

II - ASSIGNMENT

(Start Writing From Here)

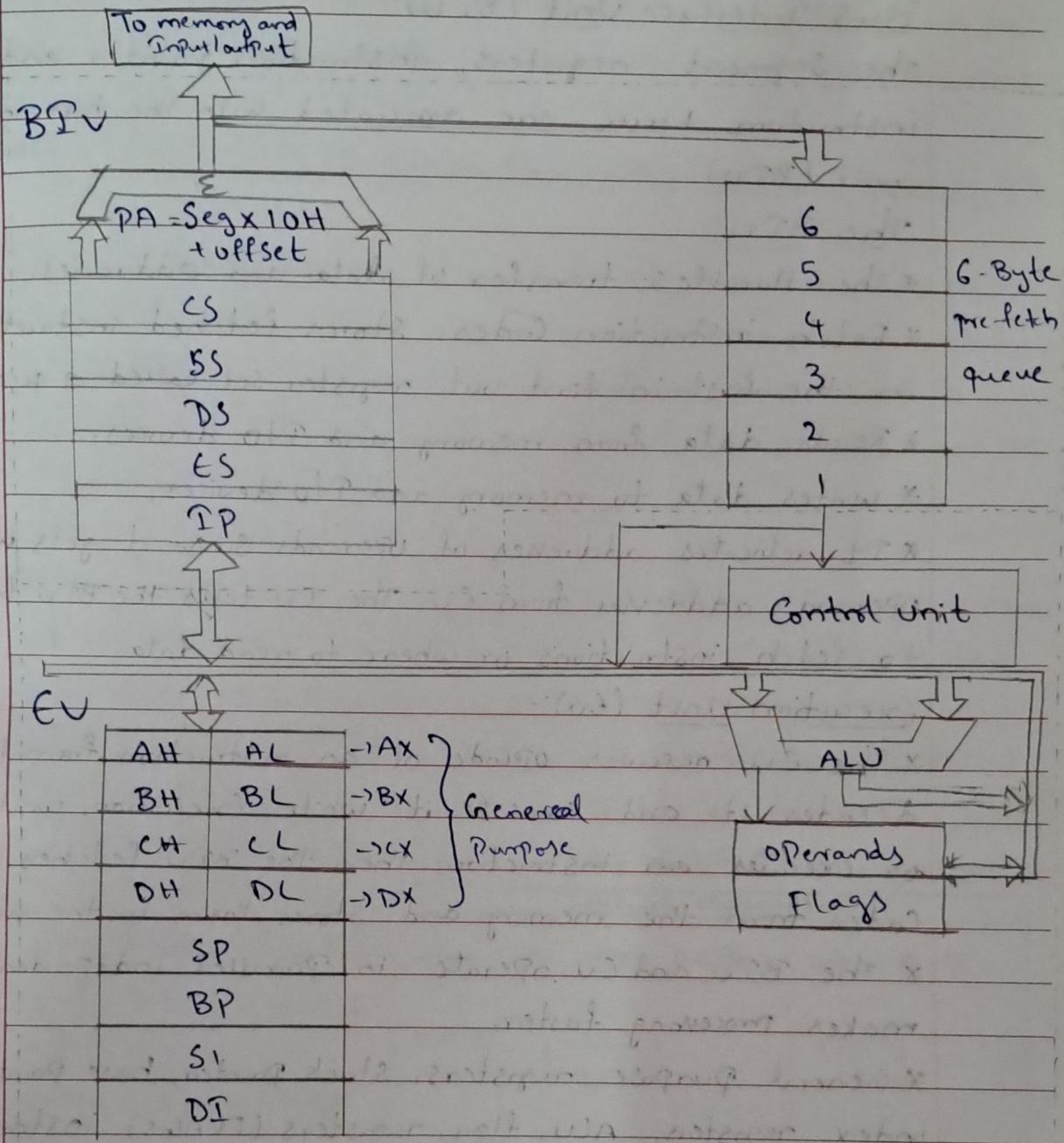
1) With a neat sketch, give the internal architecture of Intel 8086 microprocessor.

