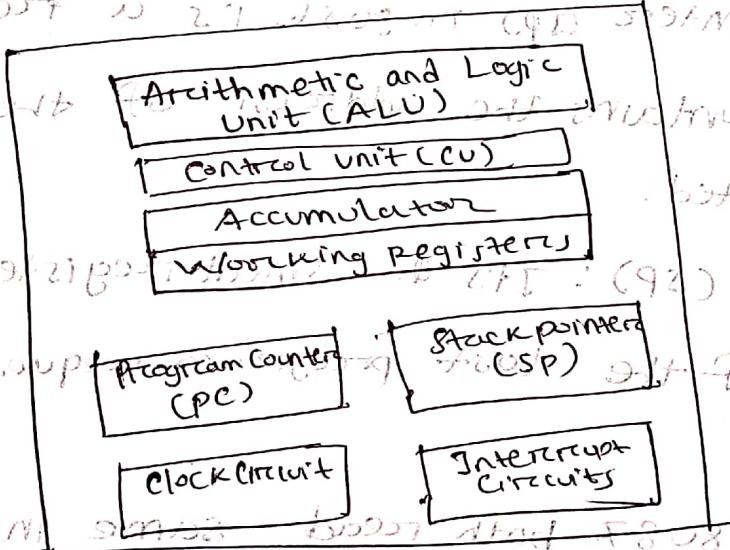


Ques. 2) Define bus architecture with 3 (3)
Ans. A microprocessor: A microprocessor is a multipurpose programmable, clock driven, register-based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions and provides results as output. A CPU built into a single VLSI chip is called a microprocessor.



International Institute of Figured Block Diagram of MP

Containing elements of MP: \$808 bns \$852 bns up

① Arithmetic & Logic Unit (ALU): It's the place where actual computation take place. It consists there circuits which perform arithmetic operations over data received from memory and capable to compare numbers.

(i) Control unit (CU): The coordination and control of the computer is the sole responsibility of CU. It interprets the instructions fetched into the computers and determines what data if they are needed, where it is stored and tells to store the result of the operations. It also tells which operations are done on which data.

(ii) Registers: It's a temporary storage locations inside CPU that hold data and address.

(iii) Program Counter (PC): It's commonly known as IP. Instruction pointer (IP) in 8086 is a register in CPU which contains the address of the next instruction to be executed.

(iv) Stack pointer (SP): It's a small register that stores the address of the last program request in a stack.

b. 8086 and 8087 both read same instructions

from memory and put in their internal instruction queue.

8086 and 8087 both decode each instruction that come into their queue

& then decode only instruction from their queue

and find that if it is an 8086 instruction then

8087 simply treats the instruction as an NOP.

and when it finds that it's an 8087 instruction,

the 8086 simply treats the instruction as an NOP.

That means each processor decodes all the instructions in the fetched instruction byte stream but executes only its own instruction. All the 8087 instruction codes have 11011 as the most significant bit of their first code byte.

C. PSW: Program Status Word

The 8051 has a 8 bit PSW register which is also known as Flag register. In the 8-bit register only 6-bits are used by 8051. The two unused bits are user definable bits. In the 6 bits, four of them are conditional flags. They are

- i. Carry (CY)
- ii. Auxiliary Carry (AC)
- iii. Parity (P)

IV. Overflow (OV)

These flags indicate some conditions that resulted after an instruction was executed. They are available to the user for general purpose.

The bits R50 and R51 are used to select bank registers.

PSW1	PSW6	PSW5	PSW4	PSW3	PSW2	PSW1	PSW0	R51	R50	Register
01	01	01	01	01	01	01	01	00	00	00H - 02H
02	02	02	02	02	02	02	02	01	01	02H - 03H
03	03	03	03	03	03	03	03	11	11	03H - 04H

PSEN: Program store enable. High enable bit.

Pin: 29 is an output pin which goes through logic driver with the TIA's output pin and is connected to the OE pin of the ROM chips used to enable external program memory. If we use an external ROM for storing program, then logic 0 appears on it, which indicates MC to read data from the memory.

29 is also

ALE : Address latch enable

break address most part of 8051

Pin: 30 is an output pin and is active high. It is used for providing both address and data. The A7 is used for demultiplexing the address and the data by connecting to the Gpin of the 74LS373 later.

EA: External access

Pin: 31

It is connected to GND to indicate the code is stored externally. If multiple memories are has to be used then the application of logic 0 to this pin instructs the MC to read data from both memory.

first internal or data entered

[1] [0] [1] [0] [1] [0] [1] [0] [1] [0]

