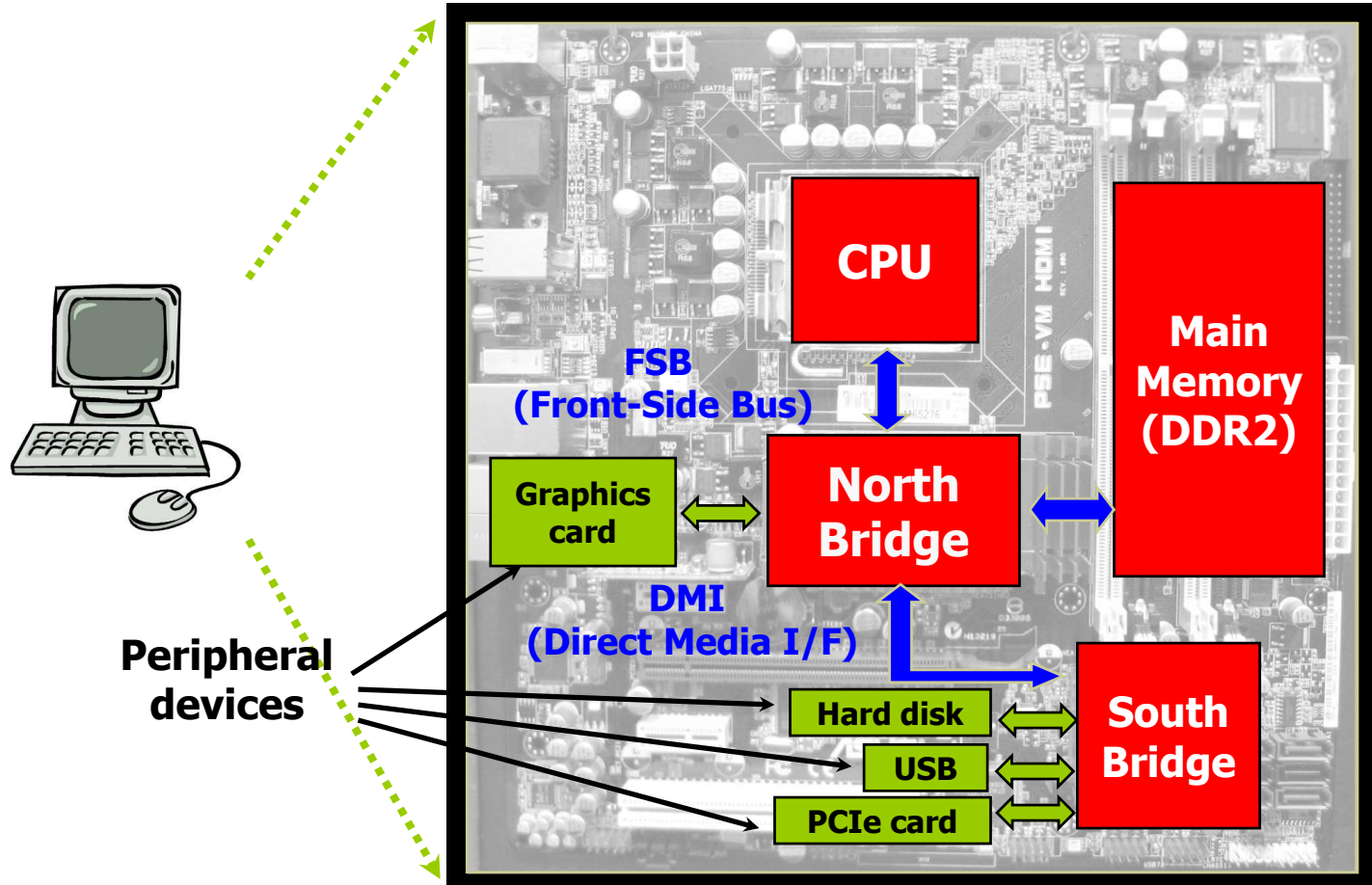


COMP427 Embedded Systems

Lecture 2. General-Purpose (GP) Computer Systems

Prof. Taeweon Suh
Computer Science Education
Korea University