A:- The 8086 microprocessor is an 8-bit/16-bit microprocessor designed by Intel in the late 1970s. It is the first member of the x86 family of microprocessors, which includes many popular CPUs used in personal computers.

The architecture of the 8086 microprocessor is based on a complex instruction set computer (CISC) architecture, which means that it supports a wide range of instructions, many of which can perform multiple operations in a single instruction. The 8086 microprocessor has a 20-bit address bus, which can address up to 1MB of memory, and a 16-bit data/bus, which can transfer data between the microprocessor and memory or I/O devices.

The 8086 microprocessor has 2 main execution units: the execution unit (EU) and the bus interface unit (BIU).

Block diagram of 8086 microprocessor



Bus Interface Unit (BIU):-

The Segment registers, instruction pointer and 6-byte instruction queue are associated with the bus interface unit (BIU).

The BIU:-

- * The Handles transfer of data and addresses.
- * Fetch instruction codes, stores fetched instruction codes in the first-in-first-out register set called a queue.
- * Reads data from memory and I/O devices,
- * Writes data to memory and I/O devices,
- * It relocates addresses of operands since it gets un-relocated operand addresses from EU. The EU tells the BIU from where to fetch instructions or where to read data.

Execution Unit (EU):-

- * The EU receives opcode of an instruction from the queue, decodes it and execute it. While execution, unit decodes or executes an instruction, then the BIU fetches instruction codes from the memory and stores them in the queue.
- * The BIU and EU operate in parallel independently. This makes processing faster.
- * General purpose registers, stack pointer, base point and index registers, ALU, flag registers (FLAGS), instruction decoder and timing and control unit constitute execution unit (EU).

2) Draw and explain the block diagram, flowchart to show the hardware for signed magnitude addition and Subtraction.

A:- Addition and Subtraction with Signed - Magnitude Data

→ We designate the magnitude of the 2 numbers by A & B. where the signed numbers are added or subtracted, we find that there are eight different conditions to consider, depending on the sign of the numbers and the operation performed.

* The last column is needed to present a negative zero. In other words, when two equal numbers are subtracted, the result should be +0 & not -0.

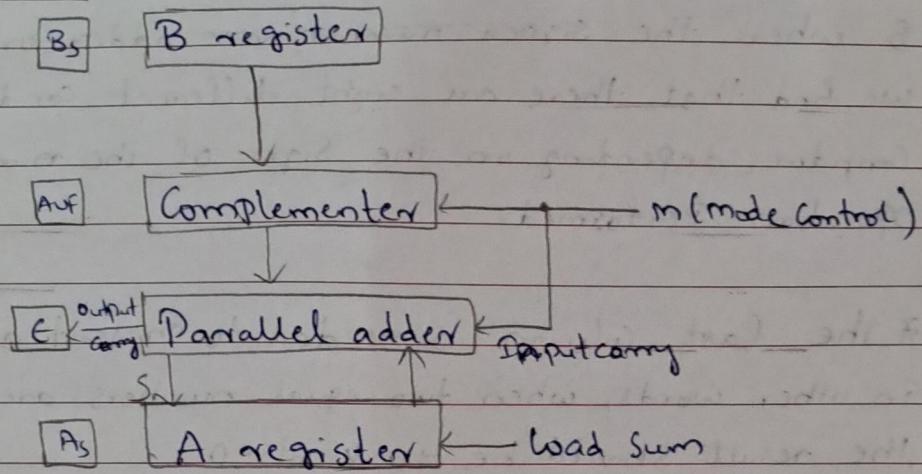
* The algorithms for addition and subtraction are derived from the table and can be stated as follows (the words Parentheses Should be used for the subtraction algorithm).

Operation	Add magnitudes	Subtract magnitudes		
		when $A > B$	when $A < B$	when $A = B$
$(+A) + (+B)$	$+ (A+B)$			
$(+A) + (-B)$		$+ (A-B)$	$- (B-A)$	$+ (A-B)$
$(-A) + (+B)$		$- (A-B)$	$+ (B-A)$	$+ (A-B)$
$(-A) + (-B)$	$- (A+B)$			
$(+A) - (+B)$		$+ (A-B)$	$- (B-A)$	$+ (A-B)$
$(+A) - (-B)$	$+ (A+B)$			
$(-A) - (+B)$	$-(A+B)$			
$(-A) - (-B)$		$-(A-B)$	$+ (B-A)$	$+ (A-B)$

→ when the signs of A & B are Same, add 2 magnitudes and attach the sign of result is that of A.