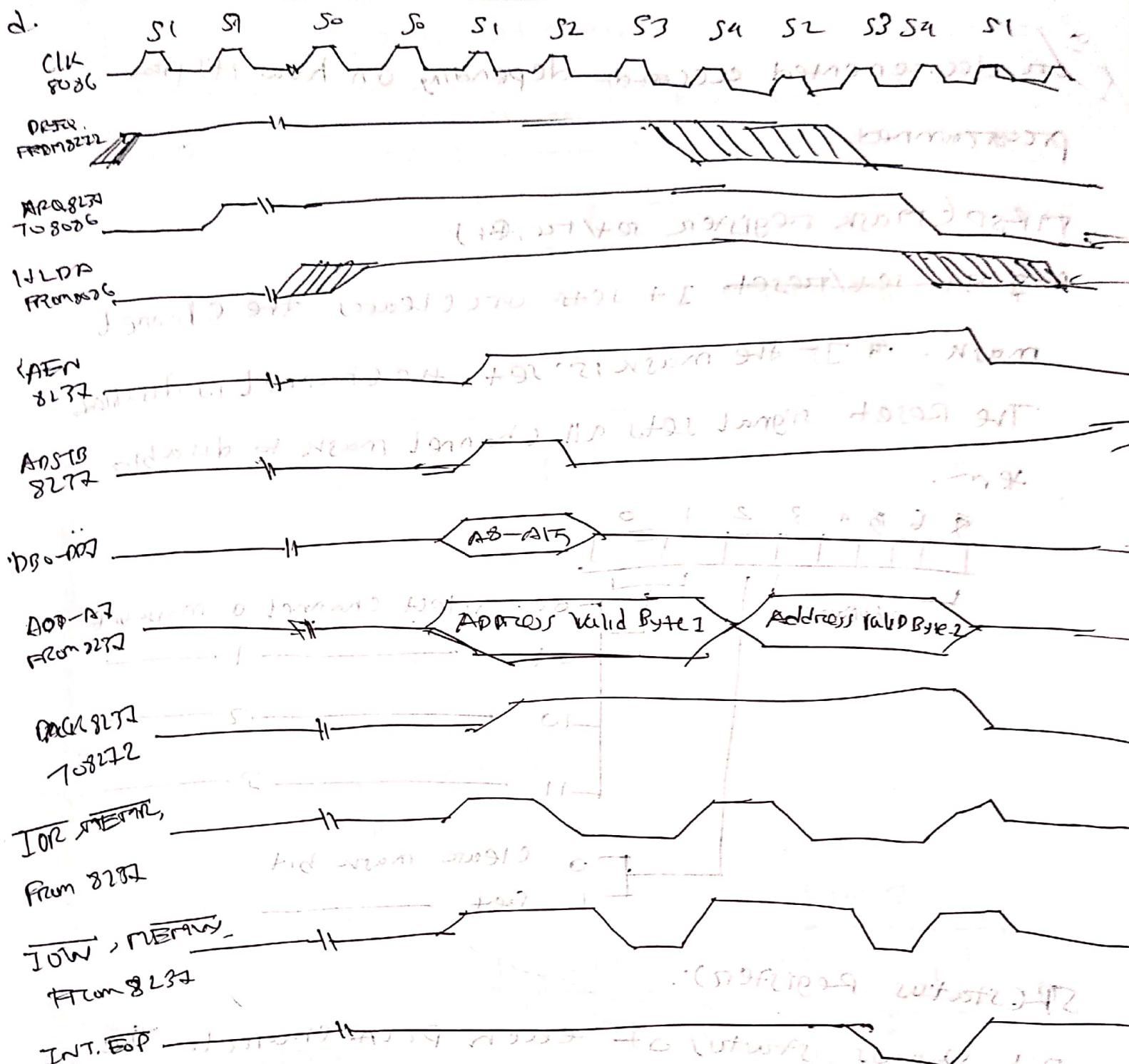


Fig: Timing Diagram for 8272 DMA Transfer

CAR (Current Address Register):

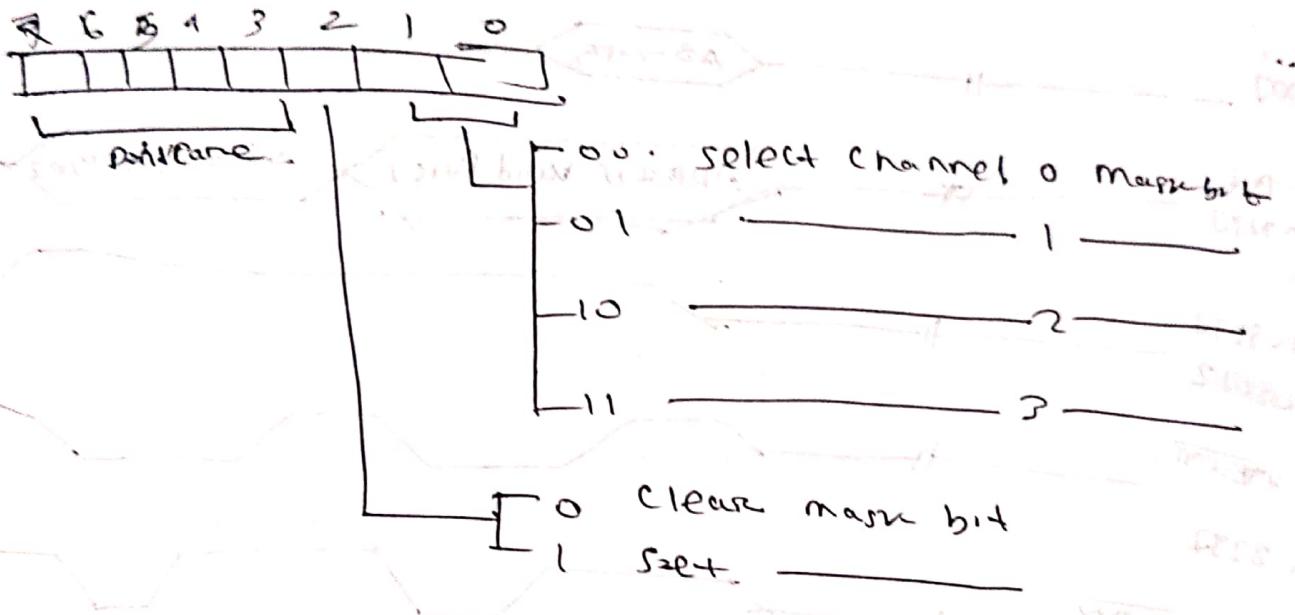
It holds 16 bit memory address used for the DMA transfer.

