Lecture 25: 80x86 Evolution 1

Seyed-Hosein Attarzadeh-Niaki

Based on the slides by Barry Brey

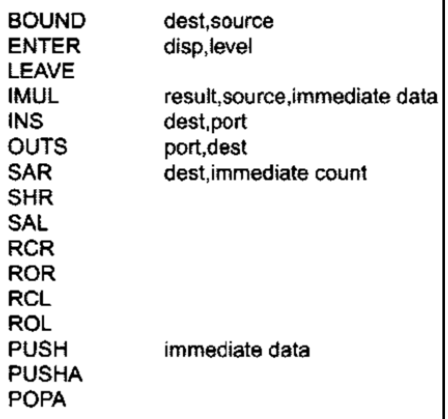
Review

- Shared bus access
- Bus interfaces
  - ISA
  - PCI
- Direct Memory Access and 8237
- DMA-Controlled IO

80186/188

- Successful in the embedded controller market
- Never used by IBM in their family of PC products
- Put a portion of peripheral chips along with the 8086/88 on a single chip
  - clock generator, two 20-bit DMA channels, three 16-bit programmable counters, interrupt controller, programmable wait-state generator, and programmable chip select decoder unit.

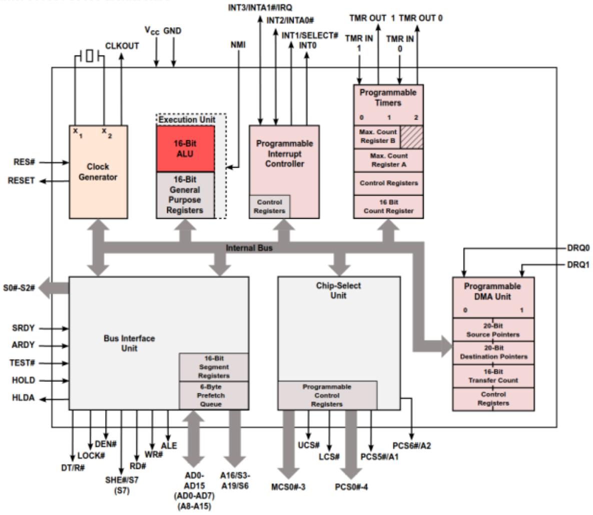
- New instructions

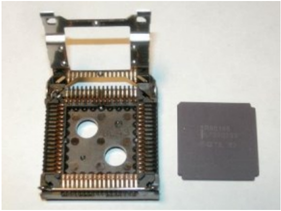


BOUND	dest,source
ENTER	disp,level
LEAVE	
IMUL	result,source,immediate data
INS	dest,port
OUTS	port,dest
SAR	dest,immediate count
SHR	
SAL	
RCR	
ROR	
RCL	
ROL	
PUSH	immediate data
PUSHA	
POPA	

80186 Chip

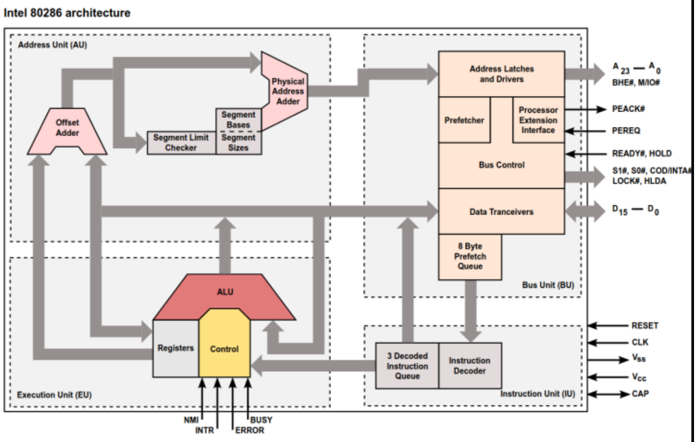
Intel 80186 / 80188 architecture



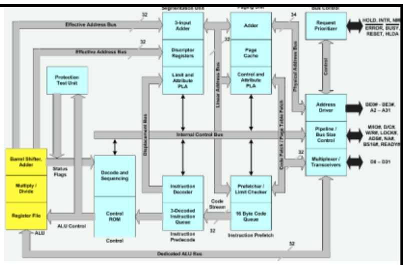


80286

- Separate pins for the address and data buses
  - no need for demultiplexing
- Memory cycle time was reduced to 2 clocks
- Introduction of virtual memory (*protected virtual-address mode*)
  - Real and protected modes of operation
- Non-overlapping code, data and stack segments
- Privileged segments