→ when the Signs of A & B are not Same, Compare the magnitudes and Subtract the smaller number from the larger.

Hardware Implementation for Signed-magnitude Data:-



Flow chart:-

Subtract operation

Add operation

Minuend in A
Subtrahend in B

Augend in A
Addend in B

$A_I \oplus B_I$

$= 11 \quad A_I \oplus B_I = 0$
 $A_I \neq B_I \quad A_I = B_I$

$CAL \leftarrow A + \bar{B} + 1$
AUFLÖ

$CAL \leftarrow A + B$

$\epsilon = 0 \quad \epsilon = 1$

$A < \bar{A}$

$A_I \leq 0$

$A \leftarrow A + 1$
 $A_I \leftarrow A_I$

$A > B$

$\neq 0 \quad A = 0$

END
(result is in A & A_I)

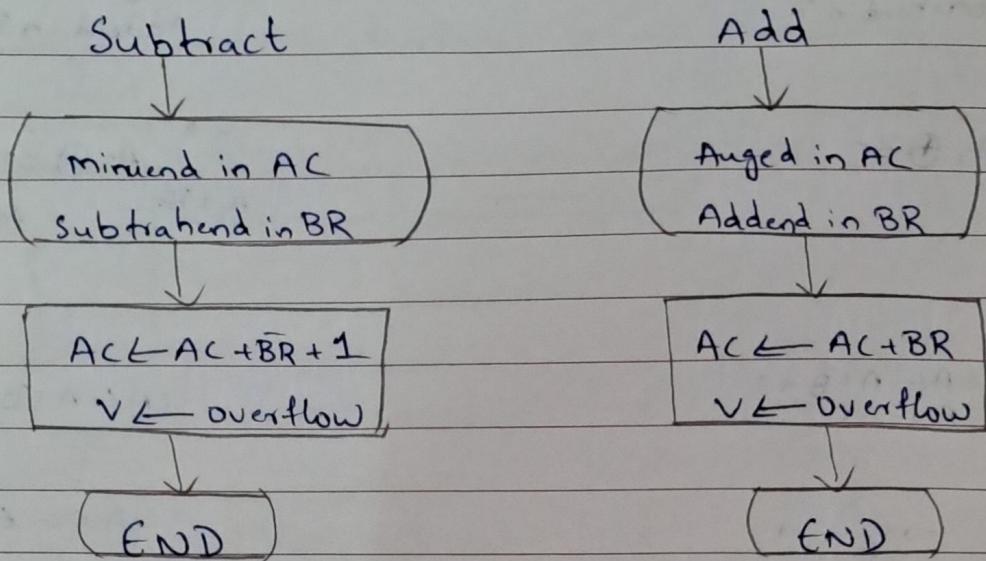
AUFLÖ

$A < B$

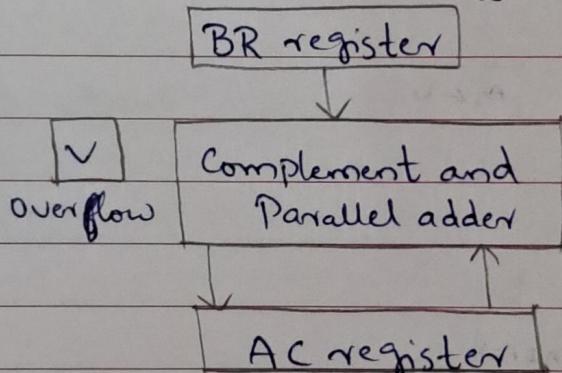
$A = 0$

Flow chart for addition & subtraction

Algorithm for adding and subtracting numbers in Signed 2's complement :-



Addition and Subtraction with Signed 2's complement Data:



