Ch 2 : Microprocessor and its Architecture (2.1-2.3 & 2.5)

2.1 Internal Microprocessor Architecture

- **8086-Core 2 Microprocessor**
- **Programming Model:**
    - *Program visible*
        - Registers used during application programming and specified by instructions
    - *Other registers are program invisible*
        - Not addressable directly during applications programming but may be used indirectly during system programming.
        - 80286 and above
        - Control and operate protected memory
    - *64-bit extension (R appended)*
        - 80386-Core2 has 32-bit internal architecture
        - 80286, 8086, 8088 are 16-bit but upward-compatible to 80386-Core2
        - 8-bit registers are:
            - AH, AL, BH, BL, CH, CL, DH. DL
        - 16-bit registers are:
            - AX, BX, CX, DX, SP, BP, DI, SI, IP, FLAGS, CS, DS, ES, SS, FS, GS
            - The ones ending with X are made of their corresponding H and L 8-bit registers.
        - 32-bit extended registers are same as 16-bit registers with an appended E. E.g. EAX, EBX but not for FS and GS
            - These are only available in 80386 and above
        - Multipurpose registers are:
            - EAX, EBX, ECX, EDX, EBP, EDI, ESI
        - Registers R8 through R15 are also 64-bit addressed as a byte, word, double-word or quad-word but only the rightmost 8-bits are the byte.
            - These don't have provision to access bits 8-15
            - AH, BH, ... (legacy high bit registers) can't be accessed in the same instruction with R8-R15 because legacy softwares don't access R8-R15
            - RNB access the byte in the register where N is the number from 8-15.
            - RNW accesses the word.
            - RND accesses the double-word.
            - RN for the whole 64-bit
- **Multipurpose Registers:**
    - R#X = E#X, #X (#H & #L)
    - General Purpose
    - *RAX (Accumulator) :*
        - Used for multiplication, division, or adjustment.
        - In 80386 and above, EAX may hold the offset address of a location in the memory
        - In Pentium4 and Core2, RAX holds the offset address allowing IT(terra) byte of the memory to be accessed through a 40-bit address bus.
    - *RBX (Base Index) :* Holds the offset address of a location in the memory for all versions.
    - *RCX (Count):*
        - Holds count for various instructions like repeated string(CX), shift and rotate (CL), loop instructions(CX or ECX or RCX).
        - In 80386 and above, ECX may hold the offset address of a location in the memory
        - In Pentium4, RCX holds the offset address.
    - *RDX (Data):*
        - Holds a part of the result from a multiplication or part of the dividend before a division.
        - In 80386 and above, it may also address memory data
    - *RBP (Base Pointer):* Points to a memory location in all versions for memory data transfers.
    - *RDI (Destination Index) :* Addresses string destination data for the string instructions.
    - *RSI (Source index):* Addresses source string data for the string instructions.
    - R8 - R15 : Only found in Pentium4 and Core2 if 64-bit extensions are enabled. Only used if 64-bit processors are common.
- **Special-Purpose registers:**
    - *RIP (Instruction Pointer)*:
        - Addresses the next instruction in a section of memory defined as a code segment.
        - IP in real mode and EIP in protected mode.
        - Only in 80286 and above.

