Advanced Processor

80186 Basic Features

- The Intel 80186 is an improved version of the 8086 microprocessor. The 80186 contains 16-bit data bus and 20-bit address bus.
- The Intel 80186 family contains two microprocessors: Intel 80186 and 80188. The only difference between them is that the 80186 have a 16-bit data bus, while the 80188 include an 8-bit data bus.
- The 80186 is packaged in a 68-pin leadless package. The 80186 does not have the MN/MX# pin. The 80186 has enough pins to generate the minimum mode-type pins.
- The internal register structure of 80186 is virtually identical to the 8086.
- 80186 = 8086 + several additional chips
    - added 9 new instructions
    - clock generator
    - programmable timer
    - programmable interrupt controller
    - circuitry to select the I/O devices

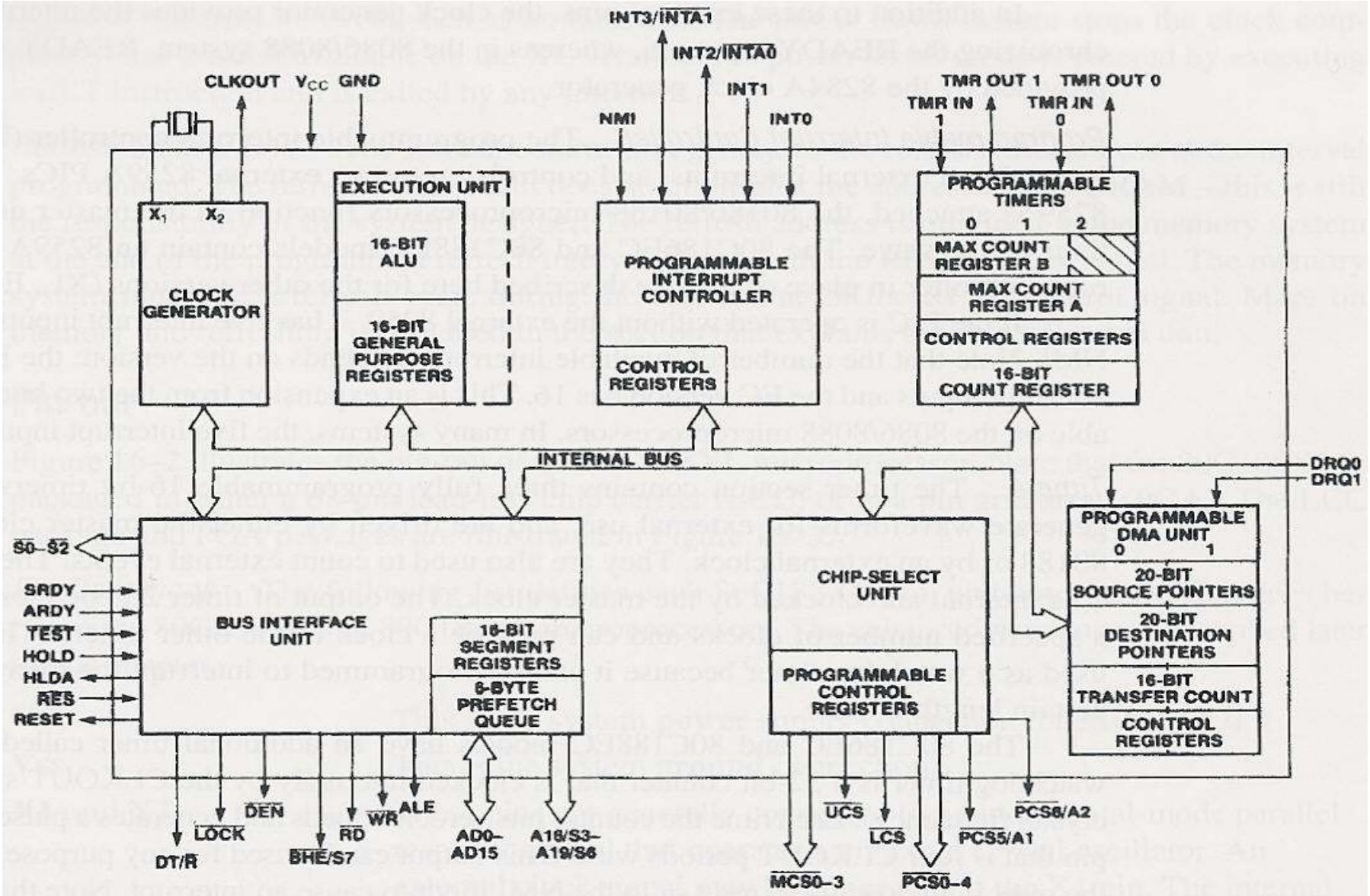


FIGURE –1 The block diagram of the 80186 microprocessor. Note that the block diagram of the 80188 is identical, except that $\overline{BHE}$/S7 is missing and AD15–AD8 are relabeled A15–A8. (Courtesy of Intel Corporation.)

80286

- 24-bit address bus that can address 16M bytes of memory directly
- There are two operating modes for 80286.
  - The real address mode
    - Can address up to 1MB of the physical memory
  - The protected virtual address mode
    - Multiuser and multitasking system
    - The memory management unit can manage upto 1 GB of the virtual memory though the real memory may be much less,only 16 MB.
- Enhanced with memory protection capabilities
  - One user do not interfere with the other. Also users cannot interfere with the operating system. These features are called protection.

