COA
Chinmoy sir

Chapter 9 Solution

Q: Explain The Booth's Algorithm with flow chart 4

Algorithm:

The **number of steps** required is equal to the **number of bits in the multiplier.**

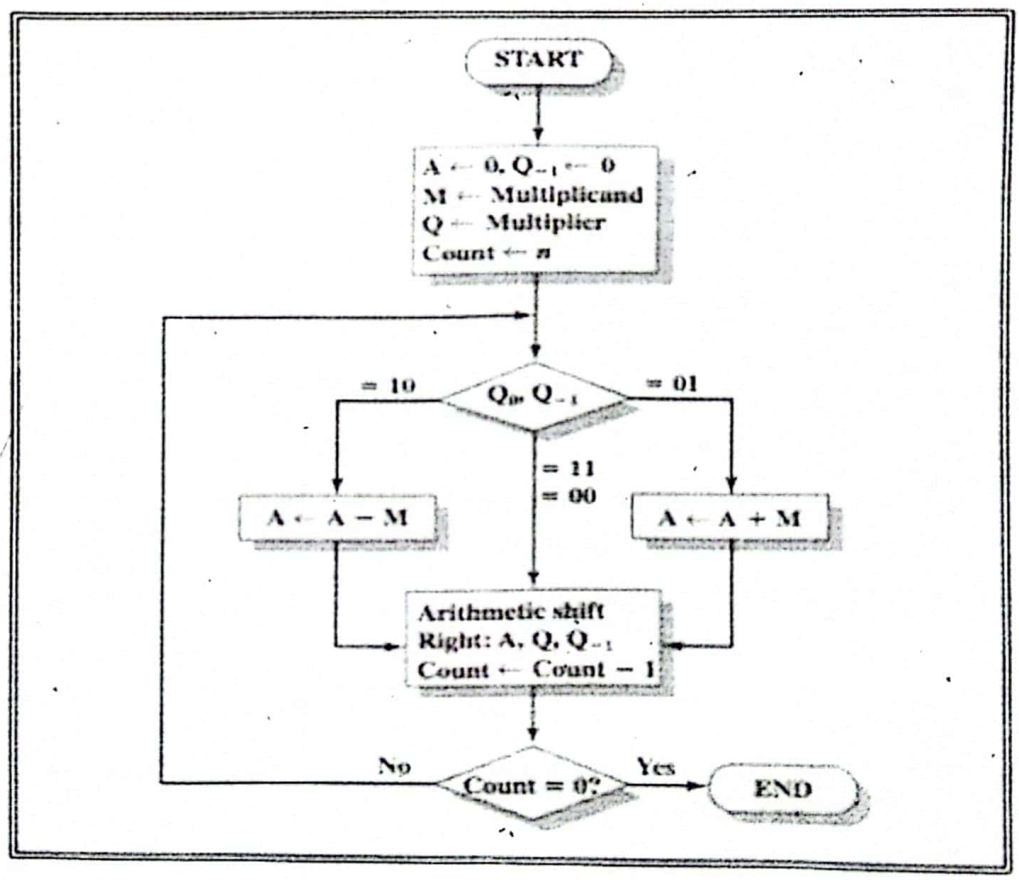
At the beginning, consider an **imaginary "0" beyond LSB of Multiplier**

1) At each step, **examine two adjacent Multiplier bits** from **Right to Left.**
2) If the transition is from **"0 to 1"** then **Subtract M** from **A** and **Right-Shift** (A & Q) combined.
3) If the transition is from **"1 to 0"** then **ADD M** to **A** and **Right-Shift.**
4) If the transition is from **"0 to 0"** then **simply Right-Shift.**
5) If the transition is from **"1 to 1"** then **simply Right-Shift.**

Repeat steps 1 to 5 for **all bits** of the multiplier.

The **final answer** will be in **A & Q** combined.

FLOWCHART FOR BOOTH'S ALGORITHM



CISC, RISC
Measurement
Virtual memory

Difference twos Complement representation of a number and twos complement of a number.

CamScanner

Q: Distinguish between 2's complement and sign magnitude representation 3

Aspect	Two's Complement	Sign-Magnitude
Representation of Positive Numbers	Same as usual binary representation. MSB is 0.	Same as usual binary representation. MSB is 0.
Representation of Negative Numbers	Take the two's complement of the absolute value. Invert all bits and add 1.	Flip the sign bit (1 for negative) and represent the absolute value in binary.
Zero Representation	Only one representation for zero (all bits are 0).	Two representations for zero: positive zero (sign bit = 0) and negative zero (sign bit = 1).
Range	Symmetric range from -2^(n-1) to 2^(n-1) - 1 in an n-bit representation.	Asymmetric range from -(2^(n-1) - 1) to 2^(n-1) - 1 in an n-bit representation.
Arithmetic Operations	Simplified, as addition and subtraction use the same hardware.	Requires special handling of the sign bit in addition and subtraction.
Positive and Negative Zero	Only one representation for zero.	Two representations for zero: positive zero and negative zero.