80386



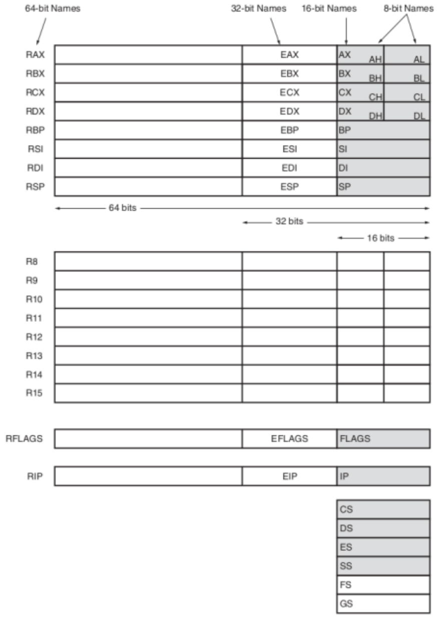
- 32-bit data bus (internally and externally)
- Registers were extended to 32 bits
- Address bus was increased to 32 bits
- Paging virtual memory mechanism was introduced
  - Capable of both segmentation and paging
- Can use general registers also as pointers
- New addressing mode called scaled index
- New bit-manipulation instructions
- Can be switched from protected to real mode by software

The Programming Model

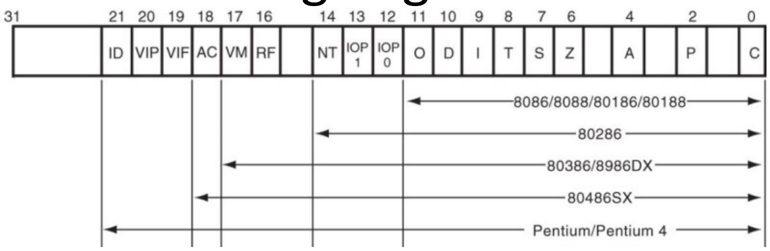
- 8086 through Core2 considered **program visible**
  - registers which are used during programming and are specified by the instructions
- Other registers considered to be **program invisible.**
  - not addressable directly during applications programming
- 80286 and above contain program-invisible registers to control and operate protected memory.
  - and other features of the microprocessor
- 80386 through Core2 microprocessors contain full 32-bit internal architectures.
- 8086 through the 80286 are fully upward-compatible to the 80386 through Core2.

The Programming Model (Including 64-bit Extensions)

- **R8 - R15** found in the Pentium 4 and Core2 if 64-bit extensions are enabled



Flag Register



- **IOP:** used in protected mode operation to select the privilege level for I/O devices.
- **NT (nested task):** flag indicates the current task is nested within another task in protected mode operation.
- **RF (resume):** used with debugging to control resumption of execution after the next instruction.
- **VM (virtual mode):** flag bit selects virtual mode operation in a protected mode system.
- **AC, (alignment check):** flag bit activates if a word or doubleword is addressed on a non-word or non-doubleword boundary.
- **VIF** is a copy of the interrupt flag bit available to the Pentium 4–**(virtual interrupt)**
- **VIP (virtual)** provides information about a virtual mode interrupt for **(interrupt pending)** Pentium.
  - used in multitasking environments to provide virtual interrupt flags
- **ID (identification):** flag indicates that the Pentium microprocessors support the CPUID instruction.

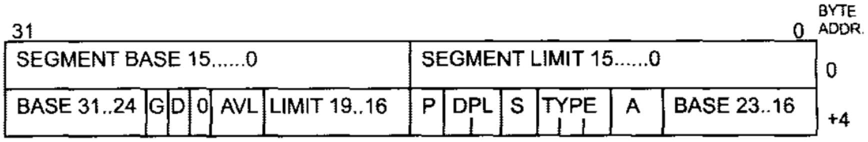
Real Mode Memory Addressing

- 80286 and above operate in the *real* or *protected* mode.
- **Real mode operation** allows addressing the first 1MB of memory
  - called the **real memory, conventional memory**, or **DOS memory** system
- A program placed in memory by DOS is loaded in the **TPA** at the first available area of memory above drivers and other TPA programs
  - The transient program area (TPA) holds the operating system; other programs that control the computer system.
- Segment plus offset addressing allows DOS programs to be relocated in memory.
  - In a *relocatable program*, the complete segment can be moved.