80286

- THE 80286 contains four processing units:

  1. Bus unit
     - All memory and I/O read /write operations are performed by BU.
     - While the current instruction is being executed, the BU prefetches instructions and keeps them in a queue of six bytes.
  2. Instruction unit
     - The function of IU is to decode the perfected instructions and to maintain a queue of 3 decoded instructions for execution.
  3. Execution unit
     - The EU executes instruction.
  4. Address unit
     - The address unit computes address of memory or I/O devices, which is to be sent by BU for read and write operation.

  – All the four units work in parallel within the CPU. This type of parallel operation is called pipelining.

- Introduced protected mode
  – Segmentation in protected mode is different from the real mode

80386

- First 32-bit processor
  - 32-bit data bus and 32-bit address bus.
- Capable of addressing 4G bytes of physical memory.
- Can address 64 terabytes of the virtual memory through its memory management unit.
- The processor can operate in two modes:
  - Real and protected.
  - In the real mode physical address space is 1Mbytes (20 address lines), which is extended to 4G bytes in the protected mode (32 address lines).
- It has basically six functional units:
  1. Bus interface unit,
  2. Code pre-fetch unit,
  3. Instruction decode unit,
  4. Execution unit,
  5. Segmentation unit and
  6. Paging unit.

80486

- Improved version of 386

- The 486 processor includes:
  - Built in math coprocessor.
  - 8K byte of code and data cache memory on the chip.
  - Highly pipelined execution unit. Therefore the execution time for many instructions is one clock period.

- The 486 contains the following functional units:
  1. Execution unit
  2. Control unit
  3. Bus interface unit
  4. Code pre-fetch unit
  5. Instruction decode unit
  6. Segmentation unit
  7. Paging unit
  8. Cache unit
  9. Floating point unit.

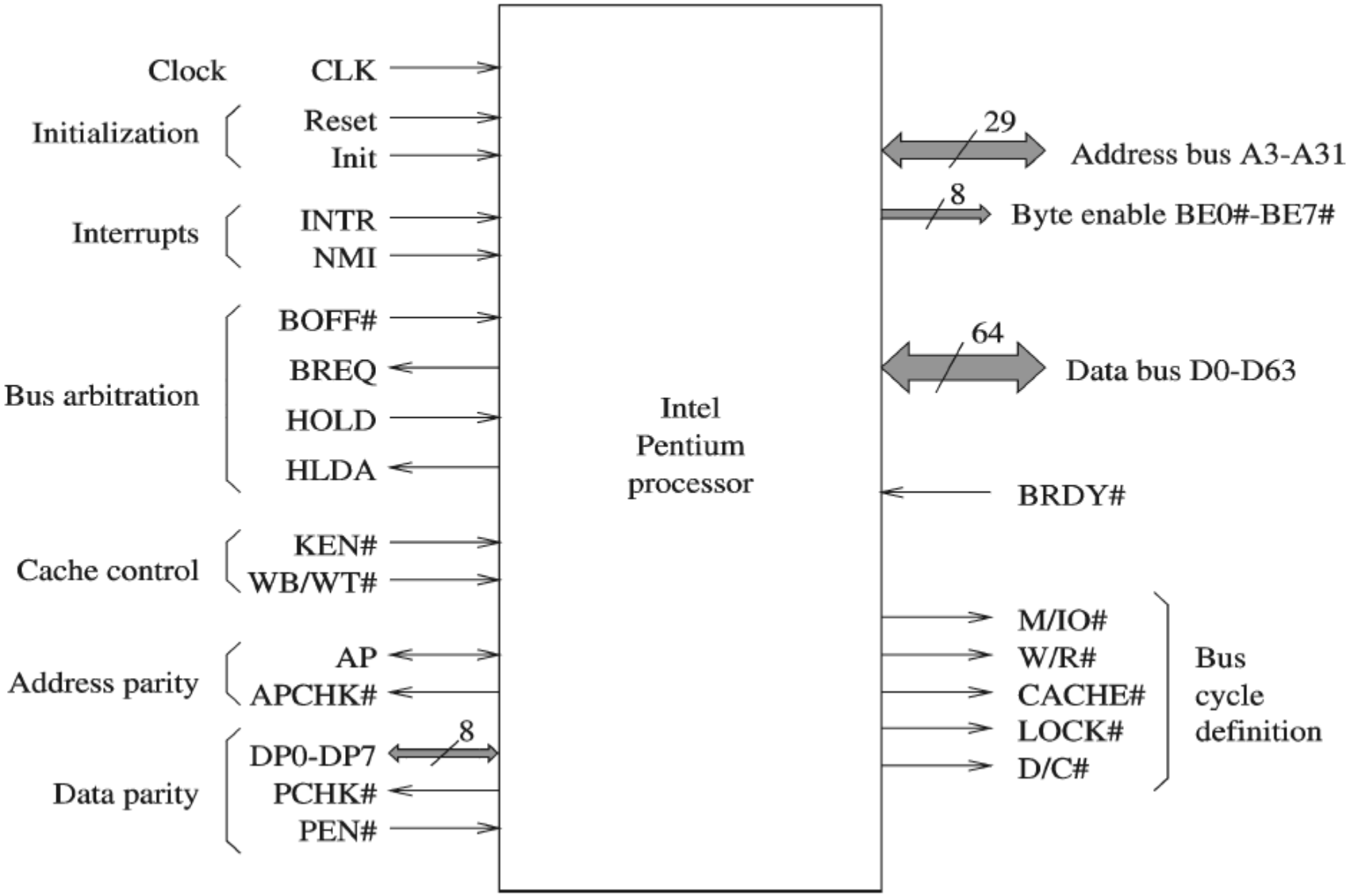
Pentium

* Pentium (80586) was introduced in 1993
  - » Similar to 486 but with 64-bit data bus
  - » Wider internal datapaths
    - – 128- and 256-bit wide
  - » Added second execution pipeline
    - – Superscalar performance
    - – Two instructions/clock
  - » Doubled on-chip L1 cache
    - – 8 KB data
    - – 8 KB instruction
  - » Added branch prediction