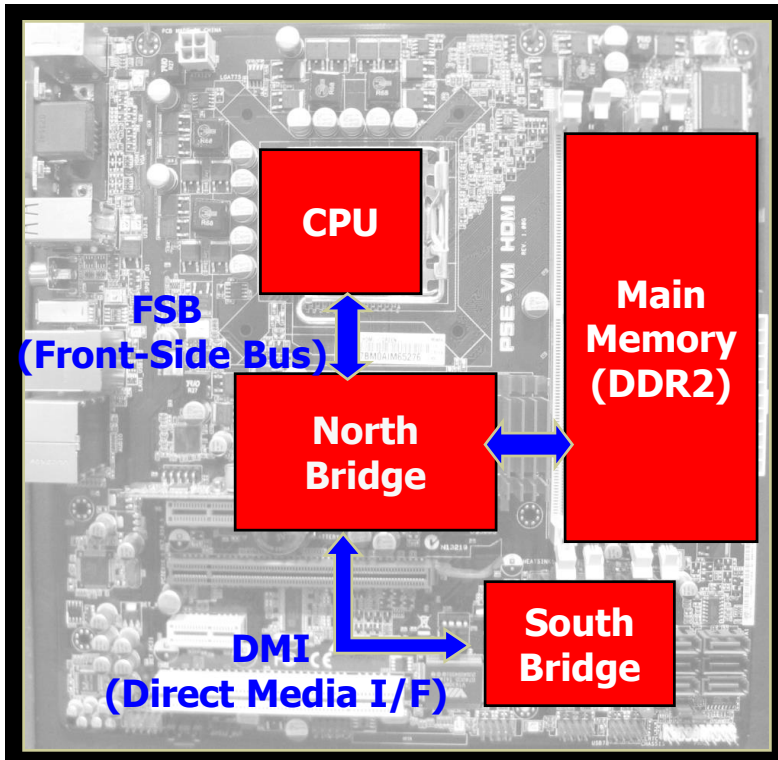
A Computer System (as of 2008)



But, don't forget the big picture!

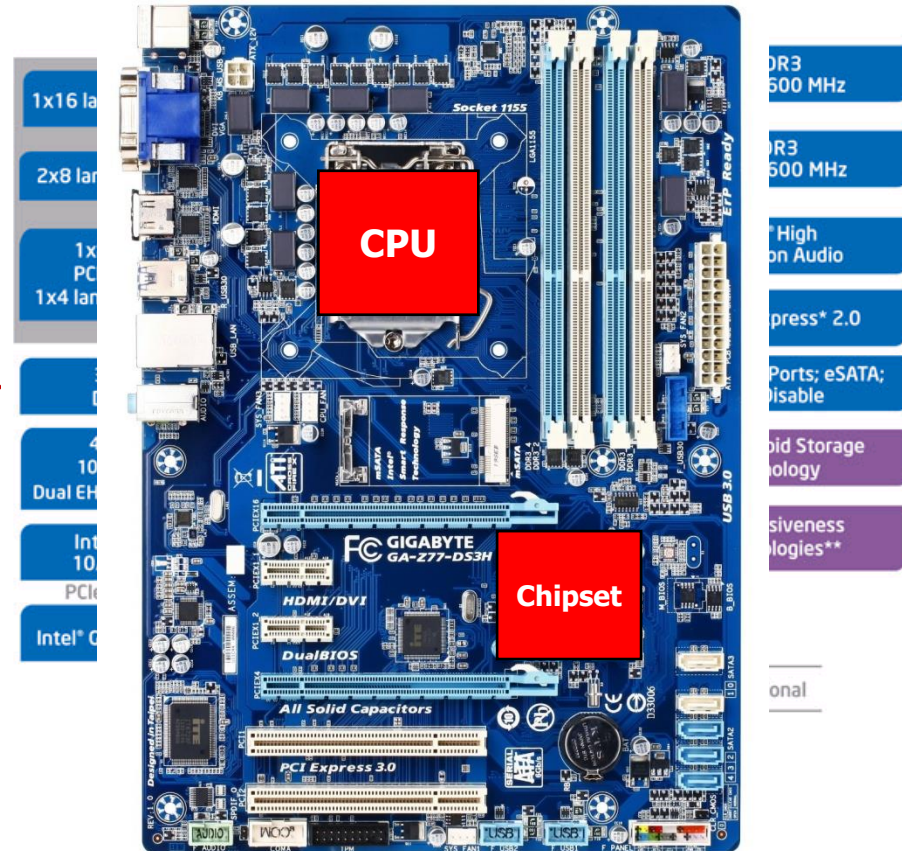
Past, Present and More...