-2^{n-1} to $2^{n-1}-1$

$-(2^{n-1}-1)$ to $2^{n-1}-1$

Q : Assume numbers are represented in six bit 2's complement representation . show the calculation of -1A +2B

To perform the addition of two numbers represented in six-bit two's complement representation (-1A and +2B), let's first convert the hexadecimal values to binary and then add:

1. **Conversion to Binary:**
   - $-1A_{16} = 100110$ (6-bit two's complement)
   - $+2B_{16} = +101011$ (6-bit number)
2. **Perform Binary Addition:**

```
  100110
+ 101011
---------
010001   (Result in binary)
```

3. **Conversion to Hexadecimal:**
   - $110112 = 11\ (16)$

So, in six-bit two's complement representation, the result of $-1A_{16} + 2B_{16}$ is $11(16)$

CamScanner

Q: Analyze the statement CPU performe's all arithmetic operation using addition

The statement "CPU performs all arithmetic operations using addition" is not entirely accurate. While it's true that addition is a fundamental operation and many arithmetic operations can be implemented using addition, CPUs use a combination of arithmetic and logical operations to perform various mathematical computations. Let's analyze the statement:

1. **Addition as a Fundamental Operation:**
   - Addition is a basic arithmetic operation, and many other operations can be broken down into sequences of additions.
   - For example, subtraction can be implemented as addition with the two's complement of the number to be subtracted.
2. **Diverse Arithmetic Operations:**
   - CPUs support a range of arithmetic operations beyond addition and subtraction. These include multiplication, division, and more.
   - Multiplication and division are typically more complex operations and often involve dedicated hardware or specialized algorithms.
3. **Optimized Hardware for Specific Operations:**
   - Modern CPUs have specialized hardware for different arithmetic operations. For example, there are specific units for floating-point arithmetic, vector operations, and others.
   - These units are optimized to perform their designated operations efficiently.
4. **Logical and Bitwise Operations:**
   - CPUs also perform logical and bitwise operations, which are crucial for tasks like conditional branching, bit manipulation, and Boolean logic.
   - Logical operations include AND, OR, XOR, and NOT, while bitwise operations manipulate individual bits in binary representations.
5. **Instruction Set Architecture (ISA):**
   - The CPU's instruction set architecture defines the set of instructions it supports. Arithmetic and logical instructions are part of the ISA, and CPUs are designed to execute these instructions efficiently.

In summary, while addition is a fundamental operation and forms the basis for many arithmetic computations, modern CPUs are equipped with a variety of specialized hardware and instructions to perform a diverse range of arithmetic and logical operations.

CS CamScanner

Q : Assume numbers are represented in 8bit twos complemented representation. show the calculation of :

(i) -13+6 and (ii) -6-13

(i) −13+6:

1. **Representation of -13 in 8-bit two's complement:**
   - -13 in decimal is 11110011 in binary.
2. **Representation of +6 in 8-bit two's complement:**
   - +6 in decimal is 00000110 in binary.
3. **Add -13 and +6:**
   - Add the two numbers obtained in steps 1 and 2.

```
  11110011   (-13)
+ 00000110   (+6)
------------
  11111001   (-7, discard overflow)
```

So, −13+6 in 8-bit two's complement is −7.

(ii) −6−13:

1. **Representation of -6 in 8-bit two's complement:**
   - -6 in decimal is 11111010 in binary.
2. **Representation of -13 in 8-bit two's complement:**
   - -13 in decimal is 11110011 in binary.
3. **Add -6 and -13:**
   - Add the two numbers obtained in steps 1 and 2.

```
  11111010   (-6)
+ 11110011   (-13)
------------
  11101101   (-19, discard overflow)
```

10011

-19

So, −6−13 in 8-bit two's complement is −19.

CamScanner

2.4 Four benchmark programs are executed on three computers with the following results:

	Computer A	Computer B	Computer C
Program 1	1	10	20
Program 2	1000	100	20
Program 3	500	1000	50
Program 4	100	800	100

The table shows the execution time in seconds, with 100,000,000 instructions executed in each of the four programs. Calculate the MIPS values for each computer for each program. Then calculate the arithmetic and harmonic means assuming equal weights for the four programs, and rank the computers based on arithmetic mean and harmonic mean

10^8

MIPS = Instruction count / Execution time $\times 10^6$

Answer:

MIPS rate given as $MIPS = \frac{I_c}{T \times 10^6}$

Arithmetic mean is $R_A = \frac{1}{m}\sum_{i=1}^{m} R_i$

Harmonic mean is $R_H = \frac{m}{\sum_{i=1}^{m} \frac{1}{R_i}}$

By applying MIPS = $I_c / (T \times 10^6) = 100{,}000{,}000/(T \times 10^6) = 100/T$. Therefore, the MIPS values are:

$= \frac{10^8}{T \times 10^6} = \frac{100}{T}$

MIPS

	Computer A	Computer B	Computer C
Program 1	100	10	5
Program 2	0.1	1	5
Program 3	0.2	0.1	2
Program 4	2	0.125	1

Arithmatic mean

	Arithmetic mean	Rank
Computer A	25.575	1
Computer B	2.80	3
Computer C	3.25	2

Harmonic mean

	Harmonic mean	Rank
Computer A	0.25	2
Computer B	0.21	3
Computer C	2.1	1

CamScanner

Q : Illustrate and list the microoperation to perform addition between two operands from memory

Performing addition between two operands from memory involves a series of microoperations that manipulate data within the CPU. Below is an illustration of the microoperations involved in a simplified form:

List of Microoperations:

1. **Memory Read Operation:**
   - MAR (Memory Address Register) is loaded with the address of Operand 1.
   - Memory reads the data from the specified address into MDR (Memory Data Register).
2. **Load Operand 1:**
   - MDR is transferred to a register (e.g., Register A).
3. **Memory Read Operation:**
   - MAR is loaded with the address of Operand 2.
   - Memory reads the data from the specified address into MDR.
4. **Load Operand 2:**
   - MDR is transferred to another register (e.g., Register B).
5. **Addition Operation:**
   - ALU performs the addition operation on the contents of Register A and Register B.
   - Flags may be set (e.g., carry flag) based on the addition.
6. **Store Result:**
   - The result from the ALU is stored in a destination register or memory location.

