CHAPTER 1:

1. How the processor uses the address bus, the data bus, and the control bus to communicate with the system memory?

* The address bus carries addressing signals from the processor to memory, I/O (or peripherals), and other addressable devices around the processor.
* A 64-bit processor has a 64-bit data bus and can communicate 64-bits of data at a time, and whether the data is read or written is determined by the control bus.

1. Which of the following are unidirectional and which are bidirectional?
2. Address Bus : The address bus is unidirectional and address always travels from processor to memory.
3. Data Bus: Data moves from both, processor to memory and memory to processor, so the data bus is bidirectional.
4. Control Bus: The control bus is a bidirectional bus and can carry information from processor to memory as well as from memory to processor.
5. What are registers and what are the specific features of the accumulator, index registers, program counter, and program status word?

* **AX is the primary accumulator**; it is used in input/output and most arithmetic instructions.
* **Index register** does not hold data but holds the address of data.
* **The program counter** holds the address of the next instruction to be executed.
* This is a special register in every architecture called **the flags register or the program status** word. The bits of the flags register work independently and individually, and combined its value is meaningless.

1. What is the size of the accumulator of a 64bit processor?

A 64bit processor has an accumulator of 64 bits.

1. What is the difference between an instruction mnemonic and its opcode?

Instruction mnemonics are easy to remember **short alphanumeric strings** .

Op codes are numbers that are understood by the computer and command it to perform various operations. They are usually inherently difficult for humans to remember.

ADD is a mnemonic.

1. How are instructions classified?

Instructions are divided into

1. Data Movement Instructions: These instructions are used to move data from one place to another
2. Arithmetic and Logic Instructions: Arithmetic instructions like addition, subtraction, multiplication, division, and Logical instructions
3. Program Control Instructions :These are instructions that control the program execution and flow by playing with the instruction pointer and altering its normal behavior to point to the next instruction
4. Special Instructions: They allow changing specific processor behaviors and are used to play with it. They are used rarely but are certainly used in any meaningful program.
5. A combination of 8bits is called a byte. What is the name for 4bits and for 16bits?

4bits is a nibble and 16bits is a Word.

1. What is the maximum memory 8088 can access?

The 8088 and 8086 used segmented addressing to allow these chips to address **1 MB** of memory using 20 address bits

1. List down the 14 registers of the 8088 architecture and briefly describe their uses.

General Registers (AX, BX, CX, and DX)

1. **AX is the primary accumulator**; it is used in input/output and most arithmetic instructions
2. **BX is known as the base register**, as it could be used in indexed addressing.
3. **CX is known as the count register**
4. **DX is known as the data register** or DX stands for Destination as it acts as the destination in input/output operations.

Pointer Registers:

1. **Instruction Pointer (IP)** − The 16-bit IP register stores the offset address of the next instruction to be executed. IP in association with the CS register (as CS:IP)
2. **Stack Pointer (SP)** − The 16-bit SP register provides the offset value within the program stack. SP in association with the SS register (SS:SP) refers to be current position of data or address within the program stack.
3. **Base Pointer (BP)-** It is also a memory pointer containing the address in a special area of memory called the stack

Index Registers:

1. **Source Index (SI)** − It is used as source index for string operations.
2. **Destination Index (DI)** − It is used as destination index for string operations.
3. **Flags Register** the 32-bit flags register combined are considered as the control registers

Segment Registers:

1. **Code Segment** − It contains all the instructions to be executed. A 16-bit Code Segment register or CS register stores the starting address of the code segment.
2. **Data Segment** − It contains data, constants and work areas. A 16-bit Data Segment register or DS register stores the starting address of the data segment.
3. **Stack Segment** − It contains data and return addresses of procedures or subroutines. The Stack Segment register or SS register stores the starting address of the stack.
4. **Extra Segment Register (ES):** also refers to a segment in the memory which is another data segment in the memory.
5. What flags are defined in the 8088 FLAGS register? Describe the function of the zero flag, the carry flag, the sign flag, and the overflow flag.
6. Sign Flag (S)
7. Zero Flag (Z)
8. Auxiliary Cary Flag (A)
9. Parity Flag (P)
10. Carry Flag (C)
11. Trap Flag(T)
12. Interrupt Flag(I)
13. Direction Flag(D)
14. Overflow Flag(O)
15. Zero Flag: The Zero flag is set if the last mathematical or logical instruction has produced a zero in its destination.
16. Carry Flag: When two 16bit numbers are added the answer can be 17 bits long or when two 8bit numbers are added the answer can be 9 bits long. This extra bit that won’t fit in the target register is placed in the carry flag where it can be used and tested
17. Sign Flag: The most significant bit (MSB) of a negative number in this representation is 1 and for a positive number it is zero. The sign bit of the last mathematical or logical operation’s destination is copied into the sign flag.
18. Overflow Flag: The overflow flag is set during signed arithmetic, e.g. addition or subtraction, when the sign of the destination changes unexpectedly.

1. 11. Give the value of the zero flag, the carry flag, the sign flag, and the overflow flag after each of the following instructions if AX is initialized with 0x1254 and BX is initialized with 0x0FFF.

add ax, 0xEDAB

add ax, bx

add bx, 0xF001

SF: 1, CF:0, OF:0,ZF:0

1. What is the difference between little endian and big-endian formats? Which format is used by the Intel 8088 microprocessor?

A big-endian system **stores the least significant byte at the largest memory address and most significant byte of a word at the smallest memory address****.**

Motorola uses big endian notation.

A little-endian system **stores the least significant byte at the lesser address and the most significant byte at the greater address.**

Intel uses little endian notation

1. For each of the following words identify the byte that is stored at lower memory address and the byte that is stored at higher memory address in a little endian computer.

|  |  |  |
| --- | --- | --- |
| Number | Higher memory address | Lower memory address |
| 1234 : 04D2h | 04 | D2 |
| ABFC | AB | FC |
| B100 | B1 | 00 |
| B800 | B8 | 00 |

1. What are the contents of memory locations 200, 201, 202, and 203 if the word 1234 is stored at offset 200 and the word 5678 is stored at offset 202?
2. 1234=(04D2)16

200 D2 201 04

1. 5678=(162E)16

202 2E 203 16

1. What is the offset at which the first executable instruction of a COM file must be placed?

The first instruction is placed at an offset address of 100

1. Why was segmentation originally introduced in 8088 architectures?

In linear memory model the whole memory appears like a single array of data. 8080 and 8085 could access a total memory of 64K using the 16 lines of their address bus. When designing iAPX88 the Intel designers wanted to remain compatible with 8080 and 8085 however 64K was too small to continue with, for their new processor. To get the best of both worlds they introduced the segmented memory model in 8088.