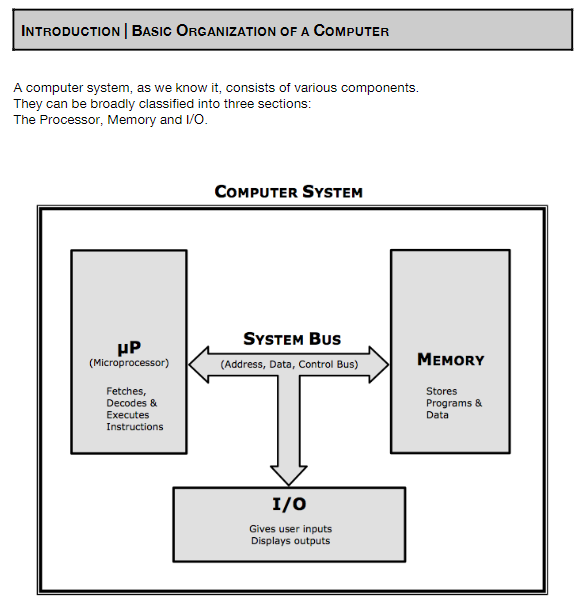
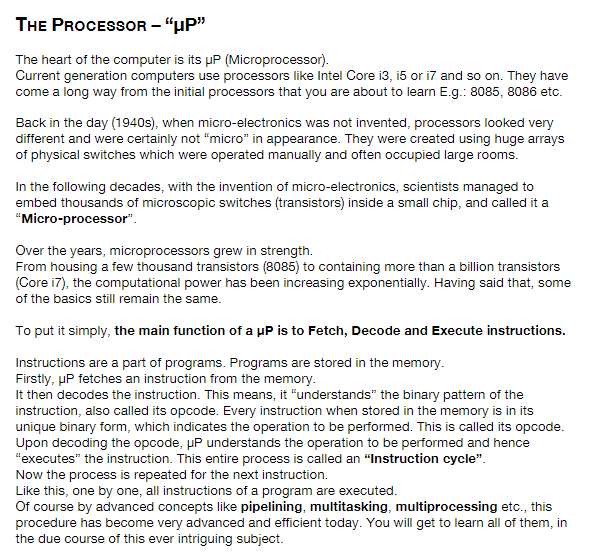
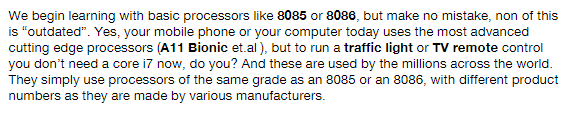
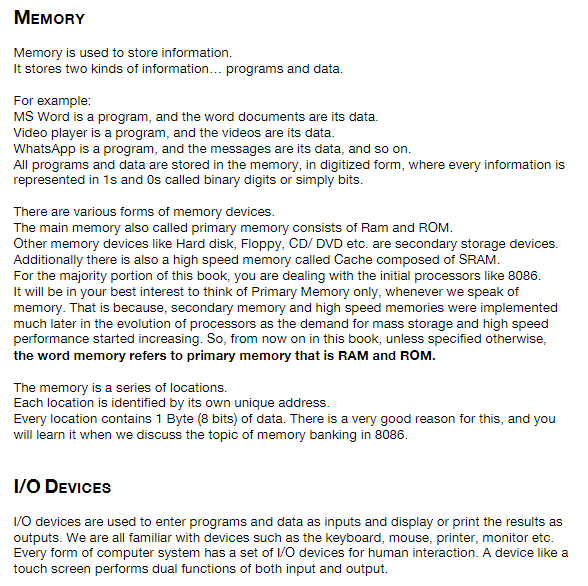
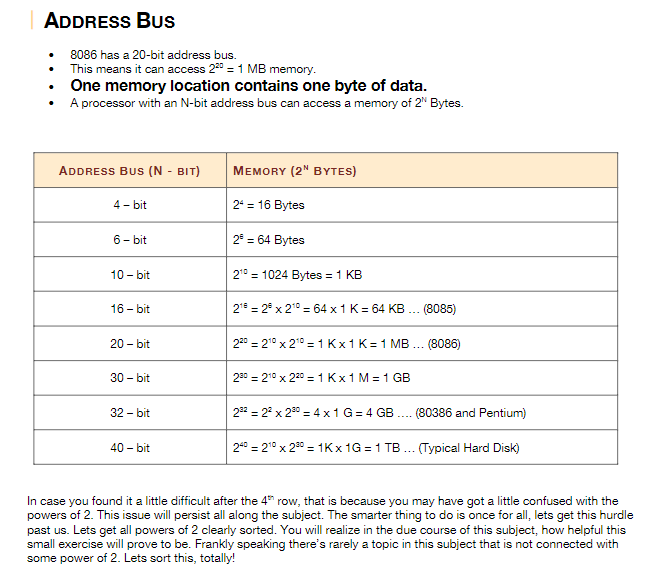
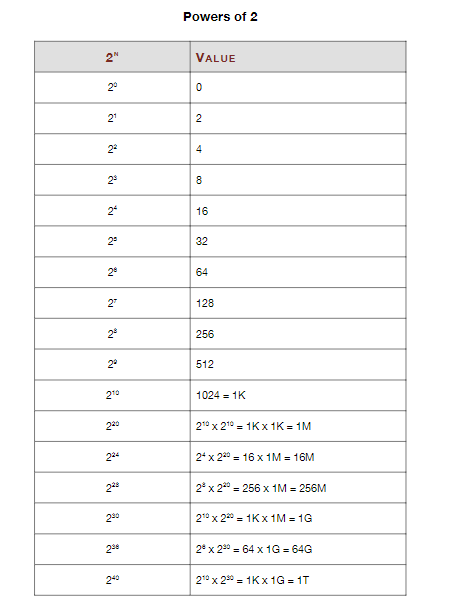