Q : What is micro operation and Types

A microoperation is a fundamental operation or manipulation performed on data at the level of individual bits or groups of bits within a digital computer's registers. Microoperations are the building blocks of more complex operations in a computer's central processing unit (CPU). These operations are executed at a very basic level, involving the transfer, manipulation, or control of data within the CPU. Microoperations can be categorized into different types based on their functions. Some common types of microoperations include:

- operations executed on data stored in registers are known as microoperation.

CamScanner

1. **Data Transfer Microoperations:**
   - Involve the movement of data from one register to another or between registers and memory.
2. **Arithmetic Microoperations:**
   - Include basic arithmetic operations such as addition, subtraction, incrementation, and decrementation.
3. **Logic Microoperations:**
   - Involve logical operations such as AND, OR, NOT, and XOR, which are applied to the binary representations of data.
4. **Shift Microoperations:**
   - Move the bits of a binary number to the left or right within a register.
5. **Control Microoperations:**
   - Influence the control signals of the CPU, affecting the sequencing of operations.

Q :what is the distinction between the written sequence and time sequence of an instruction

- **Written Sequence:**
  - Refers to the order of instructions in a program.
  - Follows the logical flow of the algorithm.
  - Sequential representation in code.
- **Time Sequence:**
  - Refers to the actual execution order of instructions.
  - Influenced by hardware factors like pipelining and parallelism.
  - May differ from the written sequence for optimization purposes.

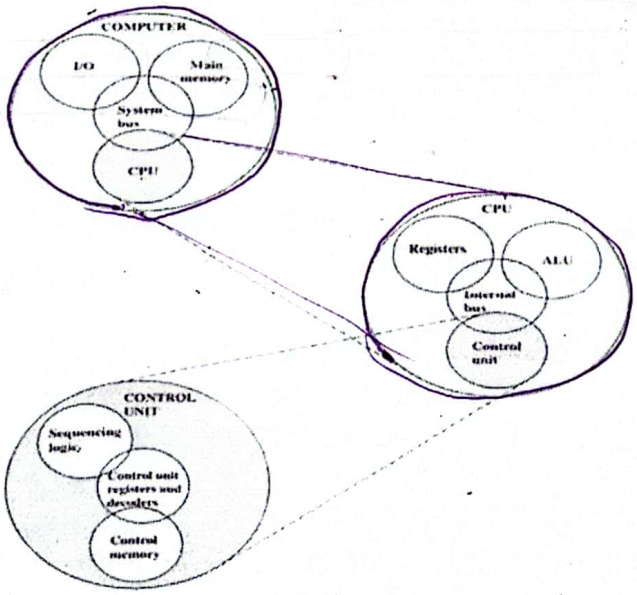
CamScanner

1. why study computer organization and architecture ?
Describe the functions of a computer .

Ans : Studying computer organization and architecture (COA) unlocks a deeper understanding of how computers work, laying a valuable foundation for many technical pursuits . Here are some key reasons why studying computer organization and architecture is important:

- Fundamental Understanding: Computer organization and architecture provide a fundamental understanding of how computers execute instructions, store and retrieve data, and communicate with other devices.
- Programming Efficiency: Understanding the architecture of a computer helps programmers write more efficient and optimized code.
- Hardware-Software Interface: It bridges the gap between hardware and software.
- System Design: For professionals involved in system design, architecture knowledge is vital.
- Performance Optimization.
- Hardware Development.
- Future Technological Advancements..

Here I describe the structure and functions of a computer.



Operating environment

movement

control

Storage

Process

PS mc

Function
- Data processing
- Data storage
- Data movement
- control

— CPU
— I/O
— memory — control

The main components of the basic structure of computers are the control processing unit (CPU), an input unit, memory unit, control unit, and output unit. Ans. The main functions performed by computers based on their basic structure include output, input, storage, and processing.

2. Write short notes (any tow)

i. Dual core ii. Pentium 4 iii. Core i7

64 64 64

Ans :

Pentium 4: The granddaddy of the bunch, released in 2000, Pentium 4 reigned supreme for a while with its high clock speeds. However, its single-core design struggled with multi-tasking and complex workloads. High power consumption and heat output weren't its best friends either.

Dual core: Pentium 4's successor (2005), dual core processors brought revolutionary change. Having two independent cores doubled the processing power, significantly improving performance for multi-threaded tasks like video editing and gaming. While not as powerful as later generations, they offered a sweet spot in terms of performance and affordability.

Core i7: Launched in 2008, Core i7 marked a giant leap forward. These CPUs boast multiple cores (often 4 or more), hyper-threading technology (allowing each core to handle multiple threads), and advanced instruction sets, making them absolute powerhouses. They excel at resource-intensive tasks like high-end gaming, 3D rendering, and scientific computing.

Performance Summary:

Single-core performance: Core i7 > Dual Core > Pentium 4

Multi-tasking performance: Core i7 > Dual Core >> Pentium 4

Power consumption: Core i7 < Dual Core < Pentium 4

Price: Core i7 > Dual Core > Pentium 4

Pentium 4: A relic of the past, suitable for basic tasks on older machines.

CamScanner

Dual Core: A budget-friendly option for moderate workloads and some multi-tasking.

Core i7: The ultimate choice for demanding users who need raw processing power.

3. Distinguish between computer organization and computer architecture ? List and briefly define the main structural components of a computer .

