GIFT UNIVERSITY
Realize Your Career Dreams

GIFT School of Engineering and Applied Sciences

Spring 2022

CS-345: Computer organization and assembly language

Lab-6 Manual

Memory System

MEMORY SYSTEM

OBJECTIVES

To learn the x86 memory system segmentation

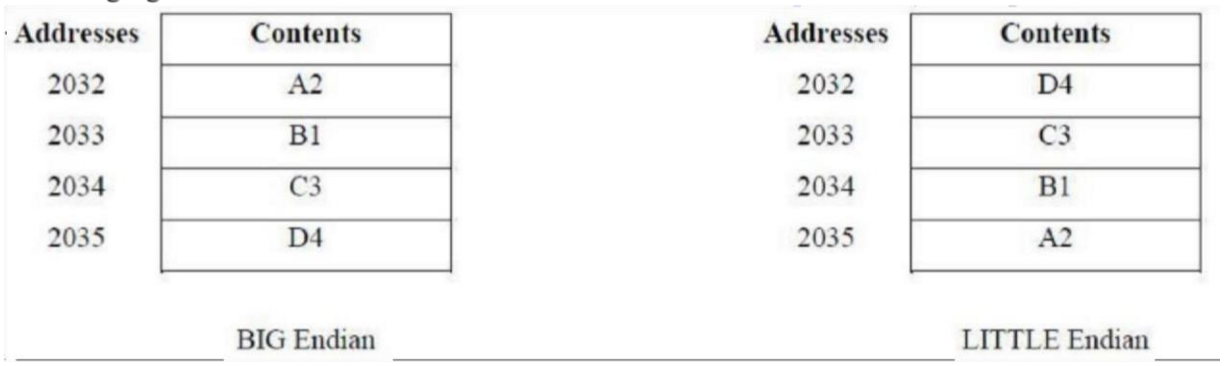
THEORY

Depending on the machine, a processor can access one or more bytes from memory at a time. The number of bytes accessed simultaneously from main memory is called word length of machine. Generally, all machines are byte-addressable i.e.; every byte stored in memory has a unique address. However, word length of a machine is typically some integral multiple of a byte. Therefore, the address of a word must be the address of one of its constituting bytes. In this regard, one of the following methods of addressing (also known as byte ordering) may be used.

Big Endian – the higher byte is stored at lower memory address (i.e. Big Byte first). MIPS, Apple, Sun SPARC are some of the machines in this class.

Little Endian - the lower byte is stored at lower memory address (i.e. Little Byte first). Intel's machines use little endian.

Consider for example, storing 0xA2B1C3D4 in main memory. The two byte orderings are illustrated in
following figure



Addresses	Contents
2032	A2
2033	B1
2034	C3
2035	D4

BIG Endian

Addresses	Contents
2032	D4
2033	C3
2034	B1
2035	A2

LITTLE Endian

Memory Models

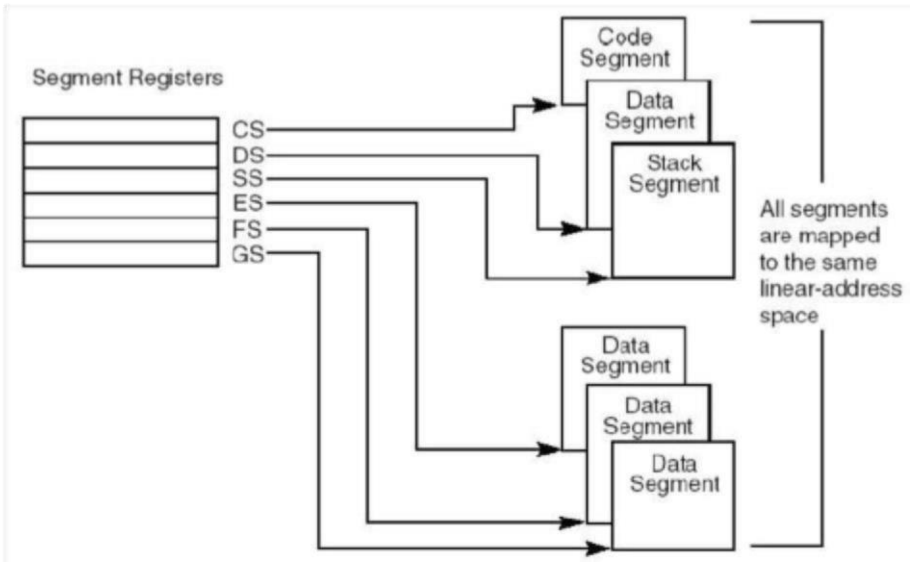
In earlier processors like 8080 and 8085 the linear memory model was used to access memory.