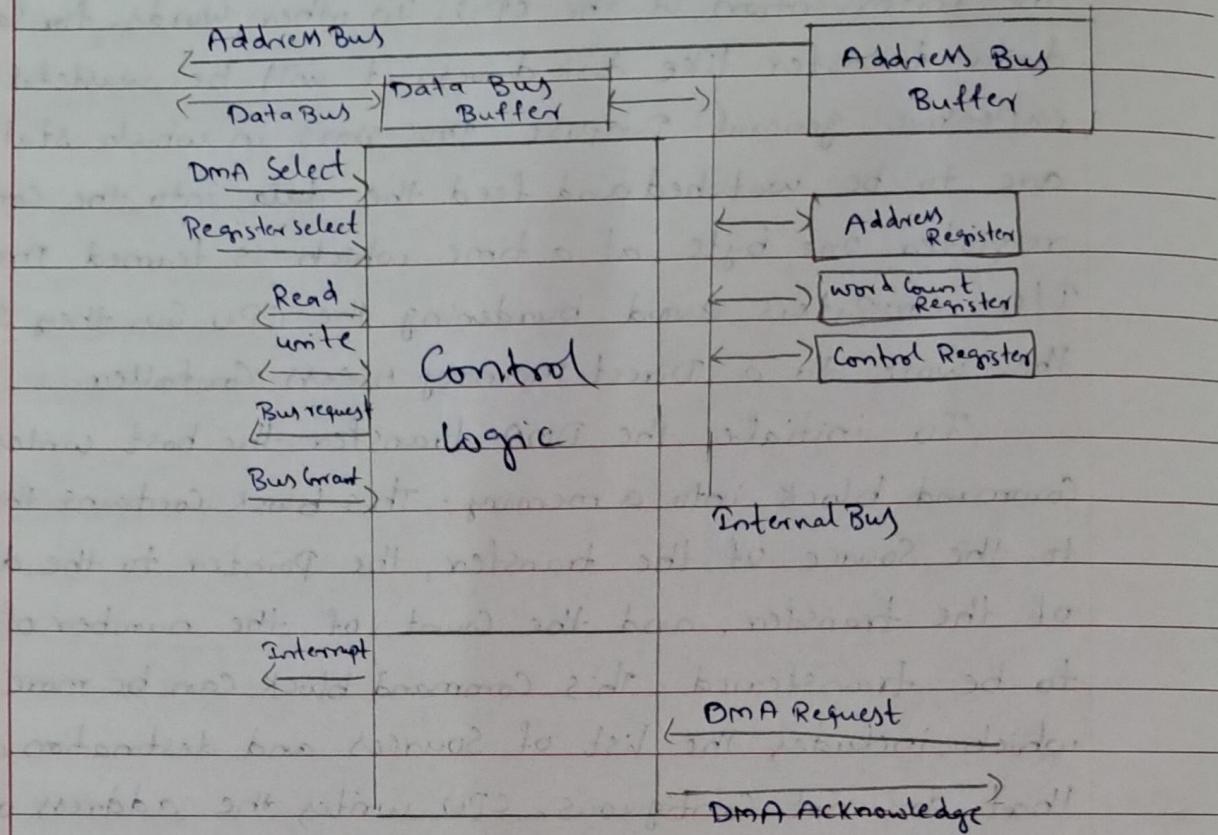
3) Discuss briefly about Direct memory Access .

A:- DMA (Direct memory access) is the special feature within the Computer System that transfers the data between memory and Peripheral devices (like hard drives) without

the intervention of the CPU. In other words, for large data transfer like disk drives, it will be wasteful to use expensive general-purpose processors in which status bits are to be watched and feed the data into the controller register one byte at a time which is termed programmed I/O. Computers avoid burdening the CPU so, they shift the work to a Direct Memory Access Controller.

To initiate the DMA transfer the host writes a DMA Command block into a memory. This block contains the pointer to the source of the transfer, the pointer to the destination of the transfer, and the count of the number of bytes to be transferred. This command block can be more complex which includes the list of sources and destination addresses that are not contiguous. CPU writes the address of this command block and goes to other work. DMA controller proceeds to operate the memory bus directly, placing the address on it without the intervention of the main CPU. Nowadays, simple DMA controller is a standard component in all modern computers.

