Chapter 1 (About the Manual/Notes?)

Daksh Pandey

February 2025

1 Introduction

First, the Intel has given a list of its processors for which the manuals' information hold true. I won't bother listing them again. Then comes the overview of the first volume. Again I wont bother listing the whole contents again and briefly describe them, just go over the manual yourself for that.

2 Notational Conventions

Some basic terminologies which will only make reading further notes and manual easier.

2.1 Bit and Byte Order

- In the figure below, the lowest address is at the bottom right corner.
- Highest address in at the upper left corner.
- Numerical value of a set bit is equal to two raised to the power of the bit position. Bit positions are numbered from right to left.
- Example: given the number 82, which in binary is 01010010, so the right-most 0 is at the 0th position and left-most 0 is at 7th position.

2.2 Reserved Bits and Software Compatibility

- Just don't depend on these bits.
- These are essential for compatibility with the future processors.
- These are not reliable, so just avoid using them.

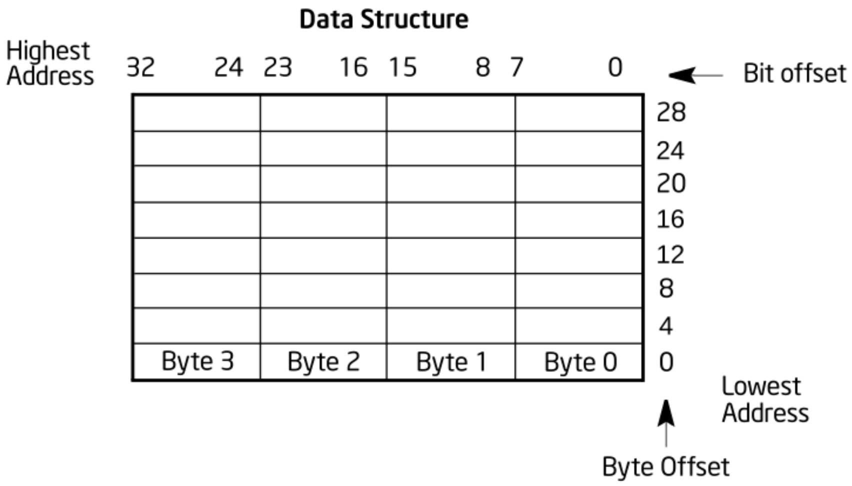

Figure 1: Bit and Byte Order

2.3 Instruction Operand

- When we use symbolic representation for instructions, some elements from IA-32 (Intel Architecture 32) AL is used.
- According to IA-32's AL, instruction has the following format:

label: mnemonic arg1, arg2, arg3

- **Label:** identifier followed by colon.
- **Mnemonic:** reserved name for a class of instruction opcodes.
- Arguments are the operands. There may be 0 to 3 arguments, depending on the opcode. When presented, they may take form of literals or identifiers for data items. Operand identifiers are either reserved names of registers or assumed to be assigned to data items declared somewhere else.
- When two operands in arithmetic or logical instruction, right is the source and left is the destination.
- Example (from the manual only):

LOADREG: MOV EAX, SUBTOTAL

- LOADREG is the label, MOV is mnemonic id of an opcode, EAX is the destination, SUBTOTAL is the source operand.

- If you are wondering what Opcode is, it is a single instruction which can be executed by the CPU, MOV is an opcode (obviously in its mnemonic form).

2.4 Binary and Hexadecimal

Cmon, you know these. Binary can sometimes be followed by B btw.

2.5 Segmented Addressing

- Processor used byte addressing.
- Other words, memory is organized and accessed as a sequence of bytes.
- A byte address is always used, whether we are accessing only a byte or bytes.
- For future reference: word = 2 bytes, doubleword = 4 bytes, quadword = 8 bytes.
- Range of memory that can be accessed: **address space**.
- Independent address spaces are called: **segments**.
- For example:

  DS:FF79H

- In the above example, DS it the segment and FF79H is the address whithin the segment.
- Example below represents an instruction in the code segment. CS register points to the code segment and EIP register contains the address of instructions:

  CS:EIP

2.6 Execption

- Event which occurs when an instruction causes an error. Example, attempt to divide by 0.
- Some types of exceptions provide an error code. Sometimes we may not get an accurate reported code.

All the image credits go to the reference at the end

References

[1] Intel® 64 and IA-32 Architectures Software Developer's Manual Volume-1