Difference between computer architecture and computer organization---

what computer does

how it does

Computer Architecture	Computer Organization
1. To integrate the components to build a computer system to achieve desired level of performance.	1. Design of components and functional blocks using which computer systems are built.
2. Refers to the attributes of system visible or accessible to programmer.	2. Refers to operational units of the computer.
3. These attributes have a direct impact on the logical execution of a program.	3. How these operational units are interconnected? .
4. Instruction set, Data Types, Memory Addressing Modes, and I/O Mechanism.	4. Includes attributes that are transparent to programmers.
5. Related to software.	5. Control signal, interface between memory & peripherals.
6. Analogy: Architects work.	6. Related to hardware.
	7. Analogy: Civil Engineers Work.

deals with hardware

program view architecture

Arch. fixed first for design

Von-neumann Harvard

deals with performance

implementation of architecture is organization

org. decided after arch.

Stack organiz

organize accom

List the computer component---

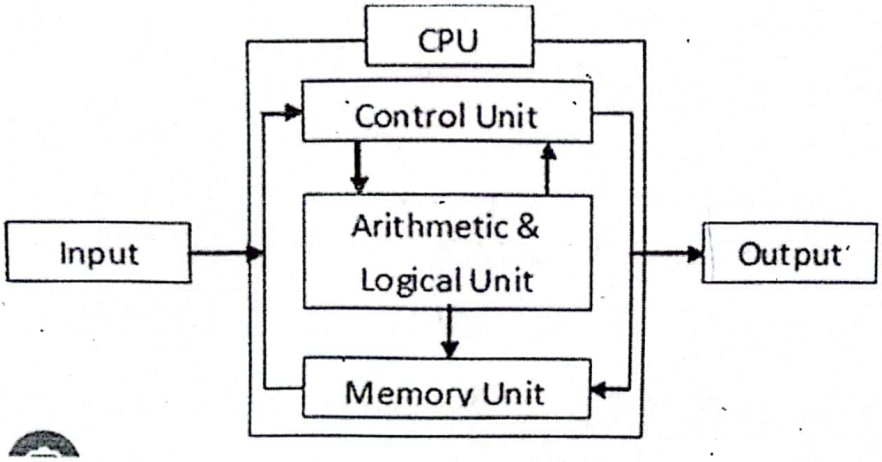
1. Central Processing Unit (CPU).
   →Arithmetic logic unit (ALU)
   →Control unit
   →Register
2. Memory.
   →Main memory (RAM)
   →Secondary memory (SSD)
3. Input / Output device.
   →Keybord,mouse,etc.
4. System bus .

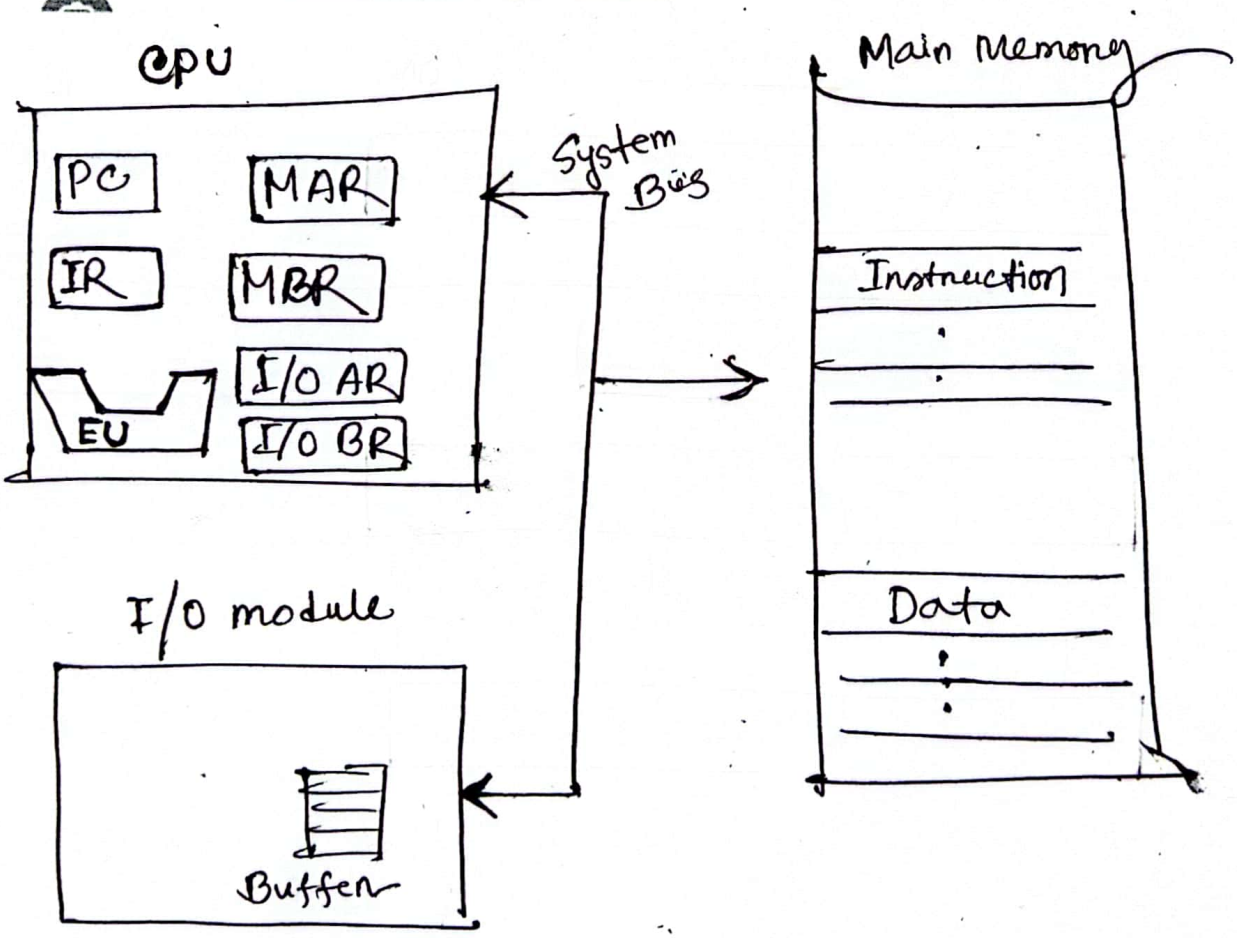
CamScanner

→ Address bus , Control bus , System bus.