- Core 2 Duo – based Systems



Keep in mind that CPU and computer systems are evolving at a fast pace!

- Core i7 (Ivy Bridge) – based Systems



- FDI: Flexible Display Interface
- SPI: Serial Peripheral Interface
- SMBus: System Management Bus

x86 History (as of 2008)

40 years of changing the world

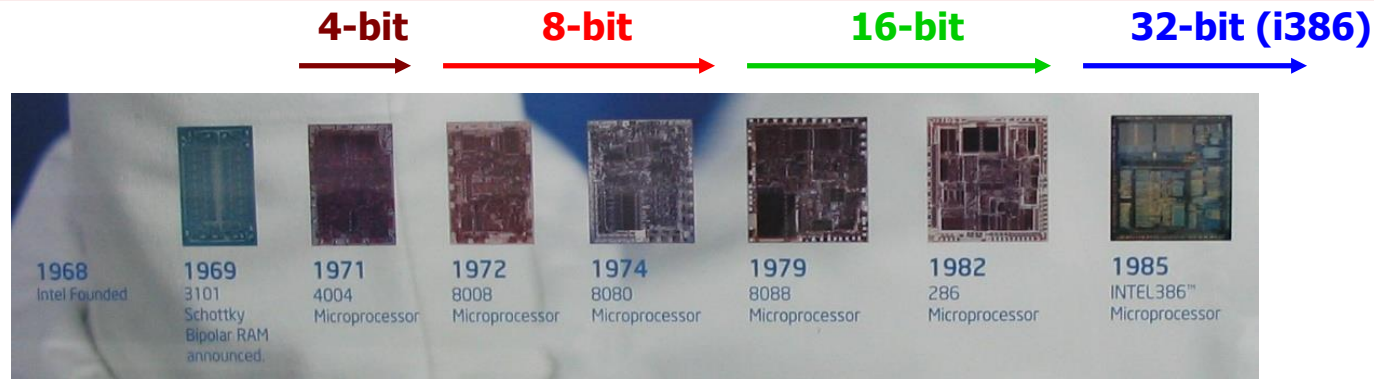
Since Intel's earliest days, we've harnessed the power of the transistor to transform what's possible for people around the world. In our first 40 years we've redefined how people live, work and play. Imagine what we can do together in the next 40.

Learn more at next40.intel.com

intel Leap ahead™

Year	Processor
1968	Intel 4004
1969	Intel 8008
1971	Intel 8080
1972	Intel 8085
1974	Intel 8088
1979	Intel 80286
1982	Intel 286
1985	Intel 386
1989	Intel 486
1993	Intel Pentium
1997	Intel Pentium Pro
1999	Intel Pentium 4
2000	Intel Pentium 4
2006	Intel Core 2 Duo
2008	Intel Core 2 Duo