    - Points to the next instruction in a program.
    - Can be modified with a jump or a call instruction.
    - In 64-bit mode, it contains a 40-bit address at present to address a IT flat address space.
  - *RSP (Stack Pointer)*:
    - Addresses an area of memory called the stack.
    - Stack memory stores data through this pointer.
  - *RFLAGS:*
    - Indicate the condition of the microprocessor and control its operation.
    - The rightmost five flag bits and the overflow flag change after many arithmetic and logic instructions execute.
    - The flags never change for any data transfer or program control operation.
    - also used to control features found in the microprocessor.
- **RFLAGS:**
  - *C (carry)* :
    - Holds the carry after addition or borrow after subtraction
    - Also indicates error conditions especially for DOS function calls.
  - *P (parity)*:
    - Logic 0 for odd parity and logic 1 for even parity.
    - Parity is the count of ones in a number expressed as even or odd.
    - Was implemented in early Intel microprocessors for checking data in data communication environment but now data communications equipment is used.
  - *A (auxiliary carry)*:
    - holds the half carry after addition or subtraction between bit positions 3 and 4 of the result.
    - Tested by the DAA and DAS instructions to adjust the value of AL after a BCD addition or subtraction.
  - *Z (zero)*: Shows if result of an operation is 0.
  - *S (sign)* : holds the arithmetic sign of the result. 1 for negative.
  - *T (trap)*:
    - enables trapping through an on-chip debugging feature.
    - Flow interrupted if 1.
    - when instructions run line by line like in debug mode
  - I (interrupt):
    - Controls the operation of the INTR (interrupt request) input pin.
    - Its state is controlled by the STI (set I flag)and CLI (clear I flag)
  - D (direction) :
    - Selects either the increment or the decrement mode for the DI or SI during string instructions.
    - 0 for increment.
    - Set with STD and cleared with CLD instruction.
  - O(Overflow):
    - Occurs when signed numbers are added or subtracted.
    - indicates exceeding capacity.
  - IOPL (I/O privilege level):
    - Used in protected mode to select the privilege level for I/O devices.
    - I/O only executes if current level is higher. Else interrupt occurs.
    - 00 is highest and 11 is lowest.
  - NT (nested task):
    - indicates that the current task is nested within another task in protected mode.
    - Set when the task is nested
  - RF (resume): Used with debugging to control the resumption of execution after the next instruction.
  - VM(virtual mode):
    - Selects virtual mode operation in protected mode system.
    - Allows multiple DOS memory partitions that are 1M byte in length to coexist in memory thus multiple DOS programs can execute.
    - Used to simulate DOS in modern windows.
  - AC (alignment check) :
    - activates if a word or a doubleword is addressed on a non-word or a non-doubleword boundary.
    - Only in 80486SX microprocessor so that its coprocessor 80487SX can use it for synchronization.
  - VF (Virtual Interrupt) : Copy of Interrupt flag bit available to Pentium-Pentium4
  - VIP (Virtual Interrupt Pending) :
    - Provides information about VM interrupt to Pentium-Pentium4
    - Used in multitasking environments for the OS

- ID (identification) :
  - Indicates that Pentium-Pentium4 supports CPUID
  - The CPUID instruction provides the system with information about the Pentium microprocessor, such as its version number and manufacturer.
- **Segment Registers:**
  - Generate memory addresses when combined with others.
  - Either 4 or 6 in one microprocessor.
  - Different functioning in real mode and protected mode.
  - Little use in the 64-bit flat model.
  - CS (code):
    - Holds the code i.e. defines the starting address of the section of memory holding code.
    - In real mode operation, the start of a 64K-byte section of memory is defined;
    - In protected mode, a descriptor, that describes the starting address and the length of the section of memory holding code, is selected.
    - Limited to 64K bytes in 8088-80286 and 4G bytes in 80386 and above (in protected mode)
    - Used in the flat model in 64-bit mode
  - DS(Data):
    - Section of memory containing most data used by the program
    - Accessed by offset address
    - Limited to 64K bytes in 8088-80286 and 4G bytes in 80386 and above
  - ES (Extra): Additional DS used by some string instructions to hold destination data.
  - SS (Stack):
    - Defines area of memory used for stack
    - The stack entry point is determined by the stack segment and stack pointer registers.
    - The BP register also addresses data within the stack segment.
  - FS and GS :
    - Supplemental registers in 80385-Core2
    - Allows 2 additional memory segments for access by programs
    - Windows uses them for internal operations (unavailable)