Protected Mode Memory Addressing

- Instead of a segment address, the segment register contains a **selector** that selects a descriptor from a descriptor table.
- The **descriptor** in the segment register describes the memory segment's location, length, and access rights.
- **Global descriptors** (system descriptors) contain segment definitions that apply to all programs.
- **Local descriptors** (application descriptors) are usually unique to an application.

Descriptors

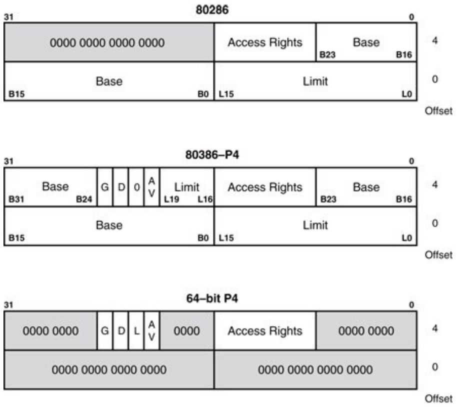


BASE	Base Address of the segment
LIMIT	The length of the segment
P	Present Bit 1 = Present 0 = Not Present
DPL	Descriptor Privilege Level 0 - 3
S	Segment Descriptor 0 = System Descriptor 1 = Code or Data Segment Descriptor
TYPE	Type of Segment (3 bits: X, E, R/W)
A	Accessed Bit
G	Granularity Bit 1 = Segment length is page granular 0 = Segment length is byte granular
D	Default Operation Size (code segment descriptors only) 1 = 32-bit segment 0 = 16-bit segment
0	Bit must be zero for compatibility with future processors
AVL	Available field for user or OS

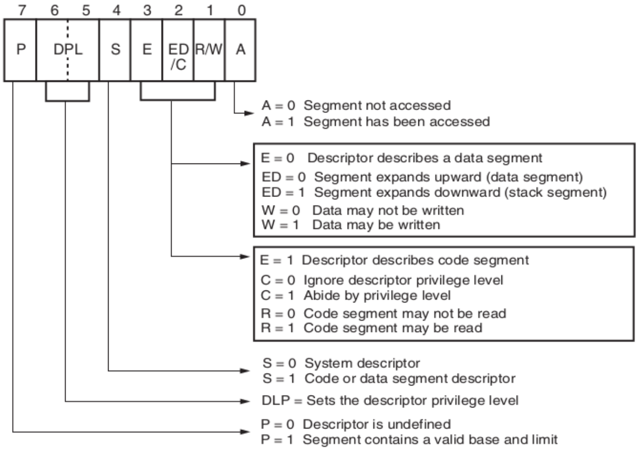
Note: In a maximum-size segment (i.e., a segment with G=1 and segment limit 19...0 = FFFFFH), the lowest 12 bits of the segment base should be zero (i.e., segment base 11...000 = 000H).

Descriptors

- Base: base address of the segment
- Limit: the length of the segment
- G: granularity Bit
  - 1 = Segment length is page granular (4GB in steps of 4K)
  - 0 = Segment length is byte granular (4KB)



Access Rights Byte



A = 0 Segment not accessed
A = 1 Segment has been accessed

E = 0 Descriptor describes a data segment
ED = 0 Segment expands upward (data segment)
ED = 1 Segment expands downward (stack segment)
W = 0 Data may not be written
W = 1 Data may be written

E = 1 Descriptor describes code segment
C = 0 Ignore descriptor privilege level
C = 1 Abide by privilege level
R = 0 Code segment may not be read
R = 1 Code segment may be read

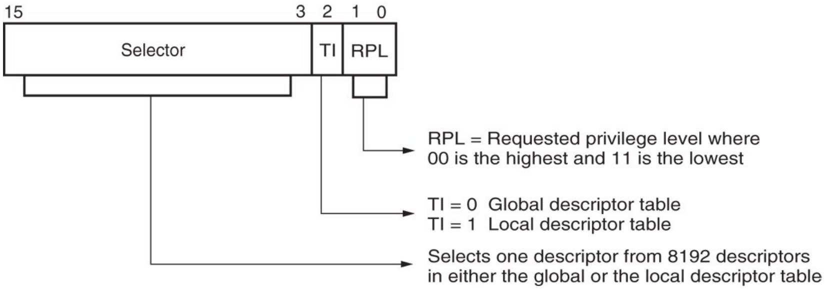
S = 0 System descriptor
S = 1 Code or data segment descriptor

DLP = Sets the descriptor privilege level

P = 0 Descriptor is undefined
P = 1 Segment contains a valid base and limit

Note: Some of the letters used to describe the bits in the access rights bytes vary in Intel documentation.