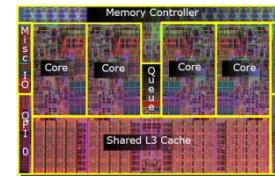
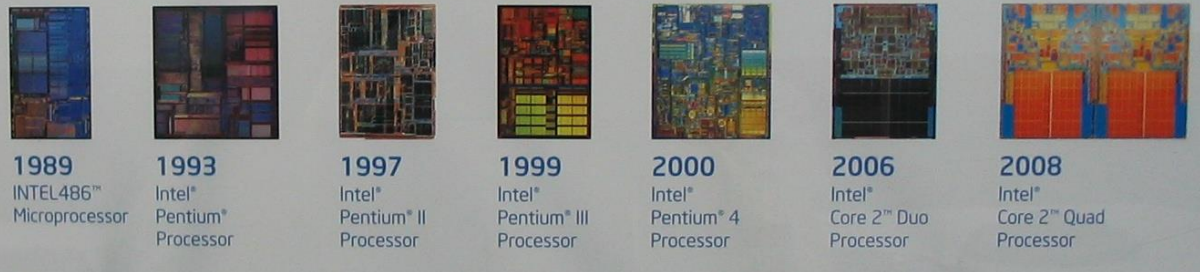
x86 History (Cont.)



32-bit (i586)

32-bit (i686)

64-bit (x86_64)

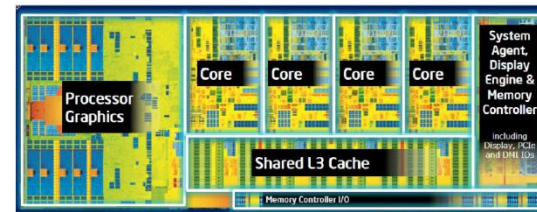
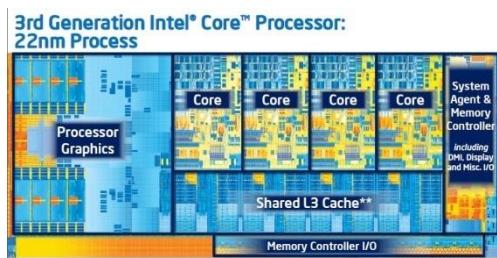


2009
1st Gen. Core i7 (Nehalem)



2011
2nd Gen. Core i7 (Sandy Bridge)

2012
3rd Gen. Core i7 (Ivy Bridge)



2013
4th Gen. Core i7 (Haswell)

x86?

- What is x86?
 - Generic term referring to processors from Intel, AMD and VIA
 - Derived from the model numbers of the first few generations of processors:
 - 80**86**, 802**86**, 803**86**, 804**86** → **x86**
 - Now it generally refers to processors from Intel, AMD, and VIA
 - x86-16: 16-bit processor
 - x86-32 (aka IA32): 32-bit processor * IA: Intel Architecture
 - x86-64: 64-bit processor
- Intel takes about 80% of the PC market and AMD takes about 20%
 - Apple also have been introducing Intel-based Mac from Nov. 2006



Power under the hood.

Inside the MacBook is a powerful 2.13GHz Intel Core 2 Duo processor based on advanced Core microarchitecture. MacBook provides a fast 1066MHz frontside bus and 3MB of shared L2 cache, so you'll have more than enough horsepower to get the job done.



Chipset

- We call North and South Bridges as Chipset
- Chipset has many PCIe devices inside
- North Bridge
 - Memory controller
 - PCI express ports to connect Graphics card
 - <http://www.intel.com/Assets/PDF/datasheet/316966.pdf>
- South Bridge
 - HDD (Hard-disk) controller
 - USB controller
 - Various peripherals connected
 - Keyboard, mouse, timer etc
 - PCI express ports
 - <http://www.intel.com/Assets/PDF/datasheet/316972.pdf>
- **Note that the landscape is being (already) changed!**
 - For example, memory controller is integrated into CPU