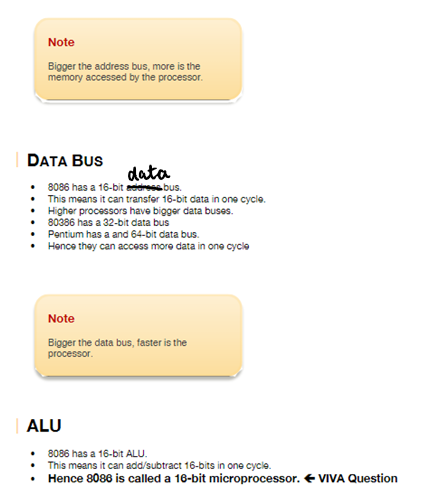
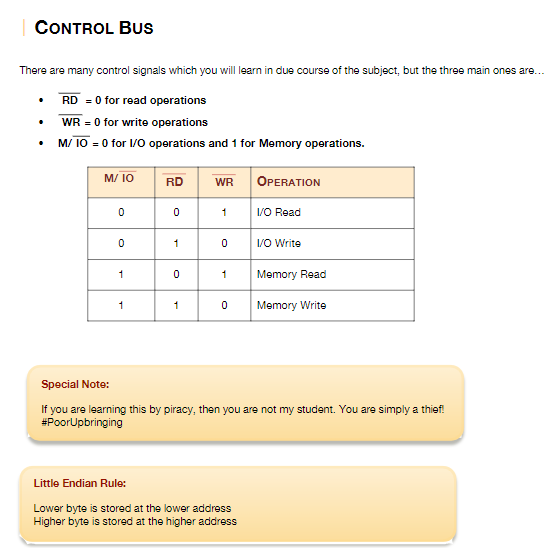
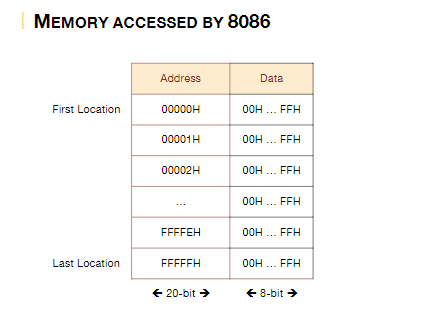
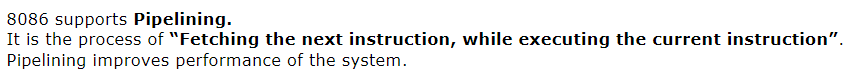
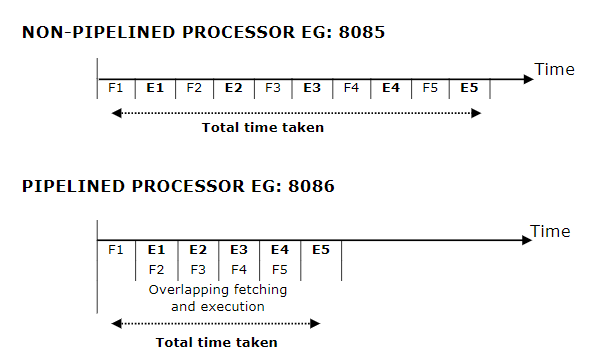
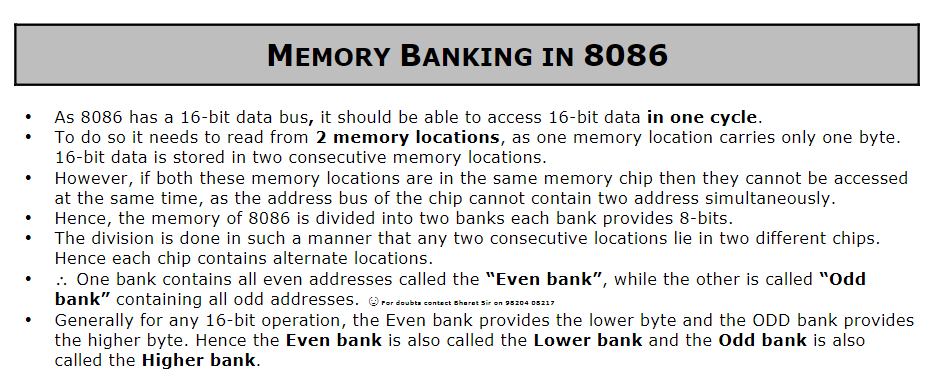
