Study of architecture and programming model of 8086

Design, Architecture

Studyof Architecture and Programming Model of 8086 Introduction: 8086Microprocessor is an enhanced version of 8085 Microprocessor. It is a 16-bitmicroprocessor which has 16 data lines and 20 address lines. 8086microprocessor can handle 2^20 memory locations, which is equivalent to 1MB. Itsupports two operating modes:(i)Maximum operating mode: This mode is suitable for system having more than oneprocessors.(ii)Minimum operating mode: This mode is suitable for system having only oneprocessor.

Features of 8086 microprocessor: Thefeatures of 8086 microprocessor are as follows: 1. It has an instruction queue, which can store 6 instruction bytes from thememory. 2. It uses two stages of pipelining: Fetch and Execute. 3. In fetch stage, 6 bytes of instructions is fetched and stored in the queue.

4. These instructions are then executed in the execute stage. 5. It has 16-bit ALU, 16-bit internal registers, internal data bus and 16-bitexternal data bus Architecture of 8086 microprocessor: 8086 microprocessor functional units: 8086microprocessor is divided into two functional units:(i)Execution Unit (EU)(ii)Bus Interface Unit (BIU) Execution Unit (EU): Theexecution gives instructions to BIU telling from where to fetch the data, decode and execute those instructions. Its function is to perform operations onthe data provided with the help of instruction decoder and ALU (Arithmetic andLogic Unit). It performs operations on the data through BIU. The parts of EUare as follows: ALU: Ithandles all the arithmetic and logical operations. Flag register: Itis a 16-bit register that changes its status according to the result stored inthe accumulator.

It consists of 9 flags and are divided into two groups:(i)Conditional Flags(ii)Control FlagsConditional Flags: Itrepresents the result of the last arithmetic or logical instruction executed. There are 6 conditional flags: 1. Carryflag: This fag is used when a carry is generated from the MSB (Most SignificantBit). This indicates an overflow condition for arithmetic operations. CF= 1 whena carry is generated and CF= 0 when carry is not generated. 2.

Auxiliaryflag: The processor uses this flag to perform binary to BCD conversion. Theflag is set when a carry/borrow from lower nibble to upper nibble is generateddue to an operation performed at ALU. 3. Parityflag: This flag is used to indicate the parity of the result. If there are evennumber of 1s, the flag is set and if odd number of 1s are present, the flag isreset. 4. Zeroflag: ZF= 1 when the result of any arithmetic or logical operation is zero, elseZF= 0.

5. Signflag: This flag holds the sign of the result. SF= 1 when the result is negative, else SF= 0. 6. Overflowflag: This flag represents the result when the system capacity is exceeded. Control Flags: Controlflag controls the operations of the execution unit.

There are 3 control flags: 1. Trapflag: It is used for stepwise execution of the program and allows the executionof only one instruction at a time for debugging. If TF= 1, then the program canbe run in single step. 2. Interruptflag: It is used to allow or prevent the interruption of a program. Forinterrupt enabled condition, IF= 1, else IF= 0.

3. Directionflag: It is used in string operation. It is set when string bytes are accessedfrom higher memory address to lower memory address and vice versa. General Purpose Register: Thereare 8 general purpose registers.

They are used in pairs to store 16-bit data. These registers are referred to the AX, BX, CX and DX. 1. AXregister: Known as accumulator register. It is used to store the operands forarithmetic operations. 2.

BXregister: It is used as base register. It is used to store the starting baseaddress of the memory area. 3. CX register: It is known as thecounter. It is used to store the loop counter. 4. DX register: This register holds the I/O portaddress for I/O instruction. Stack pointer register: Itholds the address from the start of the segment to the memory location.

Bus Interface Unit (BIU): BIUtakes care of all data and addresses transfers on the buses for the EU. EU hasno direct connection with the System Buses, so this is possible with the BIU. EU and BIU are connected with the Internal Bus. The parts of BIU are asfollows: 1. Instructionqueue: BIU contains the instruction queue. It gets upto 6 bytes of instructionsand stores them in the instruction queue. After executing the instructions, EUreads the next instruction from the instruction queue. Fetching the nextinstruction during the execution of current instruction is called pipelining.

2. SegmentRegister: There are 4 segment buses: CS, DS, SS and ES. It also has one pointerregister IP, which holds the address of the next instruction which is to beexecuted by the EU. i. CS (Code Segment): It is used foraddressing a memory location in the code segment of

the code segment of thememory. ii. DS (Data Segment): It consists of dataused by the program and is accessed in the data by segment by an offsetaddress.

iii. SS (Stack Segment): It handles memory tostore data and addresses during execution. iv. ES (Extra Segment): It is used to hold thedata which cannot be accommodated in the Data Segment.

3. InstructionPointer: It is used to hold the address of the next instruction to be executed.