Pentium

* Pentium II was introduced in 1997
  - » Introduced multimedia (MMX) instructions
  - » Doubled on-chip L1 cache
    - – 16 KB data
    - – 16 KB instruction
  - » Introduced comprehensive power management features
    - – Sleep
    - – Deep sleep
  - » In addition to the L1 cache
    - – Has 256 KB L2 cache

Pentium Processor



- Data bus (D0 – D 63)
  - 64-bit data bus
- Address bus (A3 – A31)
  - Only 29 lines
    - No A0-A2 (due to 8-byte wide data bus)
- Byte enable (BE0# - BE7#)
  - Identifies the set of bytes to read or write
    - BE0# : least significant byte (D0 – D7)
    - BE1# : next byte (D8 – D15)
    - …
    - BE7# : most significant byte (D56 – D63)
  - Any combination of bytes can be specified

- Data parity (DP0 – DP7)
  - Even parity for 8 bytes of data
    - DP0 : D0 – D7
    - DP1 : D8 – D15
    - …
    - DP7 : D56 – D63
- Parity check (PCHK#)
  - Indicates the parity check result on data read
  - Parity is checked only for valid bytes
    - Indicated by BE# signals

- Parity enable (PEN#)
  - Determines whether parity check should be used
- Address parity (AP)
  - Bad address parity during inquire cycles
- Memory/IO (M/IO#)
  - Defines bus cycle: memory or I/O
- Write/Read (W/R#)
  - Distinguishes between write and read cycles
- Data/Code (D/C#)
  - Distinguishes between data and code

- Cacheability (CACHE#)
  - Read cycle: indicates internal cacheability
  - Write cycle: burst write-back
- Bus lock (LOCK#)
  - Used in read-modify-write cycle
  - Useful in implementing semaphores
- Interrupt (INTR)
  - External interrupt signal
- Nonmaskable interrupt (NMI)
  - External NMI signal

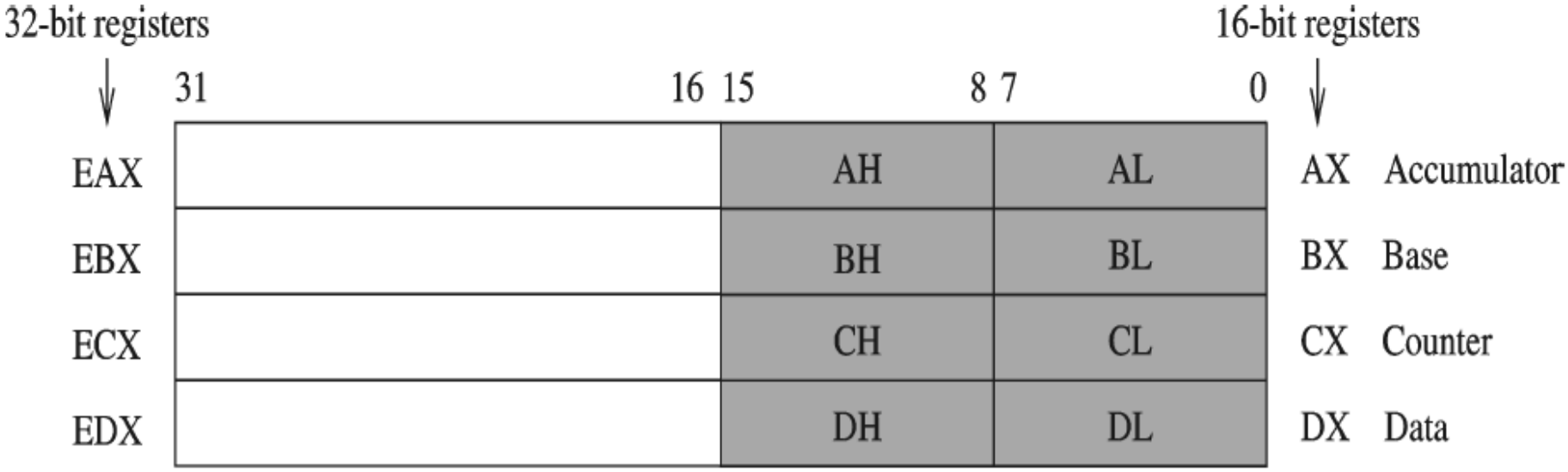
- Clock (CLK)
  - System clock signal
- Bus ready (BRDY#)
  - Used to extend the bus cycle
    - Introduces wait states
- Bus request (BREQ)
  - Used in bus arbitration
- Backoff (BOFF#)
  - Aborts all pending bus cycles and floats the bus
  - Useful to resolve deadlock between two bus masters

- Bus hold (HOLD)
  - Completes outstanding bus cycles and floats bus
  - Asserts HLDA to give control of bus to another master
- Bus hold acknowledge (HLDA)
  - Indicates the Pentium has given control to another local master
  - Pentium continues execution from its internal caches
- Cache enable (KEN#)
  - If asserted, the current cycle is transformed into cache line fill

- Write-back/Write-through (WB/WT#)
  - Determines the cache write policy to be used
- Reset (RESET)
  - Resets the processor
  - Starts execution at FFFFFFF0H
  - Invalidates all internal caches
- Initialization (INIT)
  - Similar to RESET but internal caches and FP registers are not flushed
  - After powerup, use RESET (not INIT)