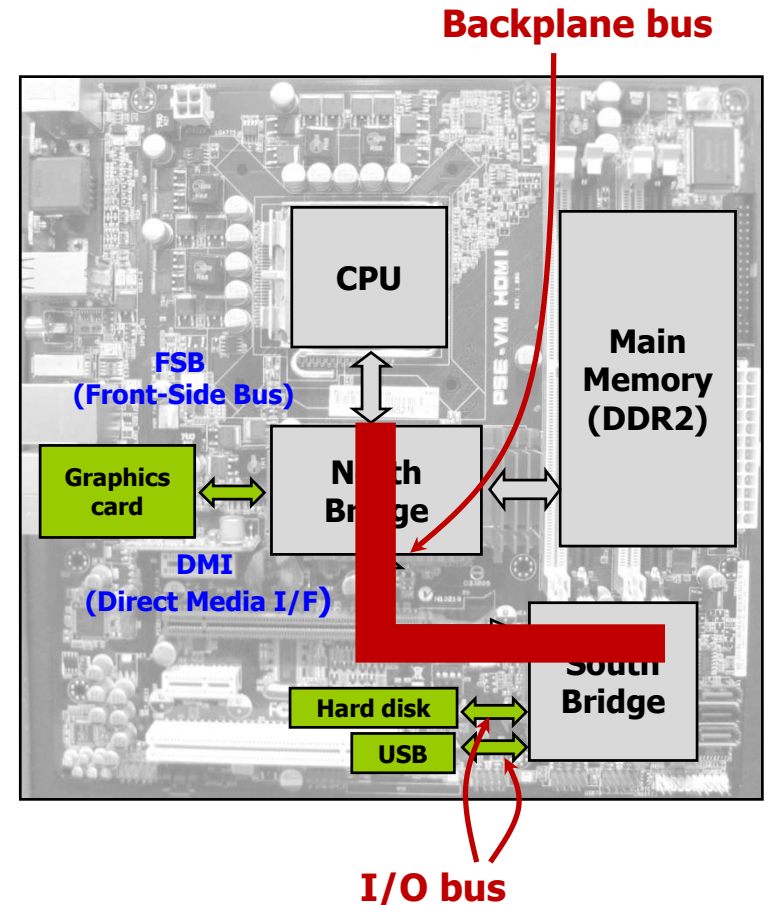
Types of Buses

- **Backplane (backbone) bus**

- Industry standard
 - e.g., **PCIexpress**
- Allow processor, memory and I/O devices to coexist on a single bus
- Used as an intermediary bus connecting I/O busses to the processor-memory bus

- **I/O bus**

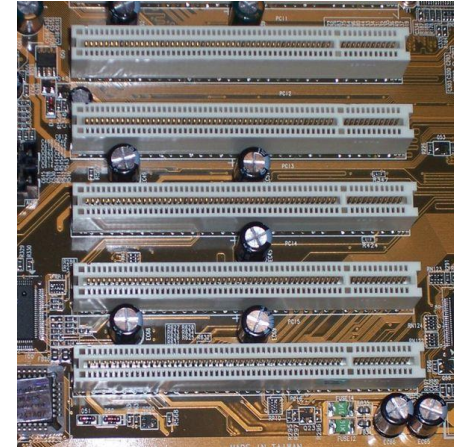
- Industry standard
 - e.g., SATA, USB, Firewire
- Usually is lengthy and slower
- Needs to accommodate a wide range of I/O devices



PCI, PCI Express Devices

- PCI (Peripheral Component Interconnect)
 - Computer bus connecting all the peripheral devices to the computer motherboard
- PCIe (PCI Express)
 - Replaced PCI in 2004
 - Point-to-point connection
- PCIe 2.0
 - Introduced in 2007
- PCIe 3.0
 - Introduced in 2010