Each channel has its own current CAR for this purpose. When a byte of data is transferred during a DMA operation, the address field is incremented.

or decremented according depending on how it's programmed

MRSP (Mask register set/reset):

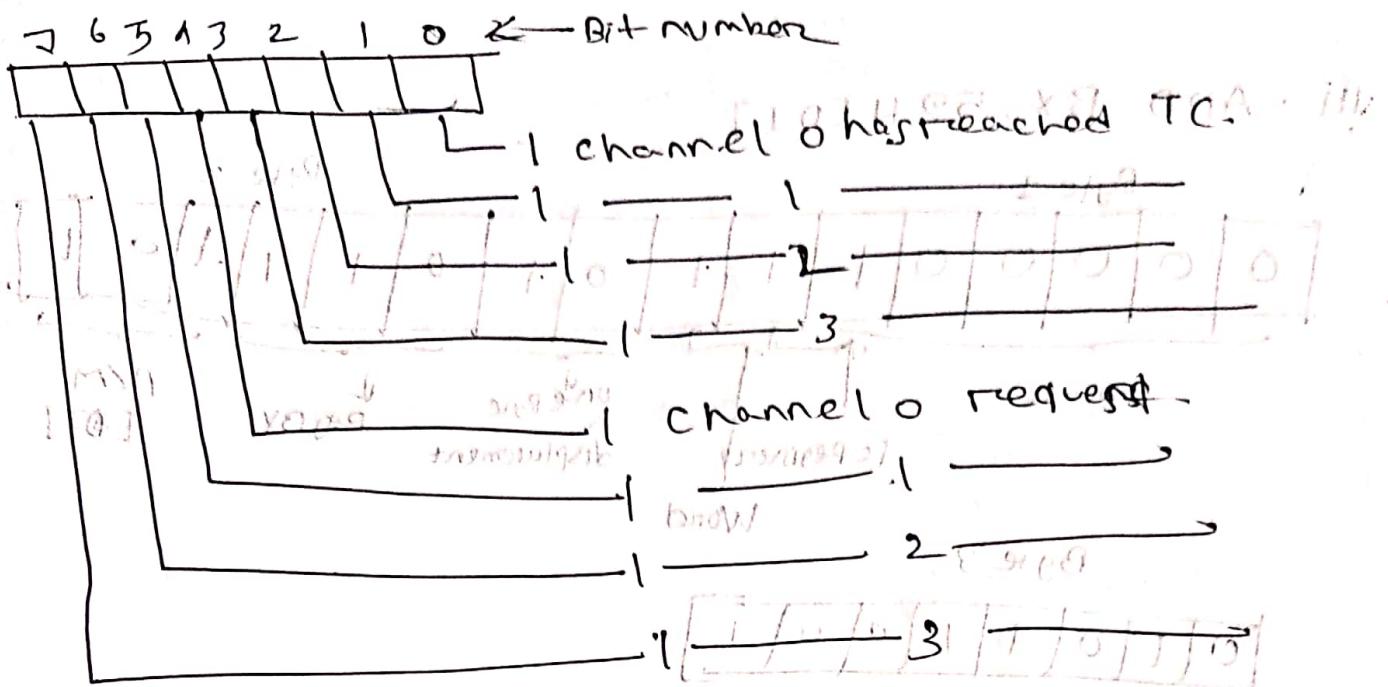
It sets/reset 7+ sets or clears the channel mask. If the mask is set, the channel is disabled. The reset signal sets all channel mask to disable form.



STR (status Register):

It shows status of each PMSI channel. The TC bits indicate if the channel has reached its terminal count (transferred all bits by).

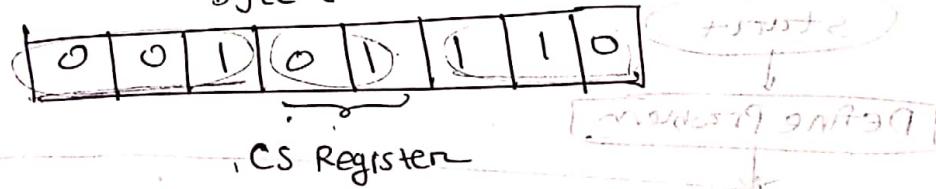
When the terminal counter reached, the DREQ transfer is terminated. The request bit indicates whether the PMSI was granted or not. With the DREQ is given, the DTR is also triggered. It is used to indicate if the channel is busy or not. The DTR is set to 1 when the channel is busy and cleared when the channel is free.