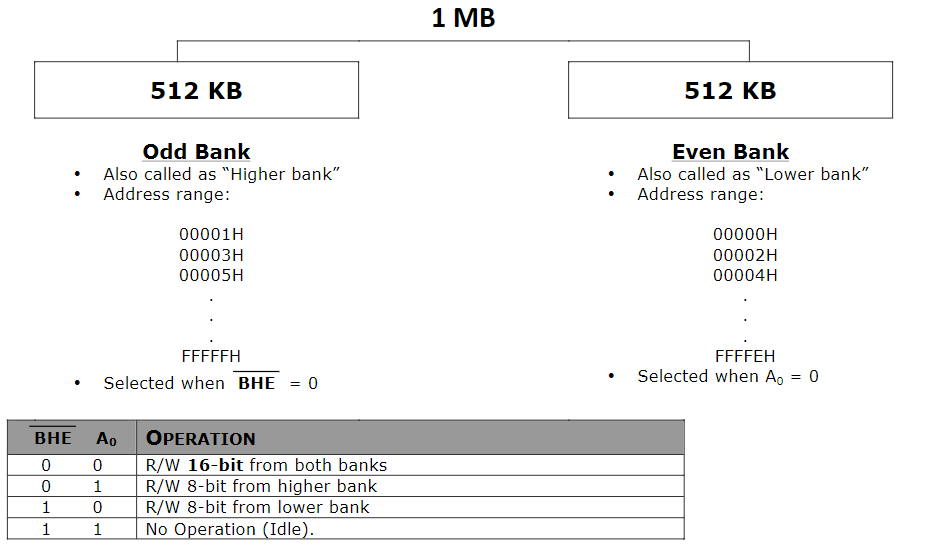
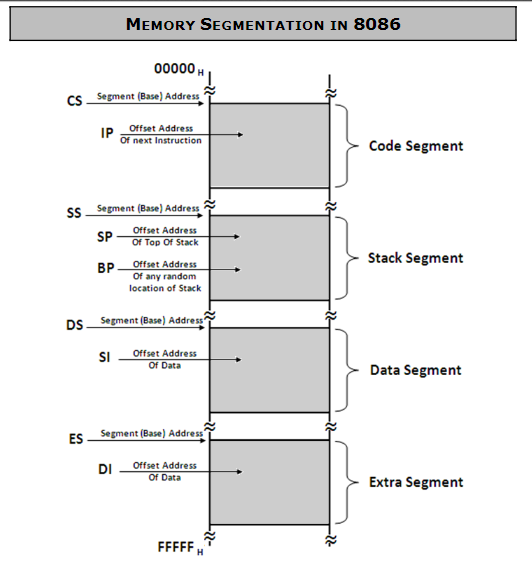
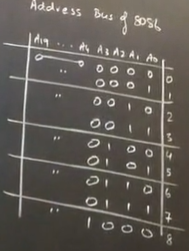
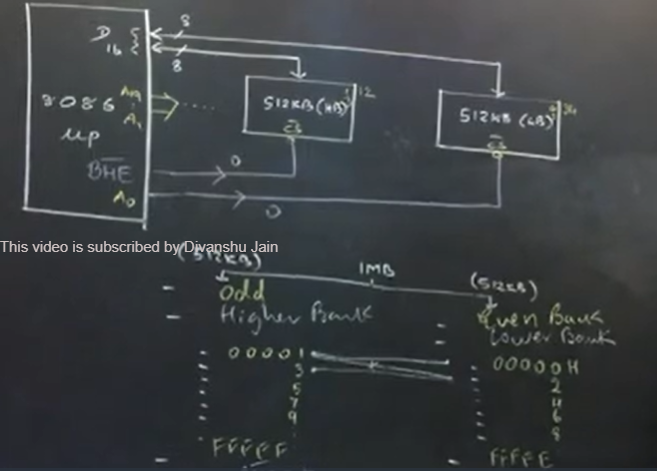
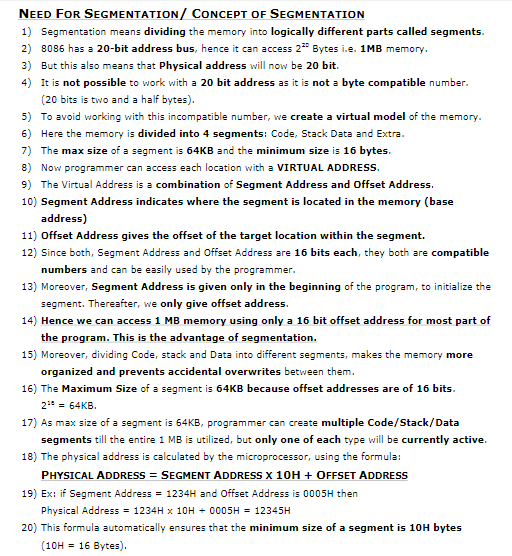
A blackboard with white text