Flat/ Linear Memory Model – memory appears to a program as a single, contiguous address space of 4GB. Code, data, and stack are all contained in this address space, also called the linear address space
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. The three logical parts of a program (data, code and stack) should appear as three distinct units in memory, but making this division is not possible in the linear memory model. The segmented memory model does allow this distinction.

Segmented Memory Model – memory appears to a program as a group of independent memory segments, where code, data, and program stack are contained in separate memory segments. To address memory in this model, the processor must use segment registers and an offset to derive the linear address. The primary reason for having segmented memory is to increase the system's reliability by means of protecting one segment from other.
The processor sees code from the code window and data from the data window. The size of one window is restricted to 64K. 8085 software fits in just one such window. It sees code, data, and stack from this one window, so downward compatibility is attained. However the maximum memory iAPX88 can access is 1MB which can be accessed with 20 bits. Compare this with the 64K of 8085 that were accessed using 16 bits. The idea is that the 64K window just discussed can be moved anywhere in the whole 1MB. The four segment registers discussed in the Intel register architecture are used for this purpose. Therefore four windows can exist at one time. For example one window that is pointed to by the CS register contains the currently executing code.

Segment Registers

The segment registers hold the segment selectors which are special pointers that point to start of individual segments in memory. The use of segment registers is dependent on the memory management model in use. When using the segmented memory model, each segment is loaded with a different memory address as shown in the following figure



The segment registers (CS, DS, SS, ES, FS, and GS) hold 16-bit segment selectors. To access a particular segment in memory, the segment selector for that segment must be present in the appropriate segment register. Each of the segment registers is associated with one of three types of storage: code, data, or stack. The DS, ES, FS, and GS registers point to four data segments. The availability of four data segments permits efficient and secure access to different types of data structures.

The CS register contains the segment selector for the code segment, where the instructions being executed are stored. The processor fetches instructions from the code segment, using a logical address that consists of the segment selector in the CS register and the contents of the IP (Instruction Pointer) register. The CS register opens a 64K window in the 1MB memory and then IP works to select code from this window as offsets. IP is 16 bit wide and works only inside this window and cannot go outside of this 64K in any case. If the window is moved i.e. the CS register is changed, IP will change its behavior accordingly and start selecting from the new window. The IP register always works relatively, relative to the segment base stored in the CS register.

PROBLEMS

1. Fill in the following tables to show storage of 0xABDADDBA at address 0119 in the memory of a machine using:

(i) Little endian (ii) Big endian byte ordering.

Address	Content
0119	
011A	
011B	
011C	

Address	Content
0119	
011A	
011B	
011C	

2. Write instructions to do the following.

   a. Copy contents of memory location with offset 0025 in AX.

   b. Copy AX into memory location with offset 0FFF.

   c. Move contents of memory location with offset 0010 to memory location with offset 002F.

3. Write a program in assembly language that calculates the square of six by adding six to the accumulator six times.

4. Rewrite program p7.asm (from class source code) without using bx.
   a. Try cx.
   b. Use si or di

5. Write a program to write 10AB and 23CD to memory using separate label for each number. Then copy each byte of your number to al and add it to dl one by one.

6. Write a program to write 10AB and 23CD to memory using separate label for each number. Then copy each byte of your number to al and add it to dl one by one.