Block Diagram of DMA:



The mutual understanding between the device controller & DMA controller is performed via pair of wires called DMA request and DMA acknowledge. Let's see what is the role of these wires in the DMA transfer.

Working of DMA Transfer:-

The device controller places a signal on the DMA request wire when a word of data is available for transfer. This cause DMA controller to seize the memory bus of CPU and place the desired address on the DMA acknowledge wire.

Up on successful data transfer the device controller receives the DMA acknowledge and then it removes the DMA request signal.

When the entire transfer is finished, DMA controller interrupts the CPU. This entire process is depicted in the above diagram. DMA controller Seizes the memory bus and CPU momentarily prevented from accessing main memory. Although it can access the data item in its cache, this cycle stealing (seizing the memory bus temporarily and preventing the CPU from accessing it) slows down the CPU computation, shifting the data transfer to DMA Controller generally improves the total system performance. Some of the computer architecture used Physical memory address for DMA, but otherwise virtual addresses (DVMA). Direct virtual memory access performs data transfer b/w memory mapped I/O without the use of main memory.

4) Illustrate memory hierarchy in a computer system using a block diagram?

In the Computer System Design, memory hierarchy is an enhancement to organise the memory such that it can minimize the access time. The memory hierarchy was developed based on a program behaviour known as locality of references. The figure below clearly

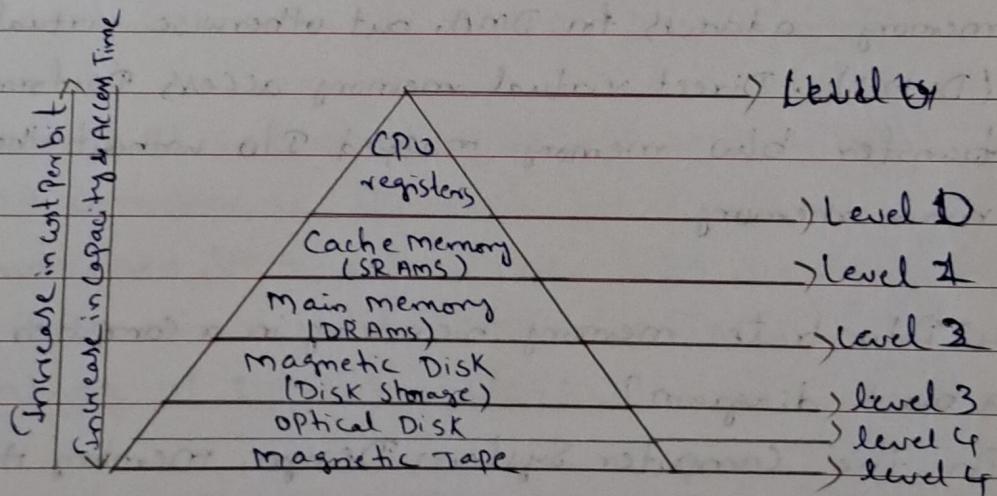
demonstrates the different levels of the memory hierarchy.

Types of memory Hierarchy:-

This memory Hierarchy Design is divided into 2 main types:-

* External memory or Secondary memory:- Comprising of Magnetic Disk, Optical Disk, and magnetic Tape i.e. Peripheral Storage devices which are accessible by the processor via an I/O module.

* Internal Memory or primary memory:- Comprising of Main memory, Cache memory & CPU registers. This is directly accessible by the processor.



Memory Hierarchy Design

1) Memory Hierarchy Details:-

1) Registers:-

Registers are small, high-speed memory units located in the CPU. They are used to store the most frequently used data and instructions. Registers have the fastest access time and the smallest storage capacity, typically ranging from 16 to 14 bits.

2) Cache Memory:-

Cache memory is a small, fast memory unit located close to the CPU. It stores frequently used data and instructions that have been recently accessed from the main memory. Cache memory is designed to minimize the time it takes to access data by providing the CPU with quick access to frequently used data.

3) Main memory:-

Main memory also known as RAM (Random Access Memory), is the primary memory of a Computer System. It has a larger storage capacity than cache memory but it is slower. Main memory is used to store data and instructions that are currently in use by the CPU.