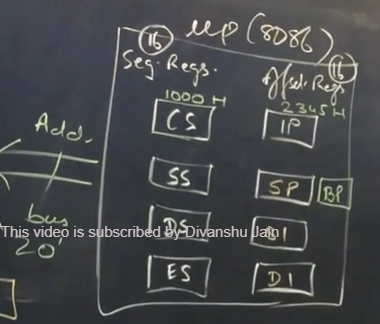
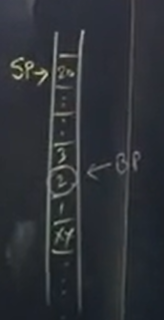
Description automatically generated

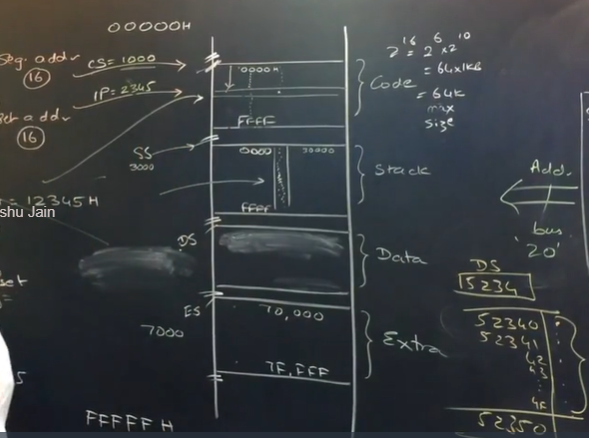


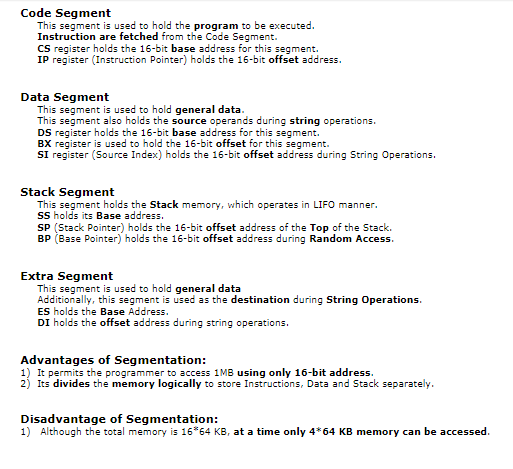
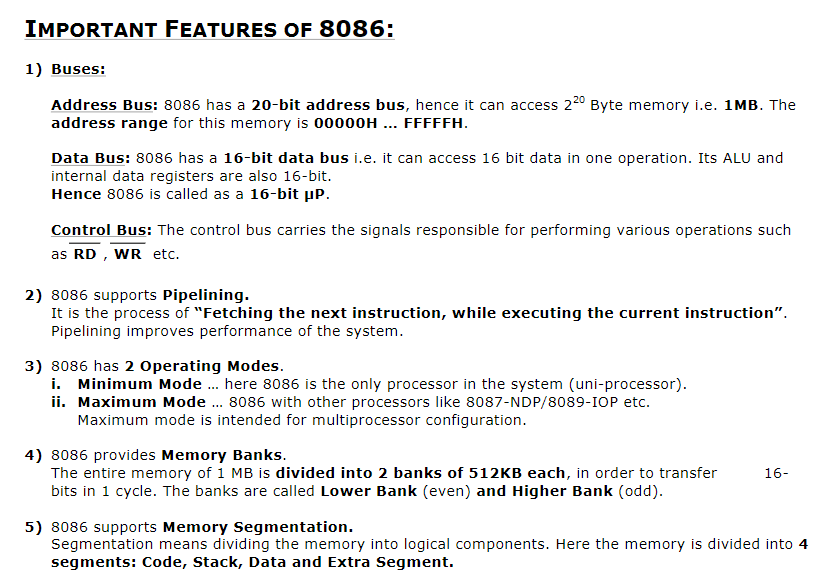
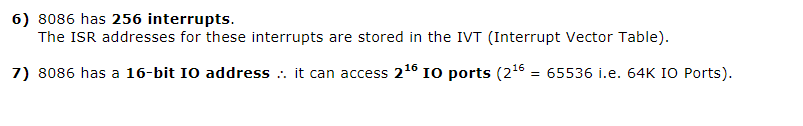