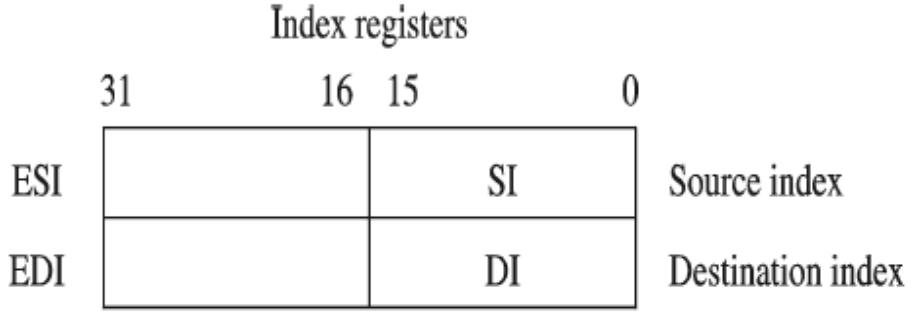
Pentium Registers

- Four 32-bit registers can be used as
  - Four 32-bit register (EAX, EBX, ECX, EDX)
  - Four 16-bit register (AX, BX, CX, DX)
  - Eight 8-bit register (AH, AL, BH, BL, CH, CL, DH, DL)
- Some registers have special use
  - ECX for count in loop instructions

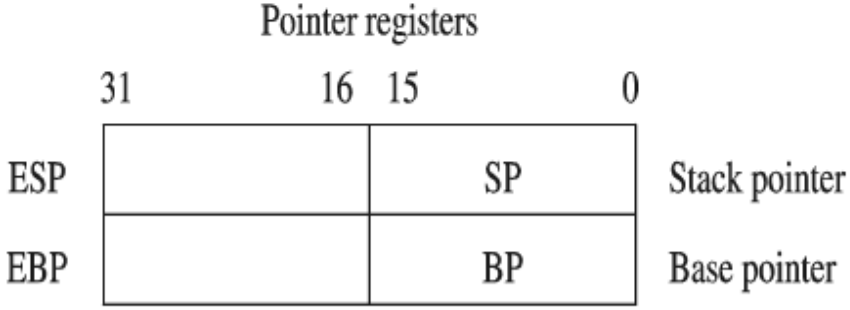


Pentium Registers (cont'd)

- Two index registers
  - 16- or 32-bit registers
  - Used in string instructions
    - » Source (SI) and destination (DI)
  - Can be used as general-purpose data registers

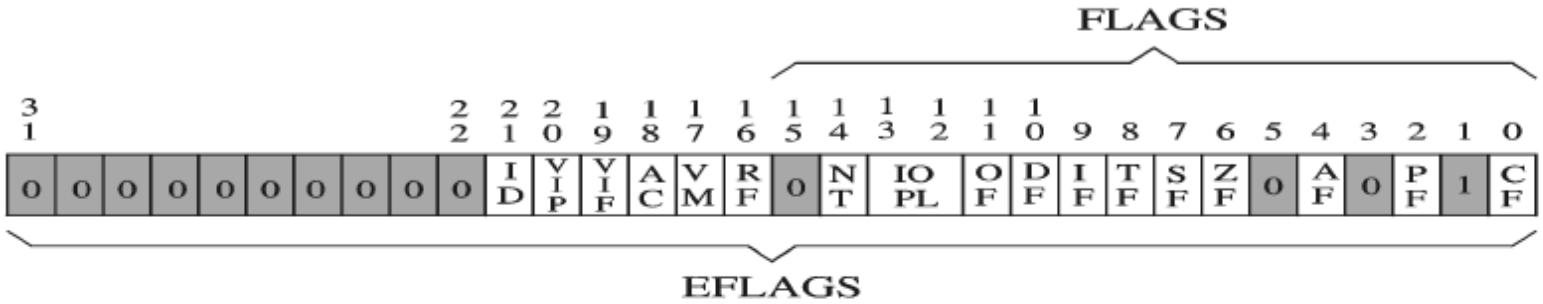


- Two pointer registers
  - 16- or 32-bit registers
  - Used exclusively to maintain the stack



Pentium Registers (cont'd)

Flags register



Status flags

CF = Carry flag
PF = Parity flag
AF = Auxiliary carry flag
ZF = Zero flag
SF = Sign flag
OF = Overflow flag