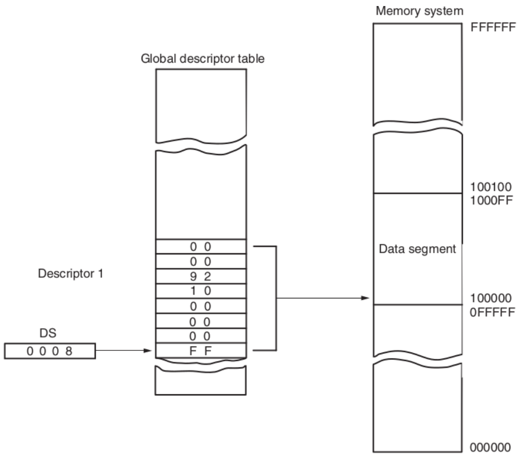
Segment Register in Protected Mode


15
3
2
1
0
Selector
TI
RPL
RPL = Requested privilege level where
00 is the highest and 11 is the lowest
TI = 0 Global descriptor table
TI = 1 Local descriptor table
Selects one descriptor from 8192 descriptors
in either the global or the local descriptor table

- The **TI bit** selects either the global or the local descriptor table.
- **Requested Privilege Level** (RPL) requests the access privilege level of a memory segment.

Choosing a Descriptor

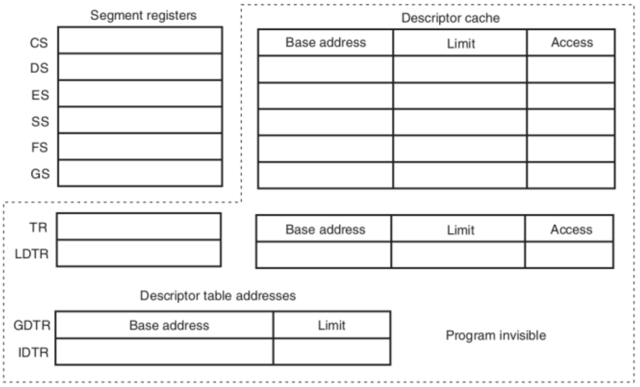
- The entry in the global descriptor table selects a segment in the memory system.
- Descriptor zero is called the null descriptor, must contain all zeros, and may not be used for accessing memory.
- In this example, the DS register accesses memory locations 00100000H–001000FFH as a data segment.



Program-Invisible Registers

- Global and local descriptor tables are found in the memory system.
- To access & specify the table addresses, 80286–Core2 contain program-invisible registers.
- Each segment register contains a program-invisible portion used in the protected mode.
- When a new segment number is placed in a segment register,
  - the microprocessor accesses a descriptor table and
  - loads the descriptor into the program-invisible portion of the segment register.
- This allows the microprocessor to repeatedly access a memory segment without referring to the descriptor table.

Program-Invisible Registers



Notes:

1. The 80286 does not contain FS and GS nor the program-invisible portions of these registers.
2. The 80286 contains a base address that is 24-bits and a limit that is 16-bits.
3. The 80386/80486/Pentium/Pentium Pro contain a base address that is 32-bits and a limit that is 20-bits.
4. The access rights are 8-bits in the 80286 and 12-bits in the 80386/80486/Pentium–Core2.

Program-Invisible Registers

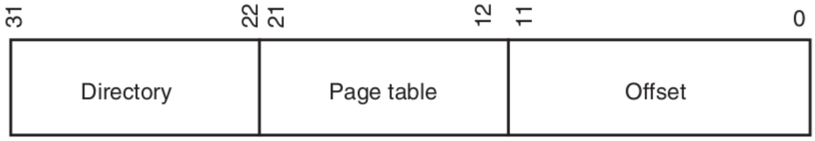
- The GDTR (**global descriptor table register**) and IDTR (**interrupt descriptor table register**) contain the base address of the descriptor table and its limit.
- To access the local descriptor table, the LDTR (**local descriptor table register**) is loaded with a selector.
  - selector accesses global descriptor table, & loads local descriptor table address, limit, & access rights into the cache portion of the LDTR
- The TR (task register) holds a selector, which accesses a descriptor that defines a task.
  - a task is most often a procedure or application
  - Allows multitasking systems to switch tasks to another in a simple and orderly fashion.