Types of Main Memory:-

* **Static RAM:** Static RAM stores the binary information in flip flops and information remains valid until power is

Supplied. It has a faster access time and is used in implementing Cache memory.

* Dynamic RAM: It stores the binary information as a charge on the capacitor. It requires refreshing circuitry to maintain the charge on the capacitor's after a few milliseconds. It contains more memory cells per unit area as compared to SRAM.

4) Secondary storage:- This includes non-volatile storage devices like Solid State Drives (SSD), Hard Disk Drives (HDD), and optical drives. These are used for long-term storage of data and programs, but they are slower compared to RAM.

5) Magnetic Disk:-

Magnetic Disks are simply circular plates that are fabricated with either a metal or a plastic or a magnetized material. The magnetic disks work at a high speed inside the computer and these are frequently used.

6) Magnetic Tape:-

Magnetic Tape is simply a magnetic recording device that is covered with a plastic film. It is generally used for the backup of data. In the case of a magnetic tape, the access time for a computer is a little slower and therefore, it requires some amount of time for accessing the strip.

5) Explain about parallel processing.

Parallel processing can be described as a class of techniques which enables the system to achieve simultaneous data-processing tasks to increase the computational speed of a computer system.

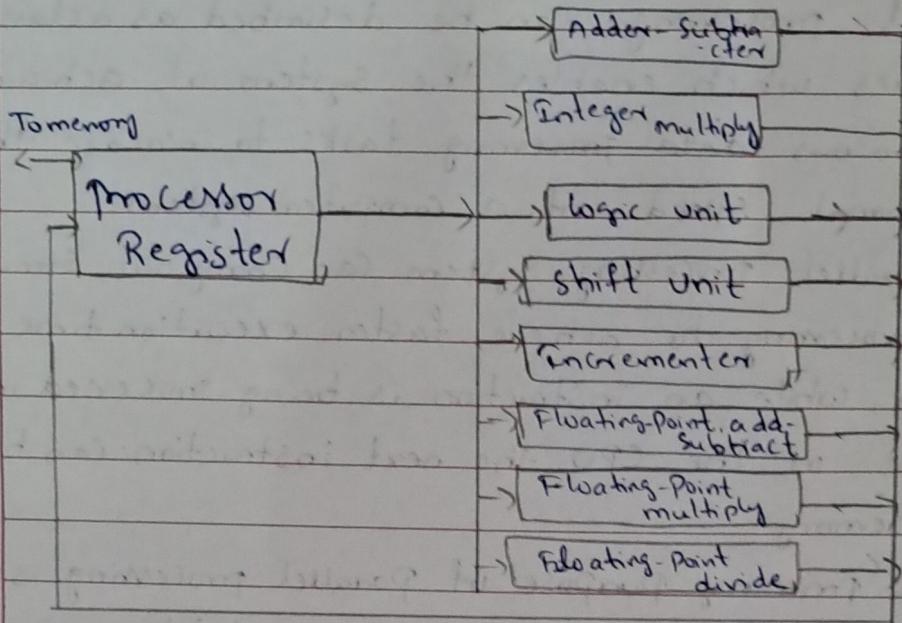
A Parallel processing system can carry out simultaneous data-processing to achieve faster execution time. For instance, while an instruction is being processed in the ALU component of the CPU, the next instruction can be read from memory.

The primary purpose of parallel processing is to enhance the computer processing capability and increase its throughput, i.e. the amount of processing that can be accomplished during a system given interval of time.

A Parallel processing system can be achieved by having a multiplicity of functional units that perform identical or different operations simultaneously. The data can be distributed among various multiple functional units.

The following diagram shows one possible way of separating the execution unit into eight functional units operating in parallel.

The operation performed in each functional unit is indicated in each block of the program.



- * The adder and integer multiplier performs the arithmetic operation $A \times B$ with integer number.
- * The floating-point operations are separated into three circuits operating in parallel.
- * The logic, shift, and increment operations can be performed concurrently on different data. All units are independent of each other, so one number can be shifted while another number is being incremented.