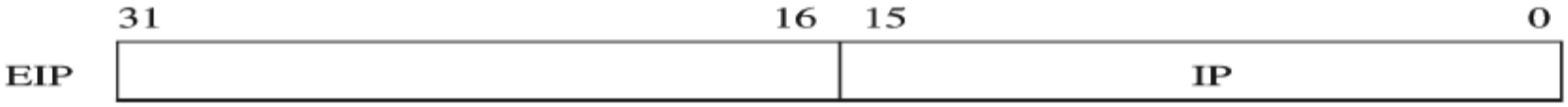
Control flags

DF = Direction flag

System flags

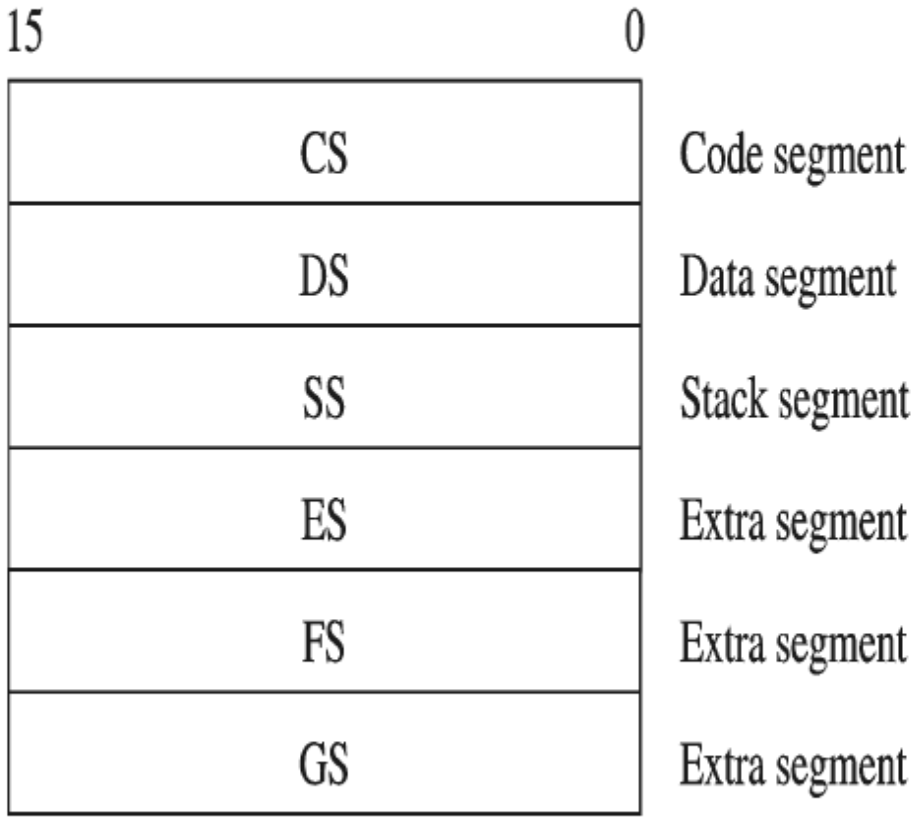
TF = Trap flag
IF = Interrupt flag
IOPL = I/O privilege level
NT = Nested task
RF = Resume flag
VM = Virtual 8086 mode
AC = Alignment check
VIF = Virtual interrupt flag
VIP = Virtual interrupt pending
ID = ID flag

Instruction pointer



Pentium Registers (cont'd)

- Segment register
  - Six 16-bit registers
  - Support segmented memory architecture
  - At any time, only six segments are accessible
  - Segments contain distinct contents
    - Code
    - Data
    - Stack

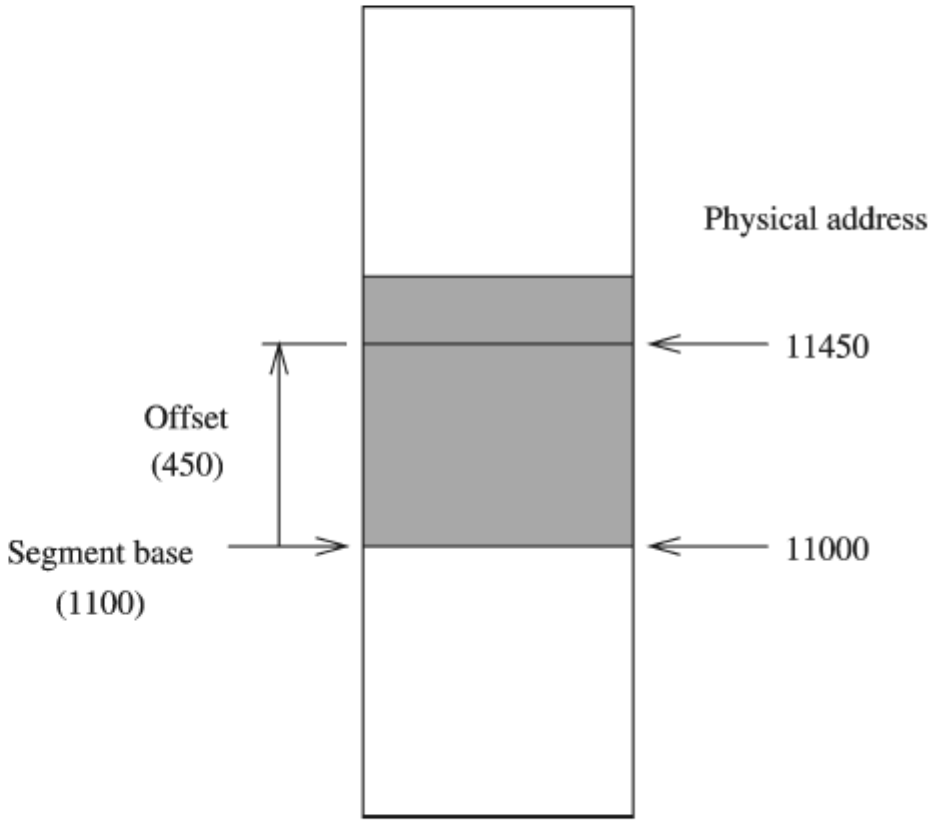


Real Mode Architecture

- Pentium supports two modes
  - * Real mode
    - » Uses 16-bit addresses
    - » Runs 8086 programs
    - » Pentium acts as a faster 8086
  - * Protected mode
    - » 32-bit mode
    - » Native mode of Pentium
    - » Supports segmentation and paging

Real Mode Architecture (cont'd)