2.2 Real Mode Memory Addressing

- 80286 and above - real or protected mode.
- 8086 and 8088 - real mode
- No real mode in 64-bit mode of Pentium4 and Core2
- Real Mode addresses only first 1M byte of memory, thus called Real/ Conventional/ DOS Memory.
- DOS requires real mode not Windows though
- Makes 8086/88 programs upward compatible
- Operation always begins in real mode unless operated in 64-bit.
- **Segments and Offsets:**
  - Memory location = segment + offset address
  - Segment Address(SA): In segment registers, defines the beginning of any 64K byte memory segment
  - Offset Address/ Displacement(OA): Selects any location within the 64K byte segment
  - SA is appended with a 0H before using, forming a 20bit memory address to access the start of the segment.
  - The microprocessor must generate a 20-bit memory address to access a location within the first 1M of memory.
  - Real mode segments can begin only at a 16-byte boundary in the memory system, often called a paragraph.
  - Add FFFFH to the SA to get the ending address of each segment since each segment is of 64K byte.
  - Often written as SA:OA e.g. 1000H:2000H where actual location comes out to be 12000H.
  - In 80286 (with special external circuitry) and 80386 through the Pentium 4, an extra 64K minus 16 bytes of memory is addressable when the SA is FFFFH and the HIHEM.SYS driver for DOS is installed in the system. This memory (0FFFF0H–10FFEFH) is referred to as high memory.
    - In this case, an A20 pin is enabled when an address is accessed using FFFFH SA.
    - E.g. FFFFH:4000H = 103FF0H
    - If A20 is not supported, the above address will come out to be 03FF0H
  - Some addressing modes combine more than one register and an offset value to form the OA.
    - E.g. OA = F000H + 3000H = 12000H. However, it is modulo 16 sum thus the 1 is dropped and OA is 2000H
- **Default Segment and Offset Registers:**
  - For CS, it is IP or EIP i.e. CS:IP or CS:EIP
  - For stack, SS:SP/ESP or SS:BP/EBP

    - Only rightmost 16 bits used in extended registers
    - If a number bigger than FFFFH is placed into an Offset Register in a microprocessor operating in real mode, system halts and indicates an addressing error
    - 8086-80286 allows 4 memory segments while 80386-Core2 allows 6.
    - Memory segments can touch or even overlap if 64K bytes of memory are not required for a segment.
    - programs can only access 4/6 segments even while having more
    - When a program is placed in memory by DOS, it is loaded at the TPA at the first available area of memory above drivers and TPA programs, indicated by a free-pointer maintained by DOS.
        - The transient program area (TPA) holds the DOS operating system and other programs that control the computer
    - Program loading is handled automatically by the program loader located within DOS.
        - It also calculates and assigns segment starting addresses.
- **Segment and Offset Addressing Scheme Allows Relocation:**
    - The scheme is complicated but advantageous to the system.
    - It allows relocation of DOS programs in memory.
    - Real Mode programs can function in protected mode.
    - Allows programs and data to be relocated without changing their contents:
        - Relocatable Program: One that can function without changing the area of memory its placed in.
        - Relocatable Data: One that can be used without changing the program.
    - Memory segments can be moved without changing the OA because OA is w.r.t. the segment.
    - All programs in windows are written assuming that the first 2G of memory are available for code and data.
        - When the program is loaded, it is placed in the actual memory, which may be anywhere and a portion may be located on the disk in the form of a swap file.

2.3 Introduction to Protected Mode Memory Addressing

- Access to the 1M memory and above
- Windows operates in protected mode
- Addressing the extended section requires a different scheme.
- OA exists but not SA. Instead selectors and descriptors
    - The segment register contains a selector that selects a descriptor from a descriptor table.
    - The descriptor describes the memory segment's location, length, and access rights.
- Because the segment register and OA still access memory, protected mode instructions are identical to real mode instructions.
- most programs written to function in the real mode will function without change in the protected mode
- Other than the different segment register interpretation, the OA can be a 32-bit number instead of a 16-bit number in the protected mode.
    - This allows segment to be 4G bytes in length.
- No paragraph boundary
- **Selectors and Descriptors**
    - Descriptor:
        - 8 bytes in length in 80286
        - describes the location, length, and access rights of the segment
        - Two tables: global and other local descriptors of 64K bytes
            - Global/System Descriptors(gd): contain segment definitions applying to all programs
            - Local/Application Descriptors(ld): contain segment definitions unique to an application
        - 8192 X 2 tables = 16,384 memory segments available for each application X 4G bytes = 64T bytes memory available
        - 80286 - upward compatible
        - base address portion indicates start location of segment
            - 24 bit in case of 80286 to point within 16M
            - 32 bit in case of 80386 and above
        - Segment limit contains the last offset address found in a segment.
            - for segment beginning at F00000H and ending at F000FFH, limit is FFH.
            - 16 bits for 80286
            - 20 bit for 80386 through Pentium4
        - 80286 access memory segments that are between 1 and 64K bytes in length.
        - 80386 and above access memory segments that are between 1 and 1M byte, or 4K and 4G bytes in length.
        - Special feature in 80386 through Pentium 4, G bit / Granularity Bit:
            - If G=0, limit specifies a segment of 00000H to FFFFFH.
            - If G=1, value of limit is 00000FFFH to FFFFFFFFH i.e. Append FFFH to limit
            - Allows segment length of 4K to 4G bytes in steps of 4K bytes.
        - Segment length is 64Kbytes in the 80286 because the OA is always 16 bits because of its 16-bit internal architecture.