PCI slot



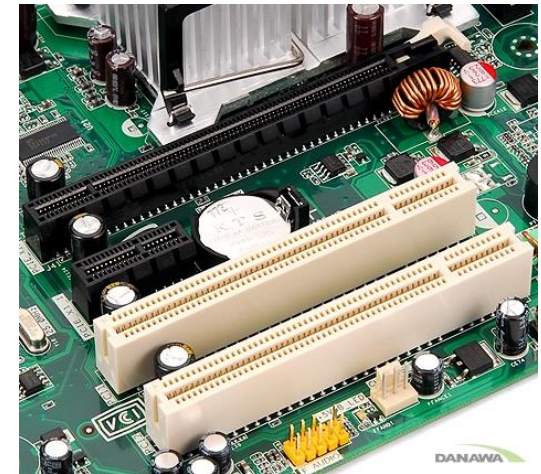
PCI express slots

PCI Express Example Connectors	
x1	BANDWIDTH Single direction: 2.5 Gbps/200 MBps Dual Directions: 5 Gbps/400 MBps
x4	BANDWIDTH Single direction: 10 Gbps/800 MBps Dual Directions: 20 Gbps/1.6 GBps
x8	BANDWIDTH Single direction: 20 Gbps/1.6 GBps Dual Directions: 40 Gbps/3.2 GBps
x16	BANDWIDTH Single direction: 40 Gbps/3.2 GBps Dual Directions: 80 Gbps/6.4 GBps

