a program that allows you to interact with the computer -- all of the software and hardware on your computer.

UNIX system is based on an Open Systems strategy .

The key to the continuing growth of the UNIX system is the free-market demands placed upon suppliers who produce and support software built to public standards. The "open systems" approach is in bold contrast to other operating environments that lock in their customers with high switching costs. UNIX system suppliers, on the other hand, must constantly provide the highest quality systems in order to retain their customers. Those who become dissatisfied with one UNIX system implementation retain the ability to easily move to another UNIX system implementation.

The continuing success of the UNIX system should come as no surprise. No other operating environment enjoys the support of every major system supplier. Mention the UNIX system and IT professionals immediately think not only of the operating system itself, but also of the large family of hardware and application software that the UNIX system supports. In the IT

marketplace, the UNIX system has been the catalyst for sweeping changes that have empowered consumers to seek the best-of-breed without the arbitrary constraints imposed by proprietary environments.

The market's pull for the UNIX system was amplified by other events as well. The availability of relational database management systems, the shift to the client/server architecture, and the introduction of low-cost UNIX system servers together set the stage for business applications to flourish. For client/server systems, the networking strengths of the UNIX system shine. Standardized relational database engines delivered on low-cost high-performance UNIX system servers offered substantial cost savings over proprietary alternatives.

The UNIX System Tradition

For the last two and a half decades, the UNIX system has upheld a tradition of providing its customers with early access to new technologies. For example:

- UNIX systems provided early access to RISC technology. Applications in CAD/CAM, multimedia, and large-scale publishing that demanded high performance workstations have always used the UNIX system platform.
- The UNIX system provided early access to symmetric multiprocessing [SMP] computers. UNIX system-based SMP parallel processors radically improved the price/performance of midsize and high-range servers.
- The UNIX system first enabled distributed transaction processing in conjunction with TP monitors from independent software suppliers. The server side of client/server technology was launched on the UNIX system.
- The UNIX system launched the Internet and the World Wide Web.

UNIX system suppliers also have a proud tradition of integration with legacy systems as well as innovation to uphold. No other system can ensure that disparate systems - usually proprietary systems - can be integrated, allowing the buyer's investment in data and information to be realized with minimal disruption and reinvestment.

There is every reason to believe that the UNIX system will continue to be the platform of choice for innovative development. In the near term, for example, UNIX system vendors will define the scope of Java and provide the distributed computing environment into which the Network Computer terminal will fit and enable it to thrive and grow.

How will Java and the Network Computer terminal manifest themselves? The exact answer is unknown; however, in open computing, the process for finding that answer is well understood. The UNIX system community has set aside (via consensus standards) the wasteful task of developing arbitrary differences among computing environments. Rather than building proprietary traps, this community is actively seeking ways to add value to the UNIX system with improved scalability, reliability, price/performance, and customer service.

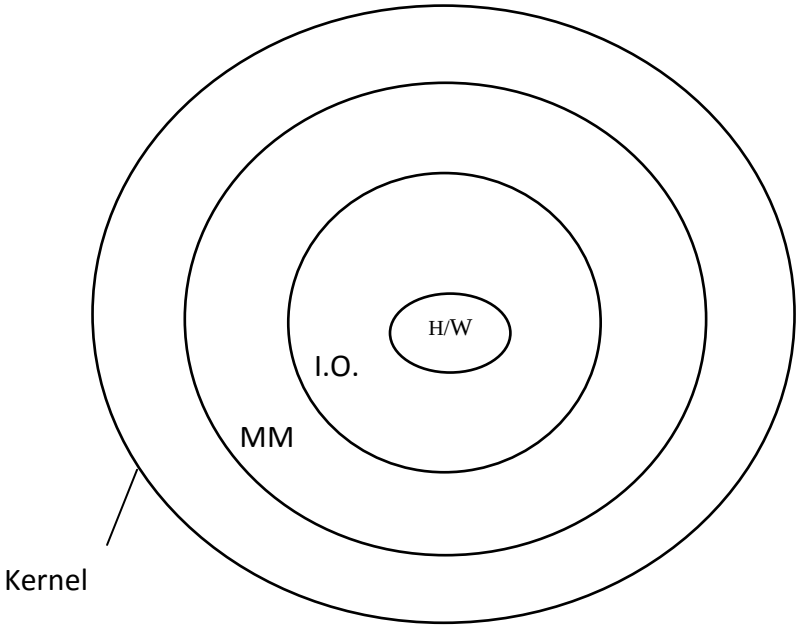
Java and the Network Computer terminal offer several potential advantages for consumers. One key advantage is a smaller, lighter, standards-based client. A second advantage is a specification that is not controlled by one company, but is developed to the benefit of all by an open, consensus process. Thirdly, greater code reuse and a component software market based on Object technology, such as CORBA and Java. All of these options and more are being deployed first by members of the UNIX system community.