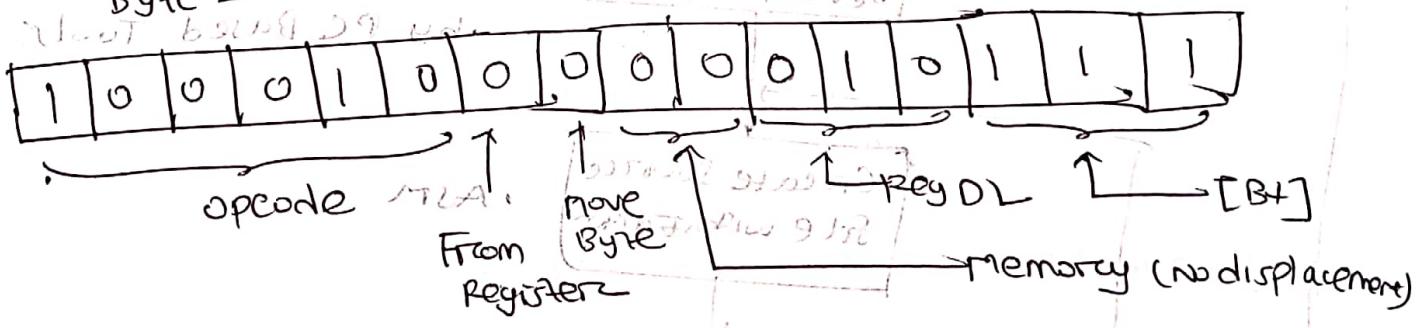
2. a) mov CS:[BX], DL

Segment override prefix.

Byte 1

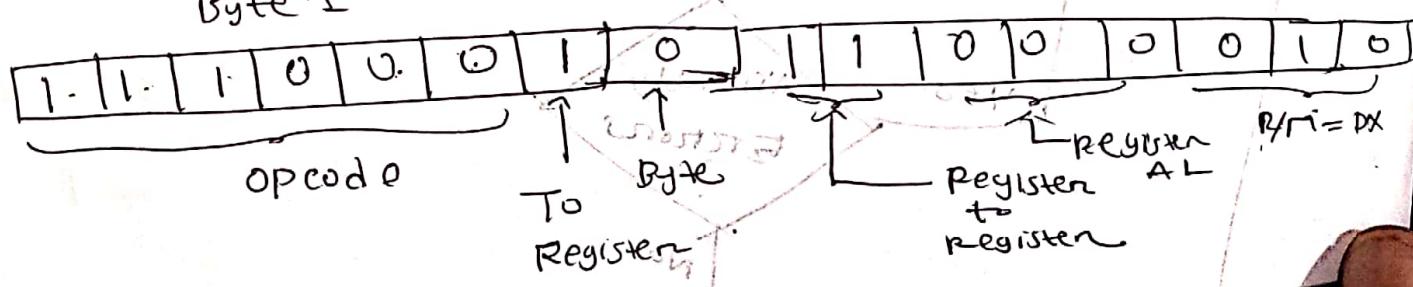


Byte 2

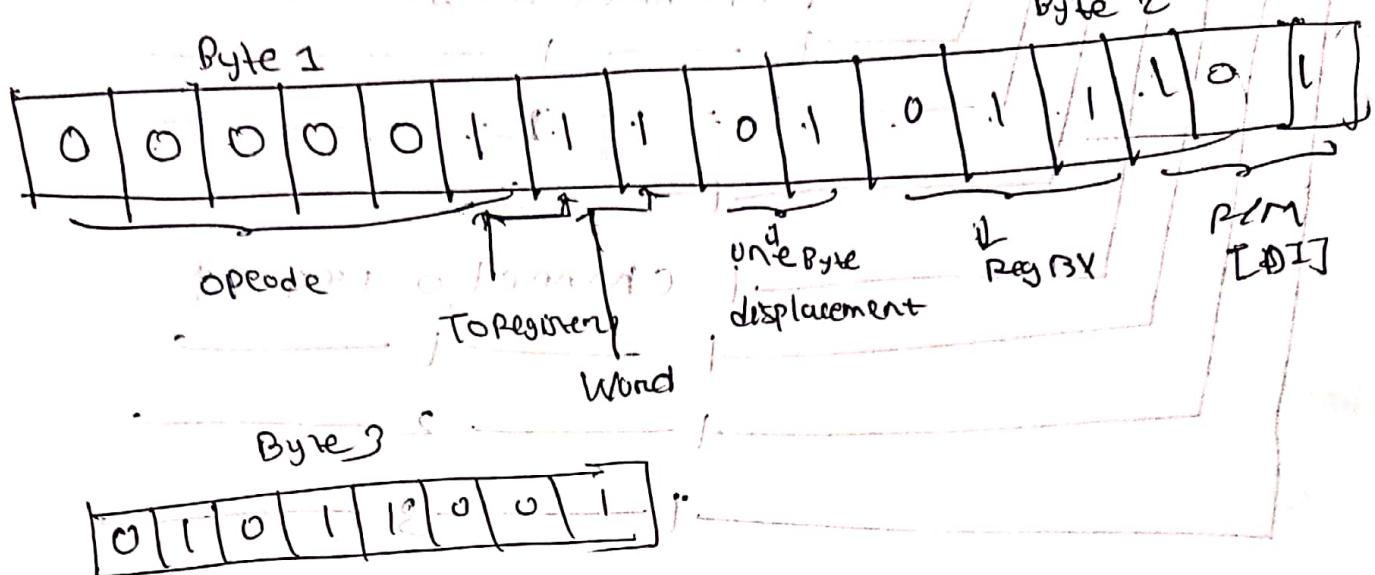


ii) IN AL, DX.