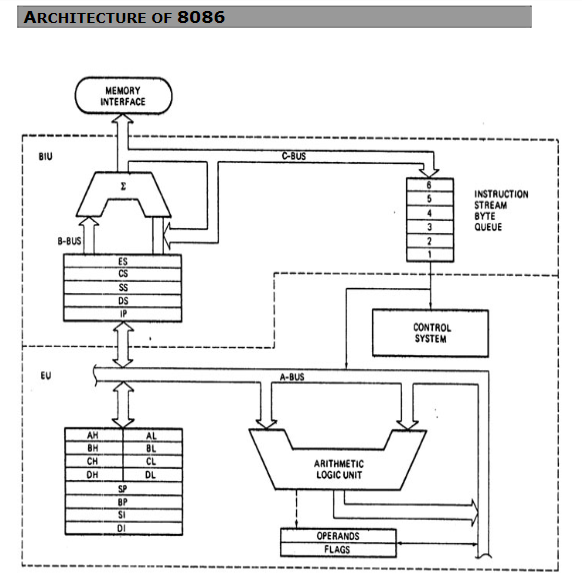
  

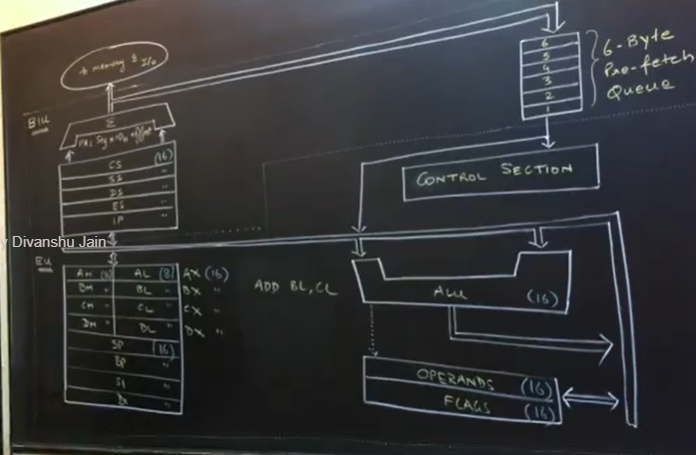
        

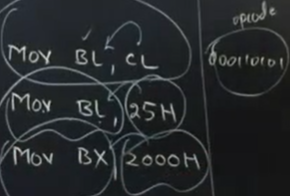
 







We write assemble program, assembler convert into machine language (series of 0 and 1) and put the program in memory and give to processor for execution.