5. Why do we need to study computer organization and architecture

4. Illustrate the basic elements of a digital computer .





SUB: question analysis

SAT SUN MON TUE WED TH

DATE: / /

(1) Write down the elements of a machine Instruction. Draw and describe the instruction cycle state Diagram →

Ans:→ Each instruction must contain the required information by the processors for execution. The elements of a machine instruction are as follows →

(1) Operation code → specifies the operation to be performed. The operation is specified by a binary numbers, known as the operation code or opcode.

Add হলে নাকি MOV হবে

(2) Source operand reference → The operation may involve one or more source operands. Operands is input for the operation.

(3) Result operand reference → The operation may produce a result.

(4) Next instruction reference → It tells the processors where to fetch the next instruction.

ADD AX BX

AX ← AX+BX

pc

CamScanner

CamScanner

SUB: _____ SAT SUN MON TUE WED THU FRT DATE: / /

instruction cycle state diagram →

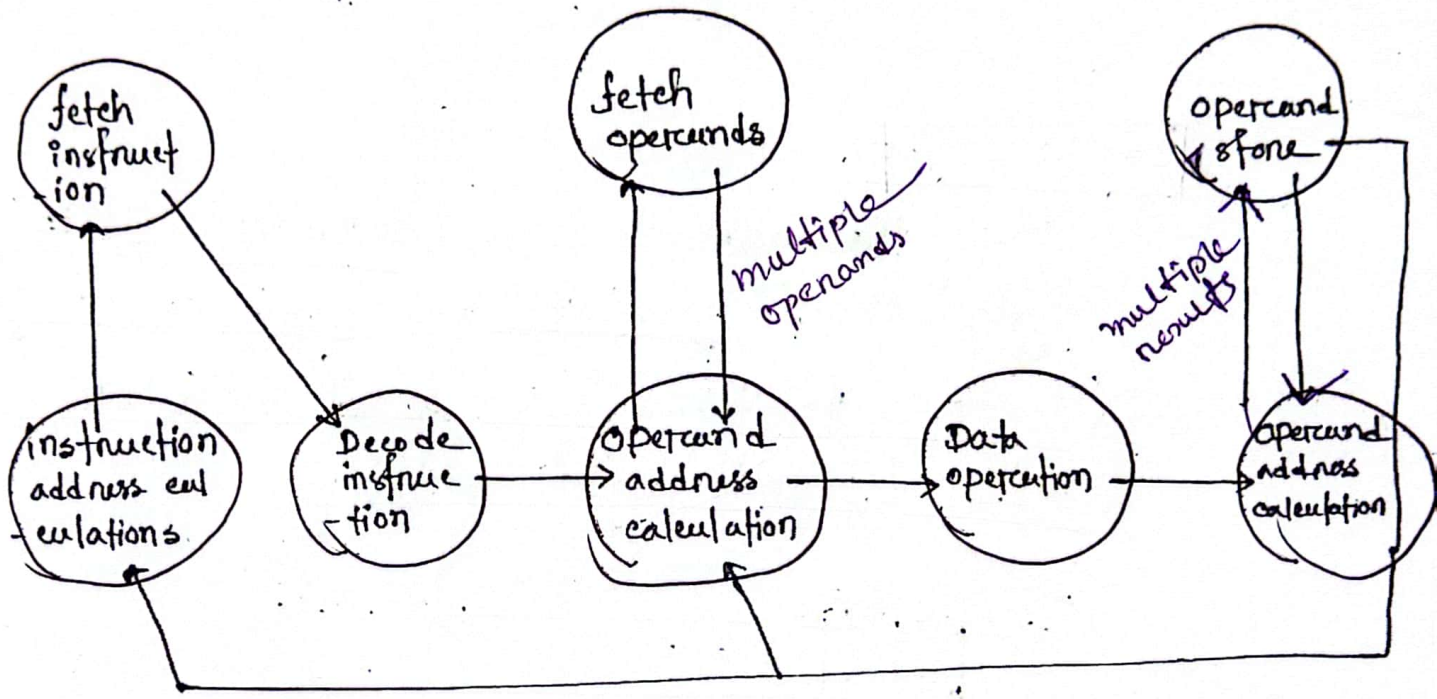


figure: Instruction cycle & state diagram.

Source and result operands can be in one of four areas →

(i) Main or virtual memory.
(ii) processor registers.
(iii) Immediate.
(iv) I/O device.