Byte 1

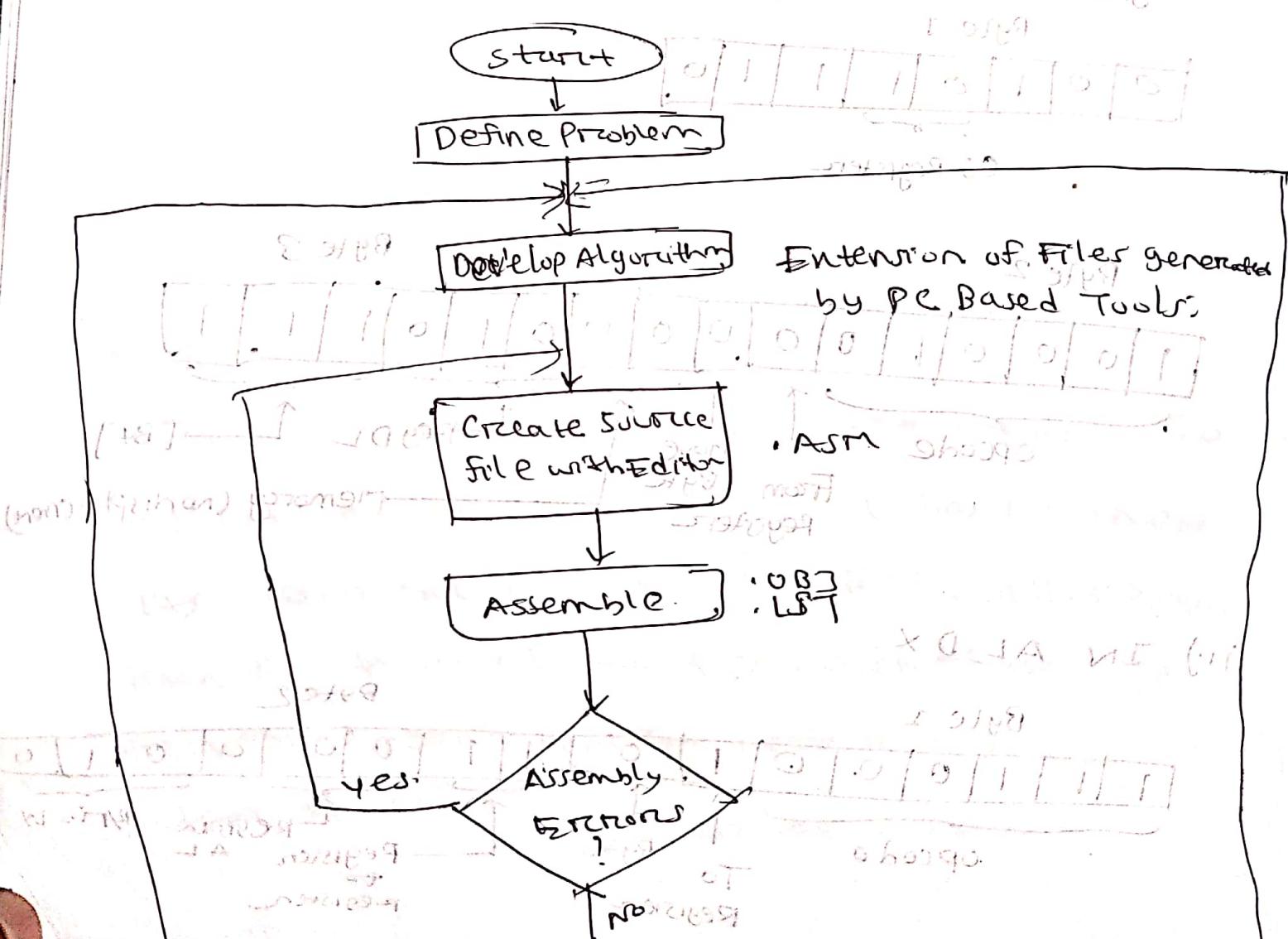


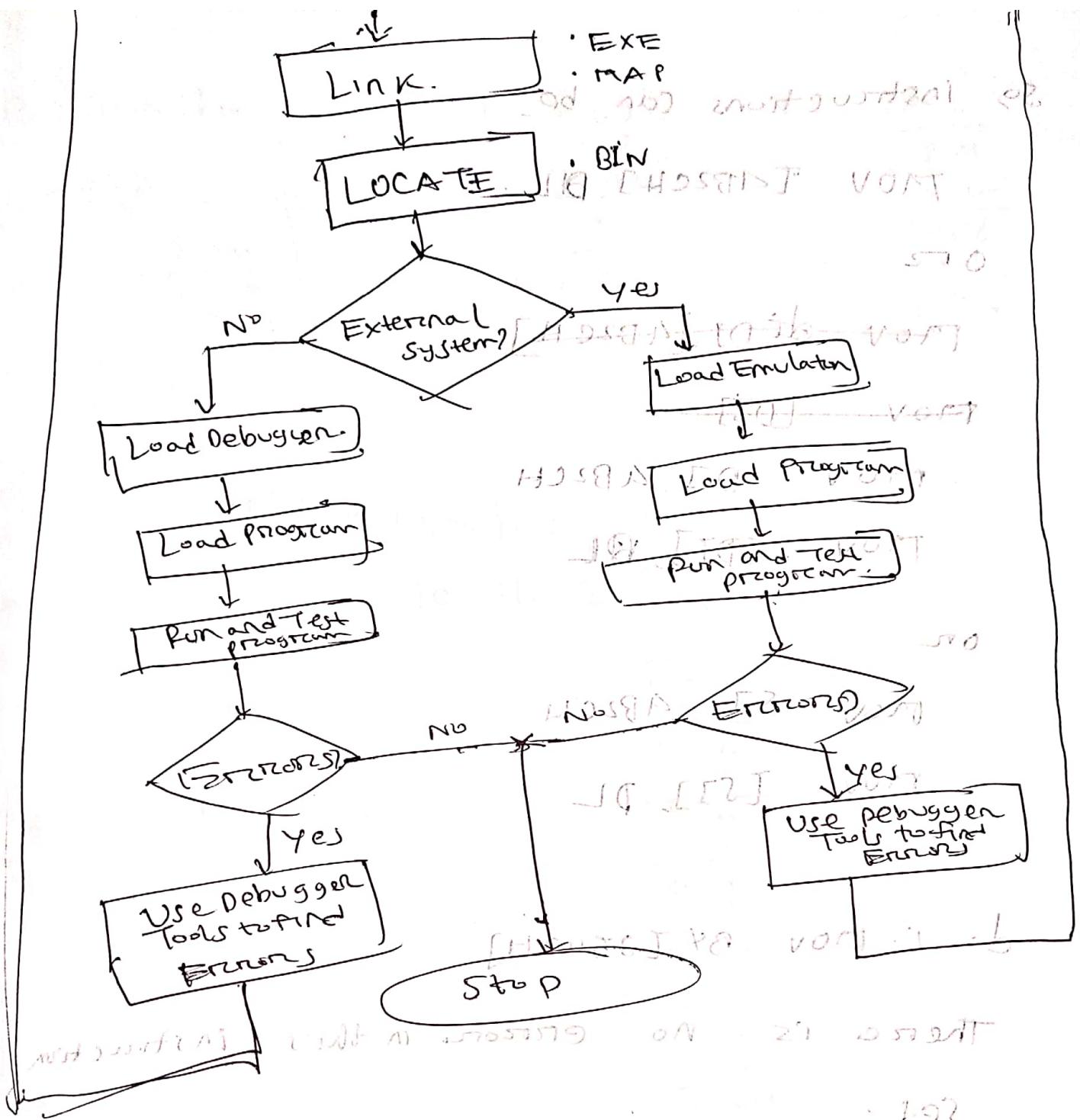
iii. ADD BX, 59H[D1]



$$\text{Displacement} = 59H$$

b. Program Development Algorithm.





c. IF the data segment register contains 7000H

it points to address 7000H. So each segment is 16 bytes. So here destination address is 79B2CH. So offset is. ($79B2C - 70000$) = 79B2CH.

That means. 79B2C + 00 = x240 V05

So, instructions can be -

~~Mov [AB2CH], DL~~

or

~~Mov DI, [AB2CH]~~

~~(instruction)~~

~~Mov [DI], DL~~

~~Mov DI, AB2CH~~

~~Mov [DI], DL~~

or

~~Mov SI, AB2CH~~

~~Mov [SI], DL~~

d. i. ~~Mov BX, [3502H]~~

There is no error in this instruction.

Set -

Movei. 2. ~~Mov DX,CX~~ temporary stub 244 21. 9

There is an error here. A byte is wanted to transfer in 16 bit register which is wrong. The correct instruction may be