CASE-Computer Assisted Software Engineering

CAD/CAM – Computer Aided Design/Manufacturing.

Unix systems have been successfully implemented on a wide variety of supercomputers, mainframes, minicomputers, advance workstations and personal computers. Early Unix systems were real storage systems but today's UNIX systems are almost exclusively Virtual storage systems. The UNIX systems is 'open' because large tasks may often be accomplished by combining existing small programs rather by developing new programs. This kind of composition is facilitated by UNIX systems ability to route the output of one program directly to the input of another.

Unix OS Layers



THE SHELL:

The shell is the UNIX systems' mechanism for interacting between users and the systems. It is command interpreter that reads lines typed by the user and causes the execution of the requested system features. It is an application program like any other, it is not part of the kernel. Customs shells are often written. Unix systems often support several different shells. The shell is not permanently resident in main memory like the kernel-it can be swapped as needed.

THE KERNEL:

Unix systems contain a kernel, one or more shells, and various utilities. The kernel is the central part of UNIX operating systems, it encapsulates the hardware and provides UNIX SYSTEMS SERVICES TO APPLICATION PROGRAMS. The kernel provides process management, memory management, I,O management, and timer management, much as the shell provides services to users, the kernel provides services to application programs (including the shell).

The kernel manages real memory and allocates the processor automatically, other kernel functions are provided in response to request.

UNIX 95 Registered Products

DIGITAL: Digital UNIX® Version 4.0 running Digital's AlphaStations and Digital's AlphaServers

HITACHI: Hitachi 3050RX, 3500/3X, 3500/4XX running HI-UX/WE2 Version 06-01

and later

HITACHI: Hitachi 3500 running HI-UX/WE2 Version 07-01

and later

Hewlett-Packard: HP-UX Release 10.20 and later on all HP9000 Series 700 and 800

HP-UX Release 11.00 or later (in both 32 and 64-bit configurations) on HP9000 Series (all models)

IBM: IBM POWER, POWER2, and PowerPC™ Systems with IBM AIX® Version 4.2

or later

IBM: OS/390 Version 1 Release 2 or later with OS/390 V1R2

or later

Security Server and OS/390 V1R2

or later C/C++ Compiler on IBM System/390 Processors that support OS/390 Version 1 Release 2

NCR: NCR UNIX System V Release 4 MP-RAS Release 3.02

or later on NCR WorldMark Series & System 3000 Series

NEC: UX/4800 R12.3 and later on UP4800 and EWS4800 Series

SCO: SCO UnixWare® Family R2.1.1

and later for single and multiprocessor Intel™ 386/486 or Pentium® PCs conforming to PC/AT architectures

SGI: IRIX 6.5 running on Silicon Graphics systems using

the MIPS R4000, R5000, R8000 and R10,000 family of

processors

SNI: Business Servers running BS2000/OSD V3.0

and higher

SNI: Reliant UNIX V5.43 running on RM Server Family

Siemens Pyramid: Reliant UNIX Version 5.43 running Cluster Server

SUN: Solaris 2.6 on SPARC based systems

SUN: Solaris 2.6 on x86pc based systems

Benefits for Application Developers

A single standard for the UNIX operating system means:

- Improved portability.
- Faster development through the increased number of standard interfaces.
- More innovation is possible, due to the reduced time spent porting applications.

Benefits for Users

The Single UNIX Specification will evolve and develop in response to market needs protecting users investment in existing systems and applications. The availability of the UNIX system from multiple suppliers gives users freedom of choice rather than being locked in to a single supplier. And the wide range of applications - built on the UNIX system's strengths of scalability, availability and reliability - ensure that mission critical business needs can be met.

In Summary

When the history of the information age is written, the extraordinary dynamics of the UNIX system marketplace will be seen as playing an important role. The UNIX system was developed

at just the right time and place to be the critical enabler for a revolution in information technology. Client/server architectures, the Internet, object databases, heterogeneous transaction processing, and Web computing all emerged on the shoulders of the UNIX system.