Every instructiuon has its own opcode. MOV, ADD….

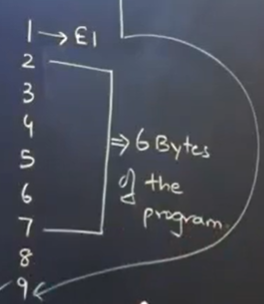
Opcode for Register operation , not for data

Data is operand

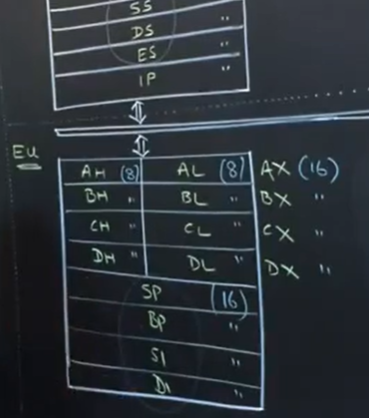
Q. If Queue 1 Byte is empty, will BIU fetch, it be filled or will wait to empy 2 byte? Data bus is 16 byte, wait for 2 byte

Pipilining saves time but

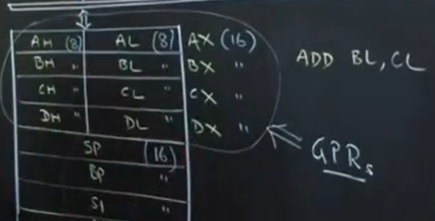
fails in case of jump/branch



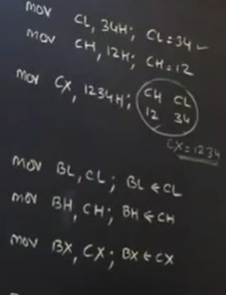
Control section will decode the machine code (binary/opcode) and will understand what to do and control signal will send this to all. Then execution will take place

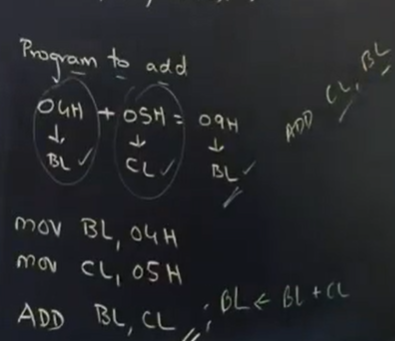


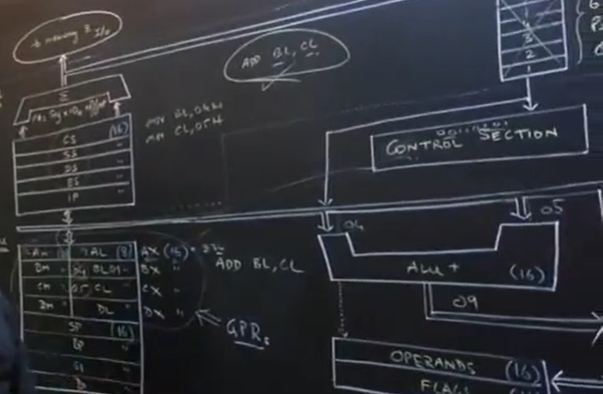
General Purpose registers A,B,C,D (16 bit registers, if AX) or can use independendet 8 bit registres AL,AH



 Ex 1234 H, 34 in AL, and 12 in AH

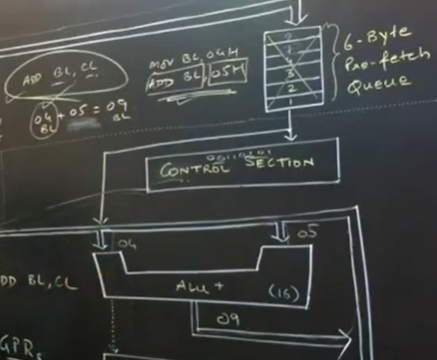






If you do , MOV BL, 04H

ADD BL, 05 H (both opcode and operand only BL)



Operand Register is temporary register , mainly used by processor itself not for programmer.