বইটুকু যে Question এ লাগাবেন।

SUB:

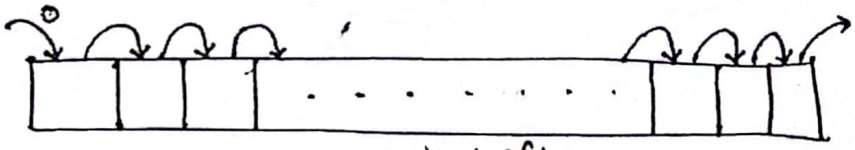
DATE: / /

Difference between arithmetic shift and logical shift →

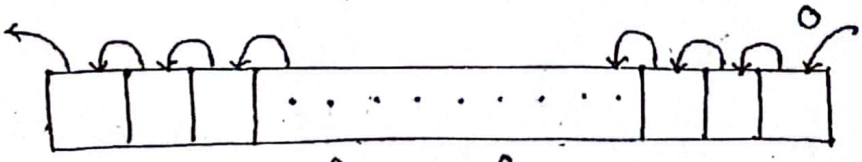
Logical shift →

→ Treats the numbers as a sequence of bits, regardless of weathers it's signed ors unsigned.

→ Vacant positions are filled with zeros on both left and right shifts.



Right shift



Left shift

~~Ans~~ **Arithmetic shift →**

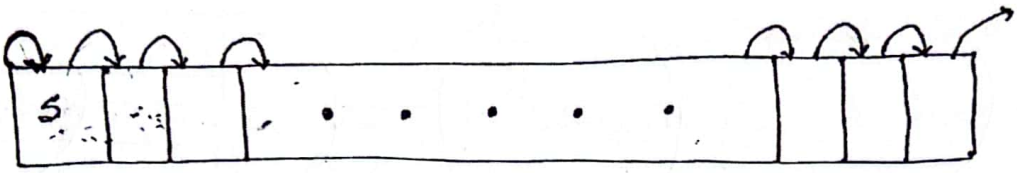
→ Preserves the sign bit of the numbers, ensuring consistency afters the shift.

→ On left shift, vacant positions are filled with ~~zeros like a logical shift~~.

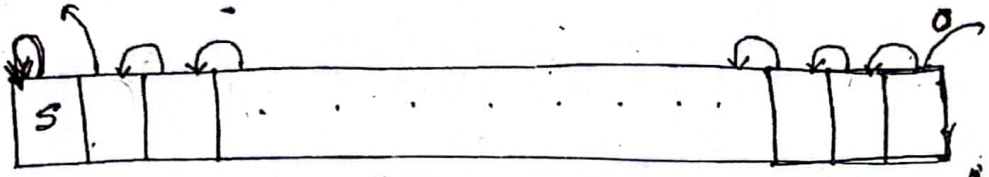
Sign bit. Sign bit unchanged.

→ On right shift

- for positive numbers, vacant position are filled with zeros.
- for negative numbers, vacant positions are filled with the value of the original sign bit.



Right shift



left shift.

1101 → 1
1110

Define Register addressing →

Register addressing is similar to direct addressing. The only difference is that the address field refers to a register rather than a main memory address.

B.A = R

↓ Effective address

↓ Register

ADD AX, BX

* effective address is set to the value currently stored in the Register R.

CS CamScanner

CS CamScanner

SUB:

SAT SUN MON TUE WED THU FRI

DATE: / /

→ very fast execution.

→ very limited address space.

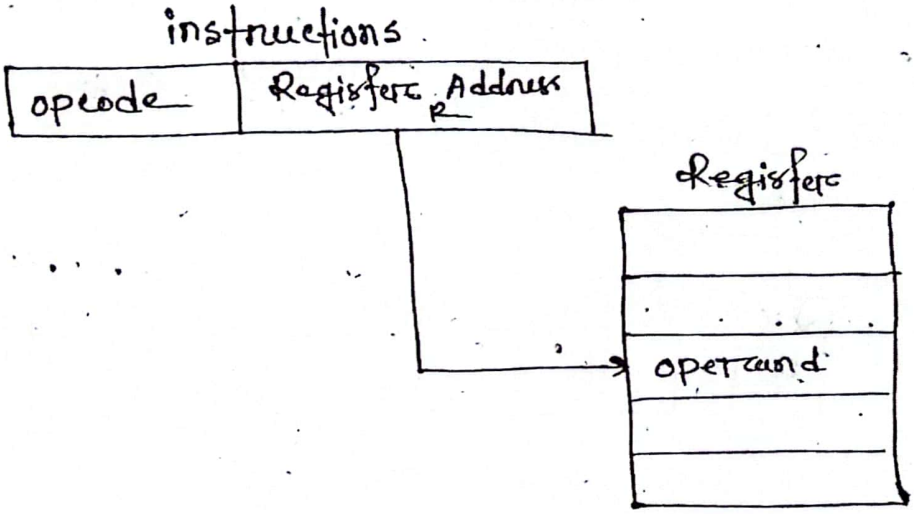


figure: Register addressing.

Q which issues are responsible for designing instruction set?

The most important of these fundamental design issues include the following.

① Operation Repertoire → How many & which operations to provide

② Data type → The various types of data upon which operations are performed

MOV AX,BX

SUB.

DATE / /

③ Instruction fromate → Instruction length, numbers of address, size of various field.

④ Registers → Numbers of processors Registers

⑤ Addressing → The address of operand is specified.

These issues are highly interrelated and must be considered together in designing an instruction sets.