~~Mov DX,CX~~

or

~~Mov DL, CL~~

~~ors 0.190v D.h, CL soft~~

$$0.67 \times 2 = 1.34$$

$$\begin{array}{r} 1.34 \\ \times 2 \\ \hline 1.68 \end{array}$$

$$\begin{array}{r} 1.68 \\ \times 2 \\ \hline 1.36 \end{array}$$

$$(8)_10 = (1000)_2$$

$$0.72$$

$$(1000.10101011)_2 \times 2 = 1.04$$

$$1.000101010110000111 \times 2 = 0.88$$

$$(127+3) = 130$$

$$\begin{array}{r} 130 \\ \times 2 \\ \hline 1.04 \end{array}$$

$$\begin{array}{r} 1.04 \\ \times 2 \\ \hline 0.08 \end{array}$$

$$\begin{array}{r} 0.08 \\ \times 2 \\ \hline 0.16 \end{array}$$

$$\begin{array}{r} 0.16 \\ \times 2 \\ \hline 0.32 \end{array}$$

$$\begin{array}{r} 0.32 \\ \times 2 \\ \hline 0.64 \end{array}$$

$$(130)_10 = (10000010)_2$$

As it's positive, so sign bit is 0

$$0 10000010 000101011000010 \times 2 = 1.12$$

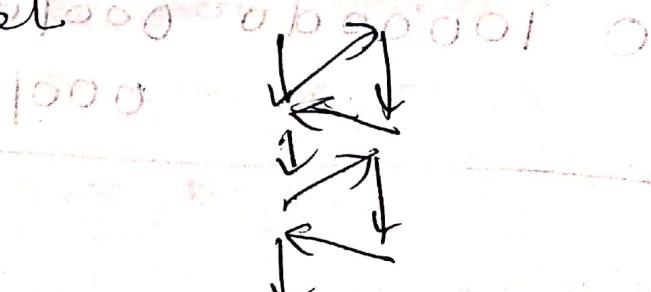
$$0001 \times 2 = 0.24$$

$$0.24 \times 2 = 0.48$$

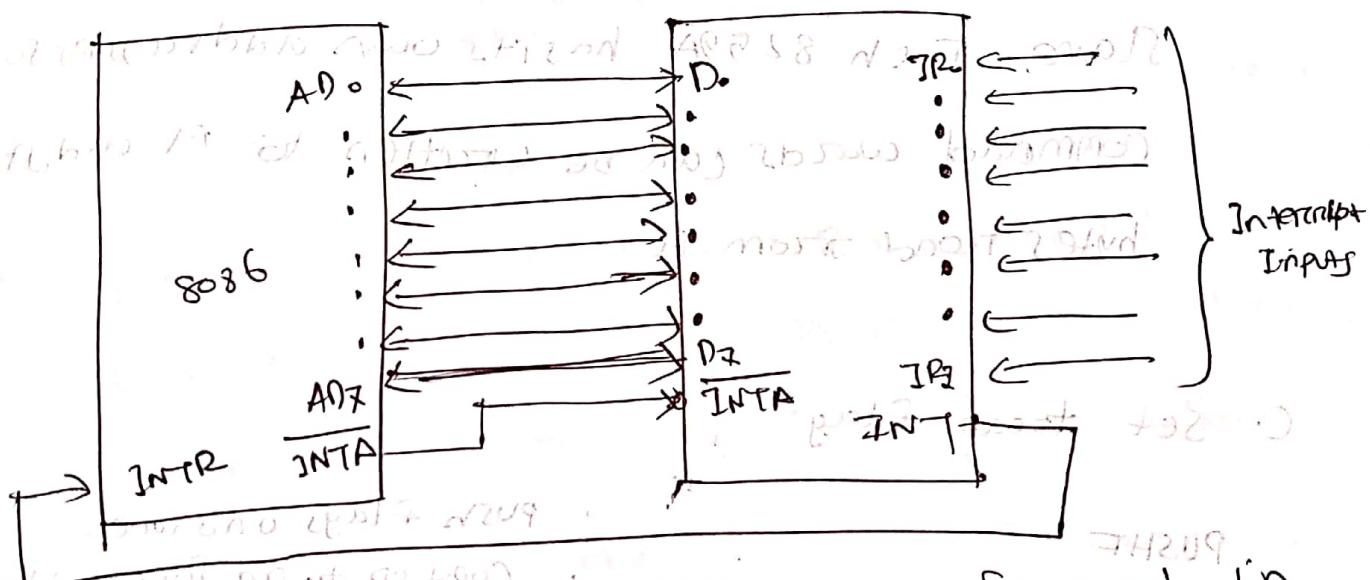
$$0.48 \times 2 = 0.96$$

$$0.96 \times 2 = 1.92$$

3. a. The INTA i/p of 8259 is connected to the INTA i/p of the 8086. The 8259 Aces the first INTA pulse from the 8086 to do some activities that depend on the mode in which it is programmed. When it receives the second INTA pulse from the 8086, the 8259A o/p in interrupt type on the 8 bit data bus. The interrupt type that it sends to the 8086 is determined by the IP of the received interrupt signal and by a number. It is sent to the 8259A when its initialized. Here each interrupt is acts as a funnel. Here each interrupt is executed like funnel and forms up to 8 different source into 8086 INTLOG and sends the 8086 a specified interrupt type for each of the 8 interrupt i/p. For this 8259 acts as a 8 funnel.



b. When more than one 8259s are connected to the CPU and run in cascade configuration. A cascaded up, it's called as a cascaded configuration. A cascaded configuration increases the number of interrupts handled by the system. As the max. numbers of 8259s interfaced can be 9 (1 master and 8 slaves) the maximum number of interrupts handled can be 64. The master 8259 has $\overline{SP}/\overline{EN} = 0V$ and the slave has $\overline{SP}/\overline{EN} = +5V$ and the slave has $\overline{SP}/\overline{EN} = 0V$. The 8259 can be easily interconnected to get multiple interrupts. Fig below shows how 8259 can be connected in the cascade mode.



In cascade mode one 8259 is configured in master mode and others are in slave mode. Each slave 8259 is identified by the number which is assigned as a part of its initialization.

Since the 8086 has only one INT input, only one of the 8259 INT pins is connected to the 8086 INT pin. The 8259 connected

without bus drivers. Then the 8086 INT pin is referred directly into the 8086 INT pin and while both 8259's are connected to the same bus, one is referred as the master. The INT pins from other 8259's are connected into the IR O/P of the master 8259. All output signals from the master 8259 are referred as slave. The cascaded 8259's are referred as slave. The INTA signal is connected to both master and slave 8259. The cascade pins CAS0 to CAS2 are connected from the master to the corresponding pins of the slave. For the master these pins function as I/P. The SP/EN signal is tied high for the master. However it is grounded for the slave. Each 8259A has its own address so that command words can be written to it and status bytes read from it.

C. Set trap flag:

PUSHF

; push flags on stack

MOV BP, SP

; Copy SP to BP for use as index

OR WORD PTR[BP+0] 100H ; SET TF flag.

; Restore flag Register

POP F

;

Reset trap flag

;

PUSH

;

AND WORD PTR[BP+0] 0FFFH

;

Trap flag is used for on chip debugging. When trap flag is set to 1 that puts the MP in single step mode for debugging. In single stepping the MP executes one instruction and enters into single step interrupt service routine (ISR). That is if the trap flag is set, the CPU automatically generates an interrupt after each instruction, allowing a program to be inspected as it executes instruction by instruction. And when trap flag is reset, no function is performed. For set OR operation is used and hence flag is set because of OR operation. For reset AND is used and hence flag is reset because of AND operation.