64 Terabytes of Virtual Memory

- 14 bits of the selector (segment) register
- Each can hold addresses of memory chunks as large as 4 gigabytes (segment limit)
- 64 terabytes of virtual memory for the 386
- Drawbacks of 386 segmentation
  - variable segment size: memory fragmentation
  - Absence of a *dirty bit* in the access byte of the descriptor table

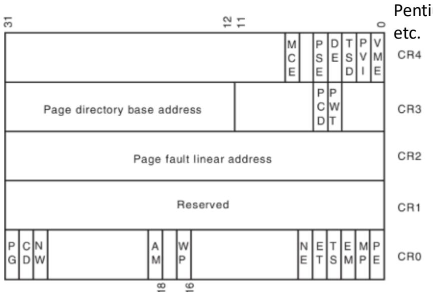
Memory Paging

- Paging: invisibly translate a linear address to a physical address
  - Linear address: address generated by the program
  - Physical address: actual memory location
- Linear address broken into: **page directory entry**, **page table entry**, and **memory page offset address**.

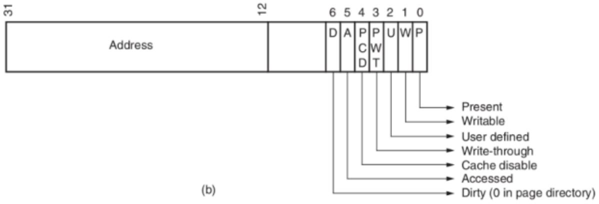


Paging Registers and A Page Table Entry

- PG: 1->paging enabled
  0->physical address
- CR3: page directory base address
- Beginning with Pentium, an additional control register labeled CR4 controls extensions to the basic architecture.

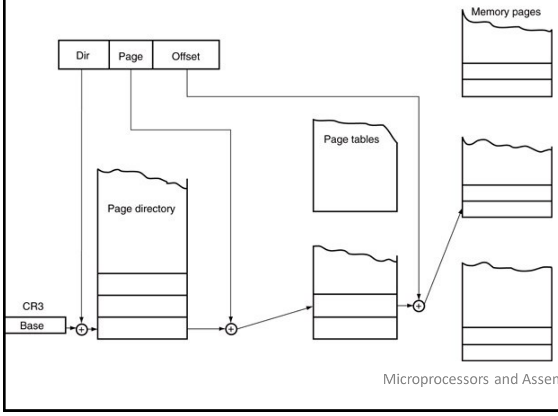


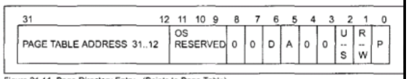
Pentium, etc.



Page Directory and Page Table

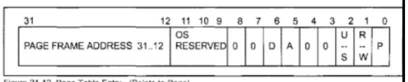
- Only one page directory in the system.
- The page directory contains 1024 doubleword addresses that locate up to 1024 page tables.
- Page directory and each page table are 4K bytes in length.





31 12	11 10 9	8	7	6	5	4	3	2	1	0
PAGE TABLE ADDRESS 31..12	OS RESERVED	0	0	D	A	0	0	U -- S	R -- W	P

Figure 21-11. Page Directory Entry (Points to Page Table)
(Reprinted by permission of Intel Corporation. Copyright Intel Corp. 1992)



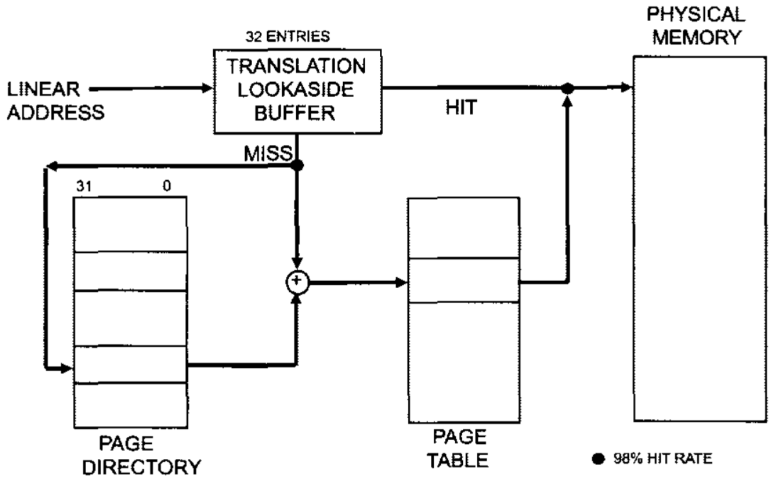
31 12	11 10 9	8	7	6	5	4	3	2	1	0
PAGE FRAME ADDRESS 31..12	OS RESERVED	0	0	D	A	0	0	U -- S	R -- W	P