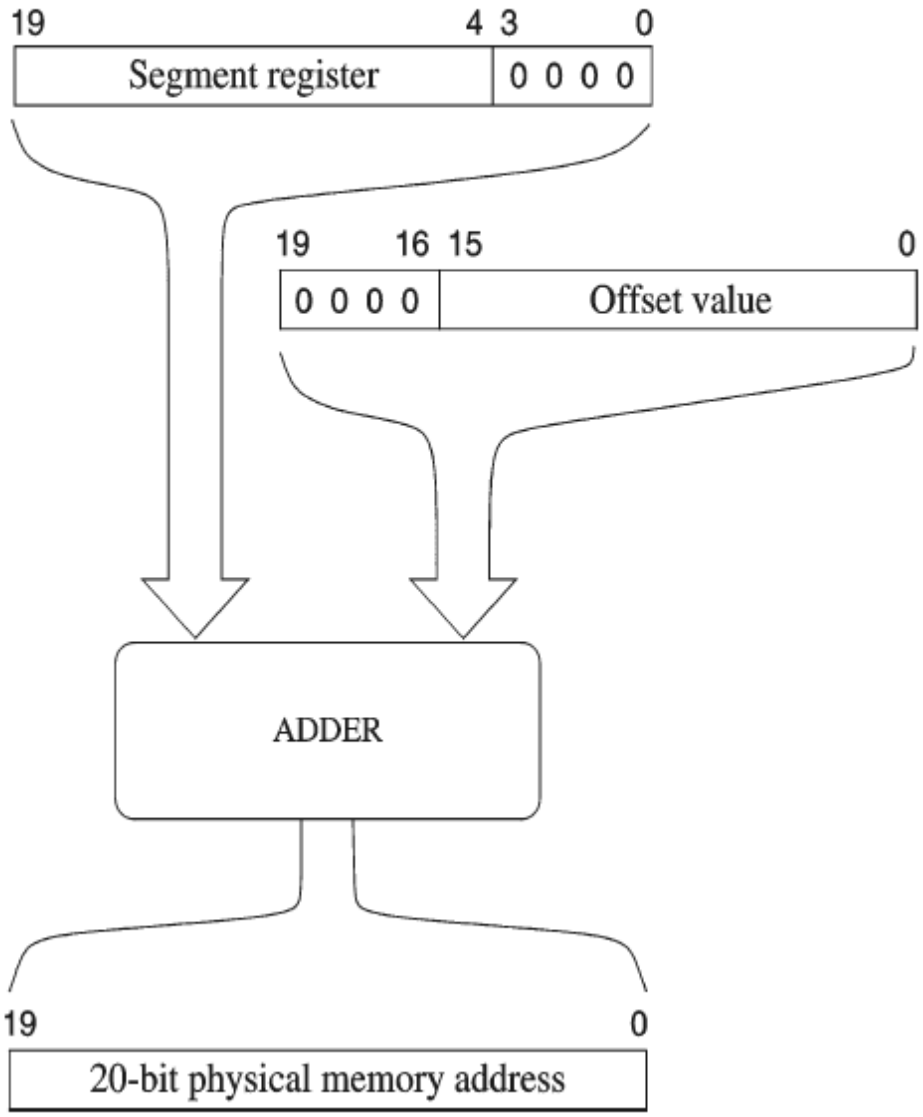
- Segmented organization
  - 16-bit wide segments
  - Two components
    - Base (16 bits)
    - Offset (16 bits)
- Two-component specification is called ***logical address***
  - Also called ***effective address***
- 20-bit ***physical address***



Real Mode Architecture (cont'd)

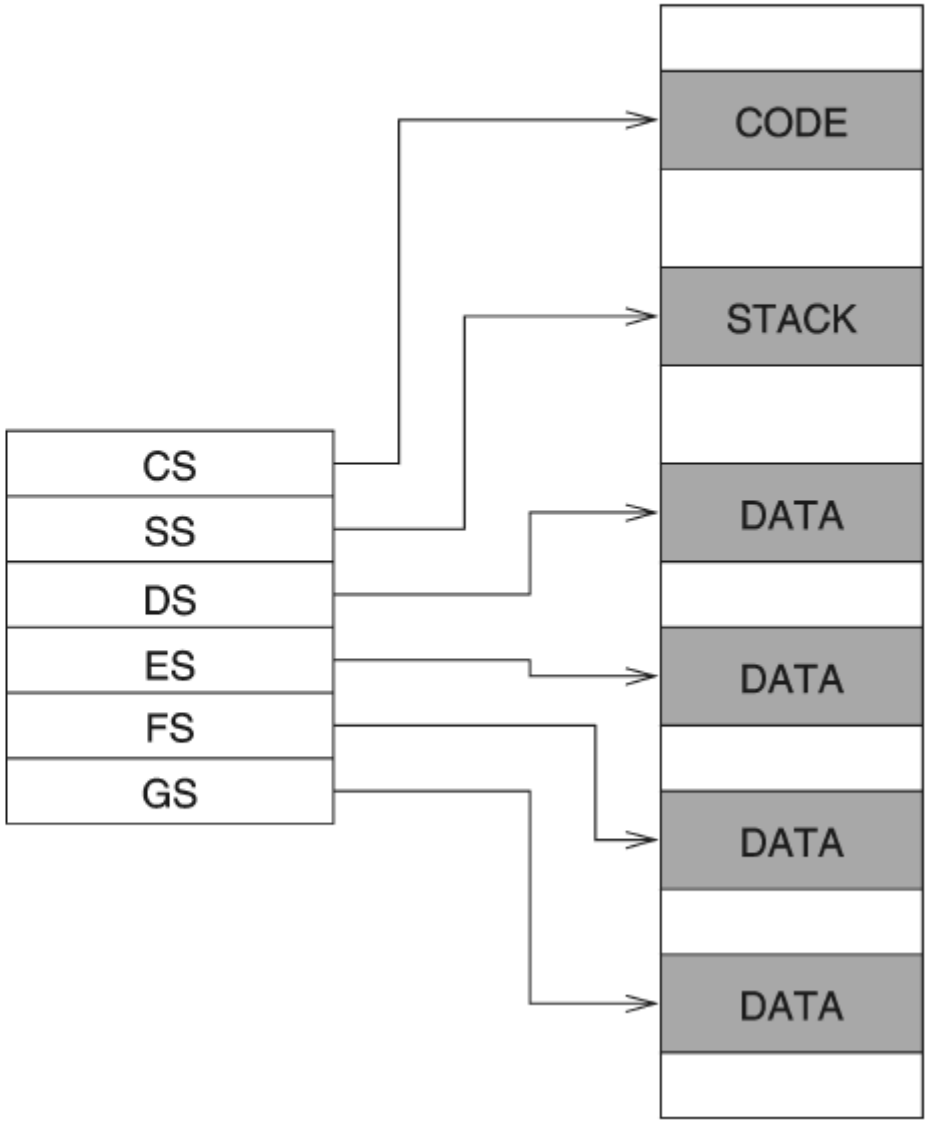
- Conversion from logical to physical addresses

```
  11000  (add 0 to base)
+   450  (offset)
-------
  11450  (physical address)
```