- d) The I/P is automatically disabled by the MP once its recognized and re-enabled by IPET or IPETO instruction. Diagram of handshake. Whereas disabling interrupts, the CPU will be unable to switch processes and processes can use shared variables without another process accessing them. It depends on interrupt so if can draft. Once a process has finished its interrupted, it can examine and update

the shared memory without fear that any other shared memory will interfere for this interrupt process will intervene.

The CPU automatically disabled as a part of response to an INTR interrupt.

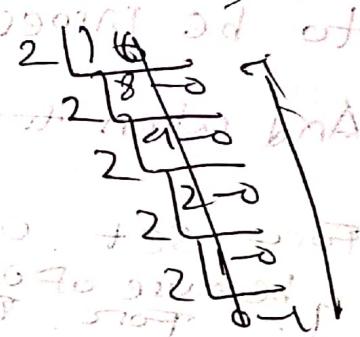
INT is automatically disabled as a part of response to an INTR interrupt.

INT is automatically disabled as a part of response to an INTR interrupt.

e. As it's a type A interrupt.

SO interrupt address = $16_{16} + 800000_{16}$

$16_{16} = (10)_2$



INT 0	21 8 400 42
INT 1	21 8 400 42
INT 2	21 8 400 42
INT 3	21 8 400 42
INT 4	21 8 400 42
INT 5	21 8 400 42
INT 6	21 8 400 42
INT 7	21 8 400 42
INT 8	21 8 400 42
INT 9	21 8 400 42
INT 10	21 8 400 42
INT 11	21 8 400 42
INT 12	21 8 400 42
INT 13	21 8 400 42
INT 14	21 8 400 42
INT 15	21 8 400 42
INT 16	21 8 400 42
INT 17	21 8 400 42
INT 18	21 8 400 42
INT 19	21 8 400 42
INT 20	21 8 400 42
INT 21	21 8 400 42
INT 22	21 8 400 42
INT 23	21 8 400 42
INT 24	21 8 400 42
INT 25	21 8 400 42
INT 26	21 8 400 42
INT 27	21 8 400 42
INT 28	21 8 400 42
INT 29	21 8 400 42
INT 30	21 8 400 42
INT 31	21 8 400 42

a.a. An instruction set architecture (ISA) is part of the abstract model of a computer that defines how the CPU is controlled by the software. The ISA acts as an interface between hardware and software, specifying both what the processor is capable of doing as well as how it gets done. The ISA provides the only way through which a user is able to interact with the hardware. It can be viewed as a programmer's manual because it's the portion of the machine that's visible to the assembly language programmers, the compiler writers and the application programmers. The ISA defines the supported data types, the registers, how the hardware manages main memory, key features such as

THE VM which instructions the MP can execute, and the I/P/O/P model of multiple ISA implementations. Then ISA can be extended by adding instructions or other capabilities or by adding support for larger addresses and data values, and so on to the bit length of the system. Now we have to divide our word to 32 bits.

b. Instruction Level Parallelism (ILP): It's the parallel or simultaneous execution of a sequence of instruction in a computer program. More specifically ILP refers to the way not of how instructions run per step of this parallel execution and not much different to how it probably to technique to execute instructions in parallel.

Dual pipelining is one of computer pipelining technique to execute instructions in parallel.

Dual pipe

Superscalar architecture is a method of parallel computing used in many processors. In superscalar computer, the fast CPU manager multiple instruction pipelines to execute several instructions concurrently during a clock cycle.

If dual pipelining is a parallel processing strategy in which an operation or a computation is partitioned into disjoint stages.

The stages must be executed in a particular order (could be a partial order) for the operation or computation to complete successfully. Each stage is implemented as a component which could be a hardware device or a software

thread. When a stage completes, it becomes available to do other work. Parallelism results from the execution of a sequence of operations or computation ~~concurrently~~ ^{in parallel} among several components so that at any given time several components ~~can~~ ^{can} be working and each one of the sequence are under execution and each one of these is at a different stage of the pipeline.

Superscalar architecture was one of such evolution. To exploit ILP superscalar processors fetch and execute multiple instructions in parallel thereby reducing the clock cycles per instruction (CPI).

ILP can be exploited either statically by the compiler or dynamically by the hardware. In this paper the # By sup. In superscalar most operations are on scalar quantities and improves the performance of the processor. Overall improvement of ILP allows the compiler and the processor to overlap the execution of multiple instructions even to change the order in which instructions are executed.

ILP is used in the form of Instruction Level Parallelism (ILP) and it is a way to utilize all the instructions that are available in a program. It is also known as a better solution.

Differences between RISC and CISC.

CISC	RISC
Emphasis on hardware	Emphasis on software
multiple instruction sizes	Instruction of same length
and formats	few formats

Less registers	Uses more registers
more addressing modes	fewer addressing modes
Extensive use of microprogramming	Complexity in compiler
Instructions take a varying amount of cycle time.	Instructions take one cycle time.

pipelining is difficult but pipelining is easy.

5.a. The feature word size determines the size of MP. Number of bits that can be processed by a processor in a single instruction is called its word size. Word size determines

the amount of RAM that can be accessed at one go and total number of pins on the microprocessor.

Example:

First commercial MP INTEL 4004 was a 4 bit processor it had 4 I/P pins and 16 O/P pins.