      - allows segment lengths of 64K bytes.
    - The 80386 and above use a 32-bit architecture that allow an OA of 32 bits, in protected mode.
      - allows segment lengths of 4G bytes
    - L bit in 64 bit descriptor selects 64 bit address when L=1, and 32 bit address when L=0
    - No base or limit in 64 bit descriptor
      - So base is fixed at 00 0000 0000H
    - AV bit in 80386 and above:
      - 1 for availability and 0 for unavailability
    - D bit indicates how 80386 Core2 instructions access register and memory data.
      - 0 for 16 bit, often called 16bit instruction mode for 16 bit OA and register
      - 1 for 32 bit , called 32 bit instruction mode
      - These overwrite default
    - Access Rights Byte: controls access to protected mode segment.
      - describes segment functions in the system
      - allows complete control over the segment
      - For a data/code segment, direction of growth is fixed. If it grows beyond, program is interrupted with a general protection fault
      - Can specify write protection (on/off)
      - only CS in 64 bit
    - null descriptor/descriptor zero contains all zeroes, may not be used to access memory
  - Selector:
    - Located in segment register.
    - 13 bit selector field: Selects one of 8192 descriptors from the table
    - a table selector bit(TI): selects which table
      - 0 = global
      - 1 = local
    - requested privilege level field(rpl): requests access privilege of memory segment
      - 00 = highest privilege
      - 11 = lowest privilege
      - Access is granted if privilege set by access rights byte is equal or lower than this field i.e. rpl should be higher.
      - In windows, 00 (ring 0) for kernel & driver and 11 (ring 3) for applications. no 01 or 10
      - privilege level violation indicated
- **Program-Invisible Registers:**
  - To access and specify the address of descriptor tables in 80286-Core2
  - Not directly addressed by the software
  - Control mp in protected mode
  - Contained in each segment register
    - often called cache memory
  - Loaded with base address, limit, access rights each time the number in segment register is changed.
  - When a new segment number is placed in a segment register, the mp accesses a descriptor table and loads the descriptor into the program-invisible portion of the segment register.
  - It is held there and used to access the memory segment until the segment number is again changed.
  - Allows repeated access until segment number is changed
  - Global Descriptor Table Register(GDTR) and Interrupt Descriptor Table Register(IDTR): contain base address of the descriptor table and its limit(16 bits)
    - GDTR is loaded to use protected mode
    - IDTR is initialised before using protected mode
  - One gd is set to address the ld table
    - Local Descriptor Table Register (LDTR) is loaded with a selector to access the ld table.
    - This selector accesses the gd table and loads the address, limit, and access rights of the ld table into the cache portion of the LDTR.
  - Task Register (TR):
    - holds a selector to access the descriptor that defines the task
    - A task is a procedure or application program
    - descriptor stored in gd table to control access
    - allows context switch in about 17 micro seconds

2.5 Flat Mode Memory

- Pentium based computers using 64 bit extensions use flat mode memory system.

- In this, there is no segmentation
- First byte = 00 0000 0000H, last byte = FF FFFF FFFFH(40 bits)
- segment register not used to address memory location but selects the privilege level
- CS register used (protected mode only) to select descriptor that defines access rights of only the CS
- The offset address is the actual physical address in 64 bit mode
- Easy to understand
- Little protection
- No real mode if 64 bit mode
- Protection and paging allowed
- If IA32 is set and L = 0, 40 bit addresses are used.
    - Addresses over this are truncated.
    - OA of 32 bit i.e. + or - 2G
    - Called RIP relative addressing
    - The move immediate instruction allows a full 64-bit address and access to any flat mode memory location.
    - Other instructions do not allow access to a location above 4G because the offset address is still 32-bits.
- If L = 1, address can be 64 or 32