Explain different types of operations →

several types of operations are found on all machine are following

1. Data transfers → specifies that the data are transfed from / to registers or memory. (MOV)

2. Arithmetic → Indicate operations of add, subtract, multiply and divide.

3. Logical → AND, OR, NOT, X-OR

4. Conversion → Translet, convert.

5. I/O → Read, write, start I/O, Test I/O.

CS CamScanner

CS CamScanner

SUB: DATE: / /

6. ~~System control~~ →

7. Transfer control → Jump, Return, Execute, Skip.

Perform arithmetic left shift (10100110) for 3 bits and right rotate (10100110)

10100110
11001100
10011000

Left shift

Given that,

	1 0 1 0 0 1 1 0	initial
hold sign bit —	(1) 1 0 0 1 1 0 0	1 bit L-shift
sign bit —	(1) 0 0 1 1 0 0 0	2 bit L-shift
sign bit —	(1) 0 1 1 0 0 0 0	3 bit L-shift

Right Rotate —

Given than,

1 0 1 0 0 1 1 0

SUB:

SAT SUN MON TUE WED THU FRI

DATE / /

0 1 0 1 0 0 1 1 — 1 bit right shift.

1 0 1 0 1 0 0 1 — 2 bit right shift.

1 1 0 1 0 1 0 0 — 3 bit right shift.

AL
AX
EAX
RAX

what do you mean by variable length addressing?

variable length addressing (VLA) refers to a memory addressing technique where the size of an address field can vary to accomodate different lengths of memory blocks or instructions. It's primarily used in computers architecture.

why it is necessary (variable length addressing):

variable length addressing is necessary for

1. Optimizing memory.
2. Enhancing instruction set flexibility.
3. supporting dynamic memory allocation.
4. Enable data compression.
5. Efficient network comunication.

CamScanner

SUB:

DATE: / /

"Each operation needs operand" Justify this statement.

The statement "Each operation needs operand" refers to the fact that in computing and programming, most operation or instructions require one or more operands to perform the intended computation or action. An operand is a value on which the operation acts.

for example

① Unary operation → Require one operand. like X++ or X--, Here X is unary operand.

INC AX
ADD AX, BX

② Binary operation → Required two operands. In the expression a+b both 'a' and 'b' are operands.

③ Ternary operation → Required three operands. like,

condition ? true : false
(I) (II) (III)

• ADD BX, AX
operand

CamScanner
CamScanner

SUB:

SAT SUN MON TUE WED THU FRT

DATE: / /

Various categories of operand →

The most important categoris of operand are —

① **Addresses** → It hold the address of required operand.

② **Numbers** → There are three types of numbers that are used in computer system. They are —

(i) Binary integers ✓

(ii) Binary floating ✓

(iii) Decimal. ✓

③ **Characters** → A common form of data is text or characters string. ('A', 'b', '2')

④ **Logical data** → Logical data refers to information or data that is organized, represented and manipulated based on principle of logic. Logical data constructed with physical data.

CamScanner

SUB:

SAT SUN MON TUE

DATE: / /

Advantage & disadvantage of variable length addressing →

⊛ **Advantages** of variable length addressing –

→ Memory efficiency

→ Instruction sets flexiblity.

→ Dynamic memory allocations.

→ Data compression.

→ Network protocols.

⊛ **Disadvantages** of variable length addressing –

→ Decoding complexity

→ Security consideration

→ Compatibility (সামঞ্জস্য)

→ Debuging challenges.

CamScanner

SUB: SAT SUN MON TUE WED THU FRT DATE / /

Q What are the typical elements of machine instructions →

① **Operation code (opcode)** –
Specifies the fundamental operation to be performed by the processor. Typically occupies few bits at the begining of the instruction.
Example :→ ADD, SUB, MOV, JUMP

② **Operands** –
Indicate the data or memory locations involved in the operation. can be specified by various forms –

1. immediate operands
2. Registers operands
3. Memory operands
4. I/O operands.

③ **Addressing Modes** –
specify how the operands are accessed in memory or registers. addressing mode include

CamScanner

1. Direct addressing.
2. Indirect addressing.
3. Register indirect addressing.
4. Indexed addressing.
5. Relative addressing.

what types of locations can hold source and destination operands →

Here are common types of locations that can hold source and destination operands in computers system —

① **Registers** → ADD AX, BX

Adds BX to AX and stored into Registers AX.

② **Memory** → Memory contain large amount of data and instruction.

MOV AL, [100]

Loads a byte from memory 100 into register

AL.

③ **Immediate values** :→

ADD CX, 5 ; adds 5 to the value in register CX.

④ **I/O ports** →

IN AL, C0h : Reads a byte from I/O port 60h into register AL.

Define instruction format x86 ? why does the variable length instruction format used in CPU operation ?

Instruction format —

opcode	Destination	Source

opcode → operation to be performed by the processors.

Destination → where the operands to be store after execution.

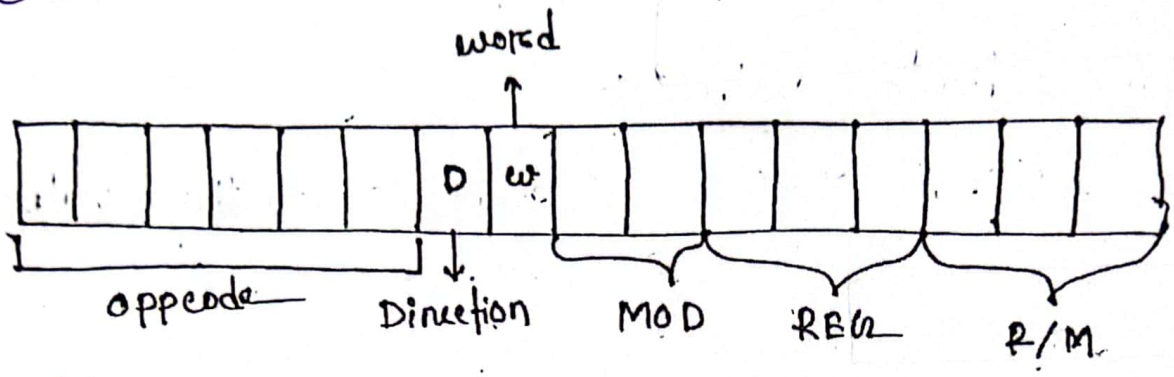
8086

CamScanner
CamScanner

SUB:

DATE: / /

source → where's the operand come from.