Figure 21-12. Page Table Entry (Points to Page)
(Reprinted by permission of Intel Corporation. Copyright Intel Corp. 1992)

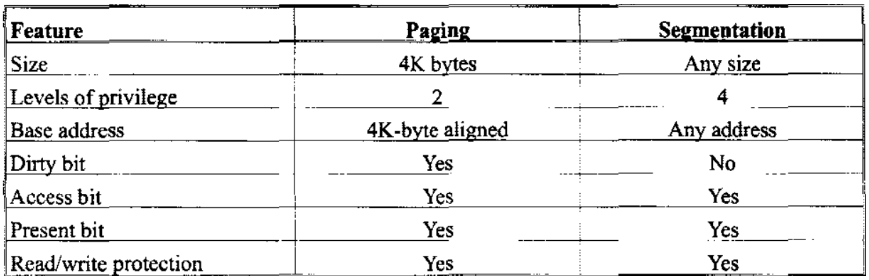
Translation Look-aside Buffer

- Intel has incorporated a special type of cache called TLB (**translation look-aside buffer**).
  - because repaging a 4K-byte section of memory requires access to the page directory and a page table, both located in memory
- The 80486 cache holds the 32 most recent page translation addresses.
  - if the same area of memory is accessed, the address is already present in the TLB
  - This speeds program execution
- Pentium contains separate TLBs for each of their instruction and data caches.

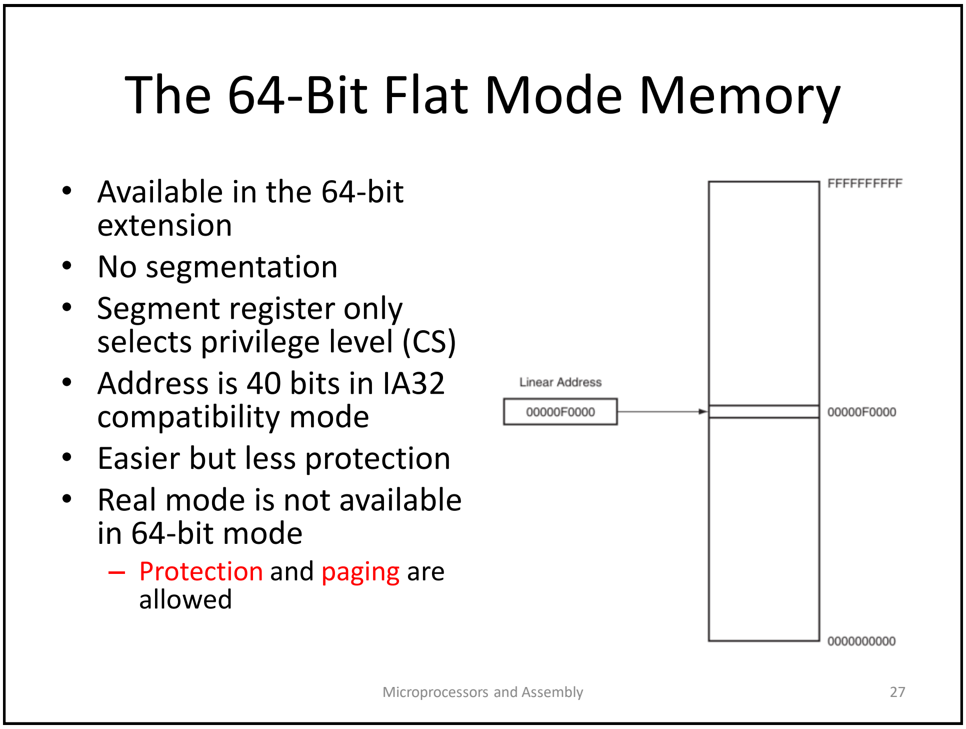
Translation Look-aside Buffer



Comparing Paging and Segmentation



Feature	Paging	Segmentation
Size	4K bytes	Any size
Levels of privilege	2	4
Base address	4K-byte aligned	Any address
Dirty bit	Yes	No
Access bit	Yes	Yes
Present bit	Yes	Yes
Read/write protection	Yes	Yes



The 64-Bit Flat Mode Memory

- Available in the 64-bit extension
- No segmentation
- Segment register only selects privilege level (CS)
- Address is 40 bits in IA32 compatibility mode
- Easier but less protection
- Real mode is not available in 64-bit mode
  - Protection and paging are allowed

FFFFFFFFFF
Linear Address
00000F0000
00000F0000
0000000000