Real Mode Architecture (cont'd)

- Programs can access up to six segments at any time
- Two of these are for
  - Data
  - Code
- Another segment is typically used for
  - Stack
- Other segments can be used for
  - data, code,..



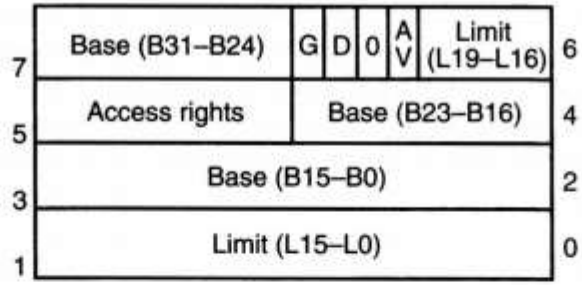
PROTECTED-MODE

- In the protected-mode, memory **larger** than **1 MB** can be accessed. **Windows XP** operates in the **protected mode**.
- In addition, **segments** can be of **variable size** (below or above **64 KB**).
- Some **system control instructions** are *only* valid in the protected mode.
- In protected mode, the base:offset logical memory addressing scheme (which is used in real mode) is changed.
- The offset part of the logical memory address is still used. However, when in the protected mode, the processor can work either with 16-bit offsets (the 16-bit instruction mode) or with 32-bit offsets (the 32-bit instruction mode). A 32-bit offset allows segments of up to **4G bytes** in length. Notice that in real-mode the only available instruction mode is the 16-bit mode (during which accessing 32-bit registers requires the prefix 66h).
- However, the segment base address calculation is different in protected mode. Instead of appending a 0 at the end of the segment register contents to create a segment base address (which gives a 20-bit physical address), 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. This is similar to selecting one card from a deck of cards in one's pocket.
- Because the segment register and offset address still create a logical memory address, protected mode **instructions** are the same as real mode instructions. In fact, most programs written to function in the real mode will function without change in the protected mode.

DESCRIPTORS:

- The selector, located in the segment register, selects one of **8192 descriptors** from one of two tables of descriptors (stored in memory): the global and local descriptor tables. The descriptor describes the **location**, **length** and **access rights** of the memory segment.
- Each descriptor is **8 bytes long** and its format is shown below:

80386 through Pentium 4 descriptor



- The 8192 descriptor table requires 8 * 8192 = **64K bytes** of memory. The main parts of a descriptor are:
- **Base (B31 – B0):** indicates the starting location **(base address)** of the memory segment. This allows segments to begin at any location in the processor's 4G bytes of memory.
- **Limit (L19 – L0):** contains the **last offset address** found in a segment. Since this field is 20 bits, the segment size could be anywhere between 1 and 1M bytes. However, if the **G bit (granularity bit)** is set, the value of the limit is multiplied by **4K bytes** (i.e., appended with FFFH). In this case, the segment size could be anywhere between 4K and 4G bytes in steps of 4K bytes.

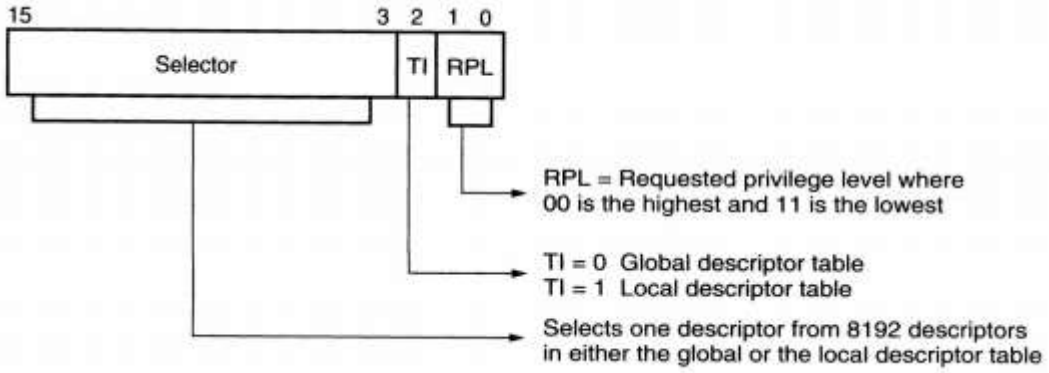
- Example,
- Base = Start = **10000000h**
- Limit = **001FFh** and G = **0**
- So, End = Base + Limit = 10000000h + 001FFh = 100001FFh
- Segment Size = **512 bytes**