2nd Part:

Variable length instruction formates in CPU design are used for various reason, each contributing to the efficiency and flexibility of the instruction set architecture. Here are some key Reasons why variable-length instruction formates are employed in CPUs.

① Improve code density → Variable length instructions are allow for more compact (সংক্ষিপ্ত) representation of instruction.

② Flexibility ~~of~~ in operand specification → Different instruction may have varing

SUB: SAT SUN MON TUE WED THU FRT DATE / /

numbers of operand.

③ **Reduced memory bandwidth Requirements.** → Variable length instruction can reduce the amount of memory bandwidth required during instruction fetch.

④ **Optimized Encoding** → It allows for more efficient encoding.

⑤ **Pipeline Efficiency** → Variable-length instruction support pipeline in CPU.

It is important to note that different CPU architectures may prioritize different aspects, and the choice between variable-length and fixed length instruction formates.

Q If an instruction contain four address, what might be the purpose of each address? 10.3 (exercise)

Microprocessor – 5th Final Exam

Virtual memory

→ RAM এ required জায়গা না থাকলে VM use করবো। → program run করা যায়
(2nd memory use)

Q ① Briefly describe the virtual memory in computer system.

Ans: Virtual memory is an elegent interaction of hardware exceptions, hardware address translation, main memory, disk files, and kernel software that provides each process with a large, uniform and private address space.

Virtual memory mange memory system efficiently with fewer errors.

Q.2 Give the capabilities of virtual memory

Ans: Virtual Memory provides three important capabilities:

① It uses main memory efficiently by treating it as a cache for an address space stored on disk, keeping only the active areas in the main memory and transferring data back and forth between disk and memory as needed.

It simplifies memory management by providin each process with a uniform addresse space.

③ It protects the addresse space for each process from corruption by other process.

Q. 3

Explain page hit and page faults in Virtual Memory.

Ans:

Page Fault: A page fault occurs when a program attempts to access a block of memory that is not stored in the physical memory or RAM.
The fault notifies the operating system that it must locate the data in virtual memory, then transfer it from the storage device, such as HDD or SSD.

Page Hit: If CPU tries to retrive the needed page from main memory, and that page is existed in the Main Memory (RAM), then it is known as "Page Hit".

Scanned with CamScanner

CamScanner

4)

Syllabus

Memory protection mechanism in Virtual Memory

Memory protection is a way to control memory access rights on a computer and is a part of operating systems.

The main ~~purpose of~~ purpose of memory protection is to prevent a process from accessing memory that has not been allocatuted to it. This prevents a bug or ~~other p~~ malware within a process from affecting other processes, or the operating system itself.

Physical memory

Process I :	SUP	READ	WRITE	Address
VP 0 :	NO	YES	NO	PP6
VP 1 :	NO	YES	YES	PP4
VP 2 :	YES	YES	YES	PP2

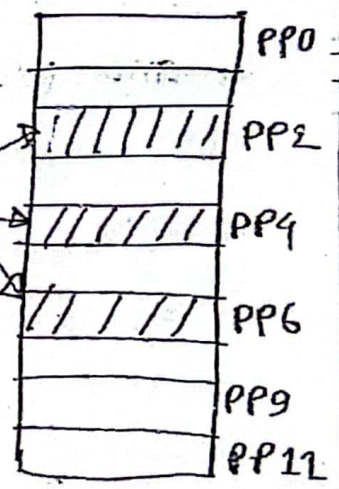


fig - 01

The address translation hardware reads a ~~the~~ PTE (Page Table Entry) each time, the CPU generates an address, it is straightforward to control access to the contents of a virtual page by adding some additional permission bits to the PTE.

In the fig 1, we have added three permission

bits; SUP, READ and WRITE.

The SUP bit indicates wheather processes must be running in kernel (supervisor) mode to access the page.

The READ & WRITE bit controls the ~~read~~ read and write access to the page.

Determine the number of page table entries (PTEs) that are needed for the following combinations of virtual address size (n) and page size (P).

n	$P = 2^P$	# PTEs
16	4K	16
16	8K	8
32	4K	2^{20}
32	8K	2^{19}

Am

$\frac{2^{16}}{4 \times 1024}$

Soln:

We know,

Page Table Entries (PTEs) = virtual address / page size

7696 Page Table

Here, 1st Problem:

Virtual address (n) = 2^{16}

Page size (P) = 4k = 4 × 1024 = 4096 = 2^{12}

∴ Page Table Entries (PTE) = $2^{16} \div 2^{12}$

= 16

2nd Problem:

Virtual address (n) = 2^{16}

Page Size (P) = 8k = 8 × 1024 = 8192 = 2^{13}

∴ Page Table Entries (PTE) = $2^{16} \div 2^{13}$

= 8

3rd Problem:

Virtual address (n) = 2^{32}

Page Size (P) = 4k = 4×1024 = 4096 = 2^{12}

∴ Page Table Entry (PTE) = $2^{32} \div 2^{12}$

= 2^{20}

4th Problem:

Virtual Address (n) = 2^{32}

Page Size (P) = 8k = 8×1024 = 8192 = 2^{13}

∴ Page Table Entry = $2^{32} \div 2^{13}$

= 2^{19}

CamScanner