Some Background on the Processor family

- IBM 8086, 8088, 80x86
- Memory Map of the XT series
- 20-bit addresses, how many addresses?
- 00000 $\rightarrow FFFFF$
- 1Mbytes of memory divided into 64K blocks
- How many such blocks are there?
  - All blocks above 9FFFF used by system
    How many such blocks are there?
  - Last block is System ROM (ROM BIOS)
    ROM = **R**ead **O**nly **M**emory (permanent)
    RAM = **R**andom **A**ccess **M**emory (volatile, non-permanent)
    BIOS = Basic Input Output Routines
  - Lowest 400_h bytes of memory: interrupt vector table
    Each INT vector address requires 4 bytes, how many handlers?

- Next 200_h bytes for BIOS data buffers
- Next, MS-DOS uses up memory (depends on version of MS-DOS)

  One item to note is that command.com, the user-interface and housekeeper, is split into two parts

  1. Resident portion (small)
  2. Transient portion (larger)

  The transient portion is not necessary when a user program is running and can be overwritten. It may have to be re-loaded from disk when the user program ends

Segmentation

- Memory address access/manipulation is a fundamental processor operation.
- It must be speeded up as much as possible
- Registers have very, very fast response times

$\rightsquigarrow$ Store Memory addresses in registers as much as possible

- Problem:
  1. Registers are 16 bits or 2 bytes
  2. Memory addresses are 20 bits wide
- Solution:
  - Could use 2 registers, but that would give us 2^{32} addresses; far more than are possible (2^{20})
  - Intel's address calculation is a little bizarre but very easy to calculate (for us and for the computer)

Address Calculation

- Intel does break up the address into two pieces (stored in two registers)
  1. Segment
  2. Offset
- Rule

  To calculate an address given its segment and offset, multiply the segment by 16_{10} and add the offset.

  This is the Intel Segmentation Scheme
- Easy to calculate:
  1. In binary, simply append 4 zeros to segment and add offset
  2. In Hex, Even more simply, append a single zero and add offset
- In Intels's 8086/80286 Scheme, two 16 bit numbers team up to form a 20 bit address

Four Segments that matter

- Segments are ranges of memory that spread across the 1Mbyte address space. Maximum length of a segment is 64Kbytes

  1. Segment A = 40000 to 4FFFF
  2. Segment B = 7A000 to 82741
  3. Segment C = 1C400 to 28880
  4. Segment D = 17000 to 1FF00
  5. Segment E = 1C400 to 28880

  Paragraphs: Segment Addresses must be divisible by 16: Last digit of a segment's 16bit hex address must be 0. That is, Segments must start at paragraph boundaries.