- Base = Start = **10000000h**
- Limit = **001FFh** and G = **1**
- So, End = Base + Limit * 4K = 10000000h + **001FFFFFh = 101FFFFFh**
- Segment Size = **2M bytes**

- **AV bit:** is used by some operating systems to indicate that the segment is available (AV = 1) or not available (AV = 0).
- **D bit:** If D = 0, the instructions are 16-bit instructions, compatible with the 8086-80286 microprocessors. This means that the instructions use 16-bit offset addresses and 16-bit registers by default. This mode is the 16-bit instruction mode or DOS mode. If D = 1, the instructions are 32-bits by default (Windows XP works in this mode). By default, the 32-bit instruction mode assumes that all offset addresses and all registers are 32 bits. Note that the default for register size and offset address can be overridden in both the 16- and 32-bit instruction modes using the 66h and 67h prefixes. In 16-bit protected-mode, descriptors are still used but segments are supposed to be a maximum of 64K bytes.
- **Access rights byte:** allows complete control over the segment. If the segment is a data segment, the direction of growth is specified. If the segment grows beyond its limit, the microprocessor's operating system program is interrupted, indicating a **general protection fault**. You can specify whether a data segment can be written or is write-protected. The code segment can have reading inhibited to protect software. This is why it is called protected mode. This kind of protection is unavailable in real-mode.

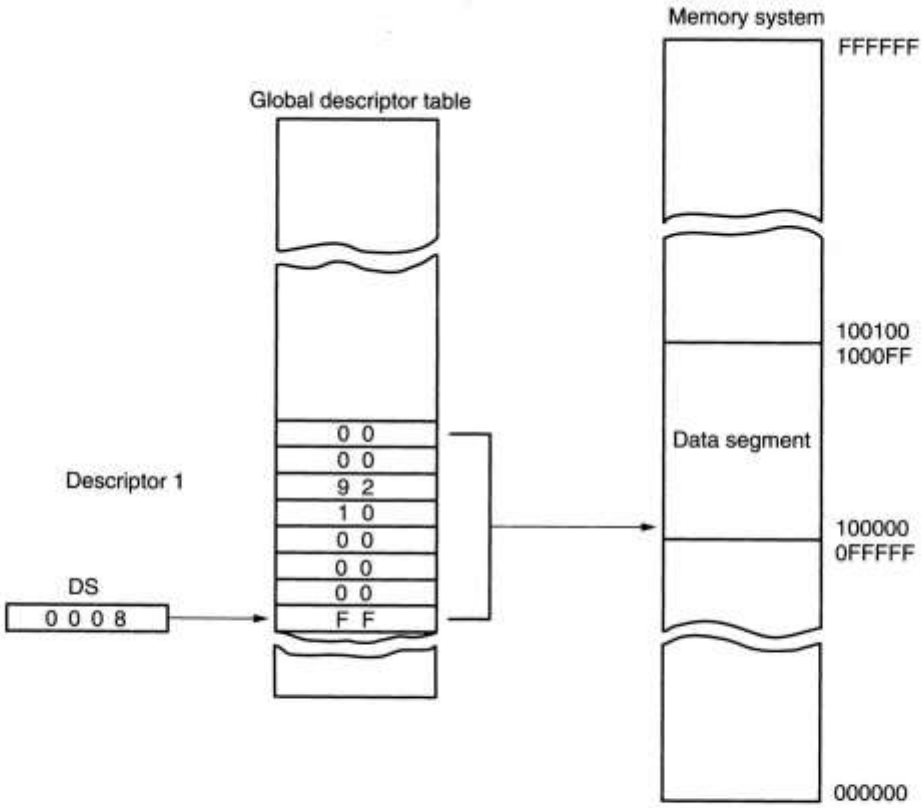
SELECTORS:

- Descriptors are chosen from the descriptor table by the segment register. There are two descriptor tables:
- **Global descriptors table:** contains segment definitions that apply to **all** programs (also called **system descriptors**).
- **Local descriptors table:** usually unique to an application (also called **application descriptors**).
- Each descriptor table contains 8192 descriptors, so a total of 16,384 descriptors are available to an application at any time. This allows up to 16,384 memory segments to be described for each application.
- The Figure below shows the segment register in the protected mode. It contains:



- **13-bit selector field:** chooses one of the 8192 descriptors from the descriptor table ($2^{13} = 8192$).
- **Table indicator (TI) bit:** selects either the global descriptor table (TI = 0) or the local descriptor table (TI = 1).
- **Requested privilege level (RPL) field:** requests the access privilege level of a memory segment. The highest privilege level is 00 and the lowest is 11. If the requested privilege level matches or is higher in priority than the privilege level set by the access rights byte, access is granted. Windows uses privilege level 00 (ring 0) for the kernel and driver programs and level 11 (ring 3) for applications. Windows does not use levels 01 or 10. If privilege levels are violated, the system normally indicates a **privilege level violation**.

- Example:
- **Real Mode: DS = 0008h**, then the data segment begins at location **00080h** and its length is **64K bytes**.
- **Protected Mode: DS = 0008h** = 0000 0000 0000 **1000**, then the selector selects **Descriptor 1** in the **global** descriptor table using a requested privilege level of **00**. The global descriptor table is stored in memory as shown below.



- Descriptor number 1 contains a descriptor that defines the base address as 00100000h with a segment limit of 000FFh. This refers to memory locations **00100000h – 001000FFh** for the data segment.