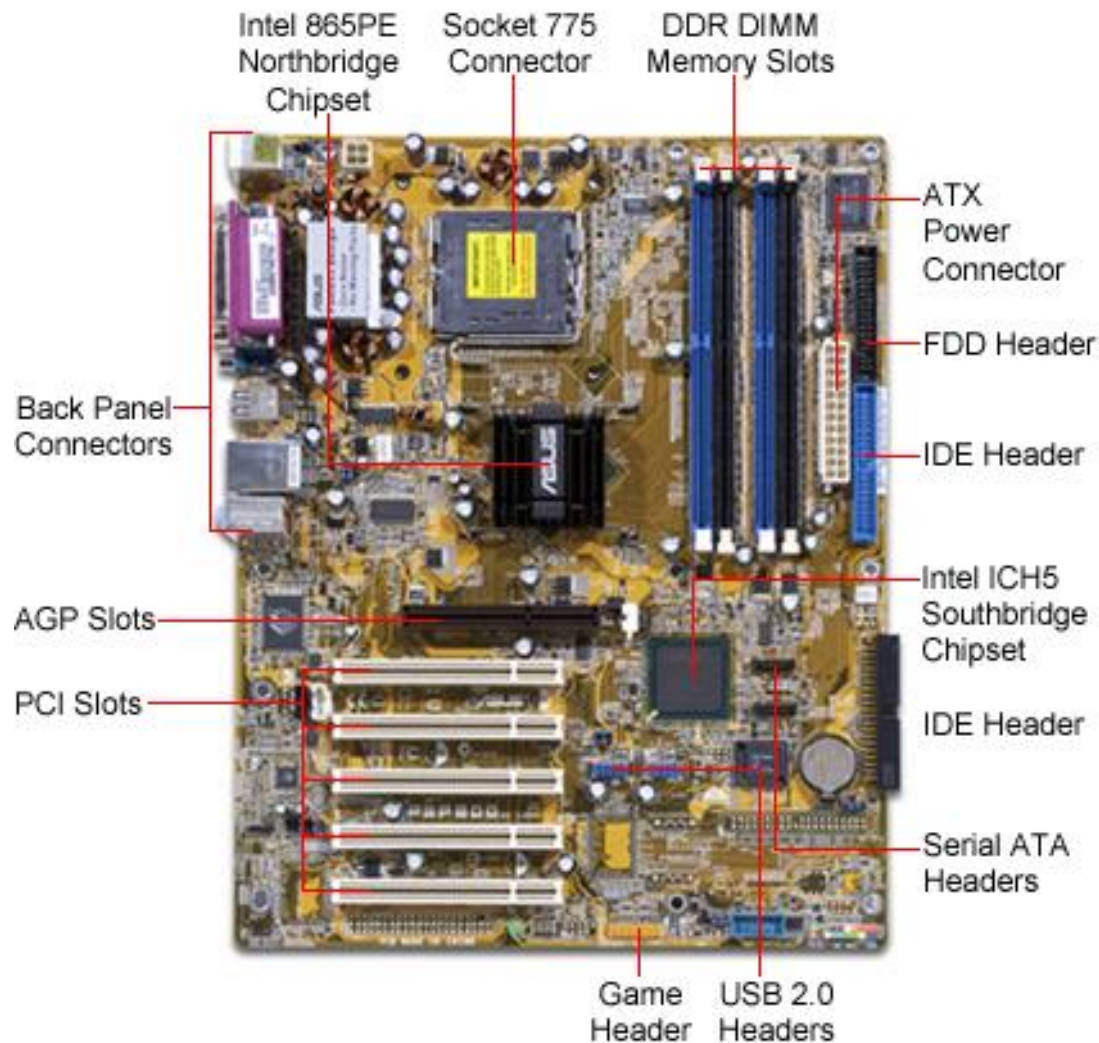
Source: IBM

©2005 HowStuffWorks

PCI express slot x16

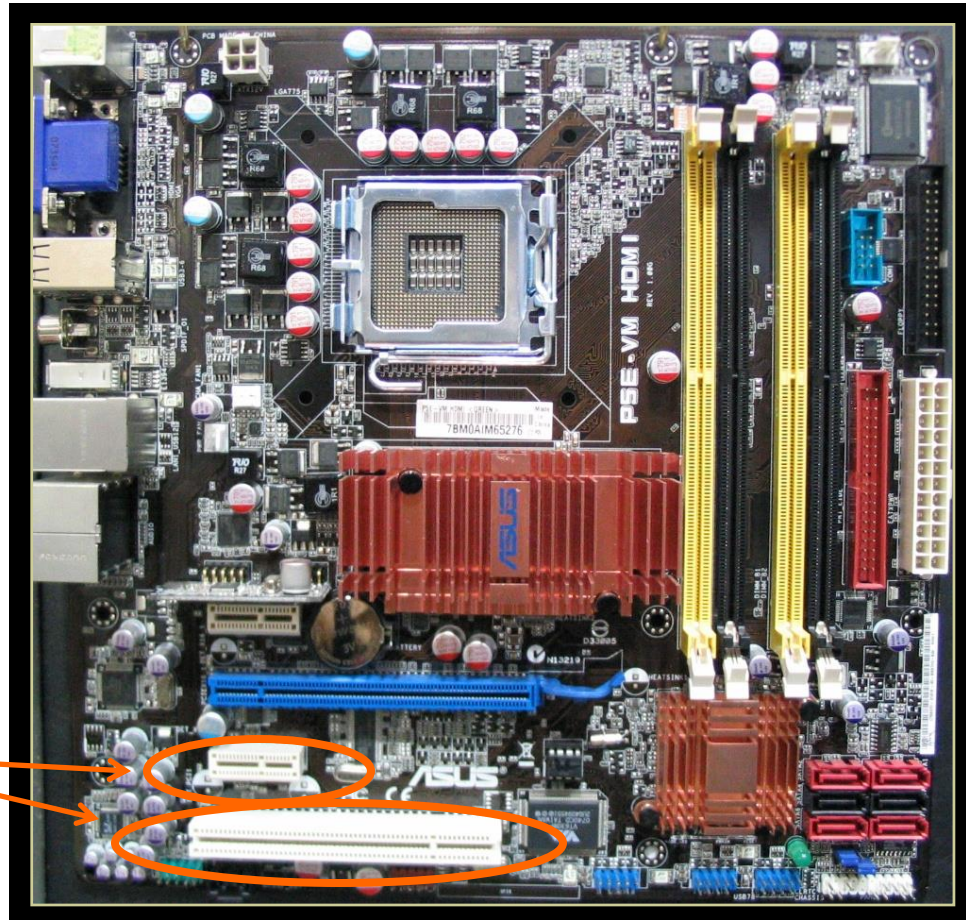


An Old GP Computer System Example

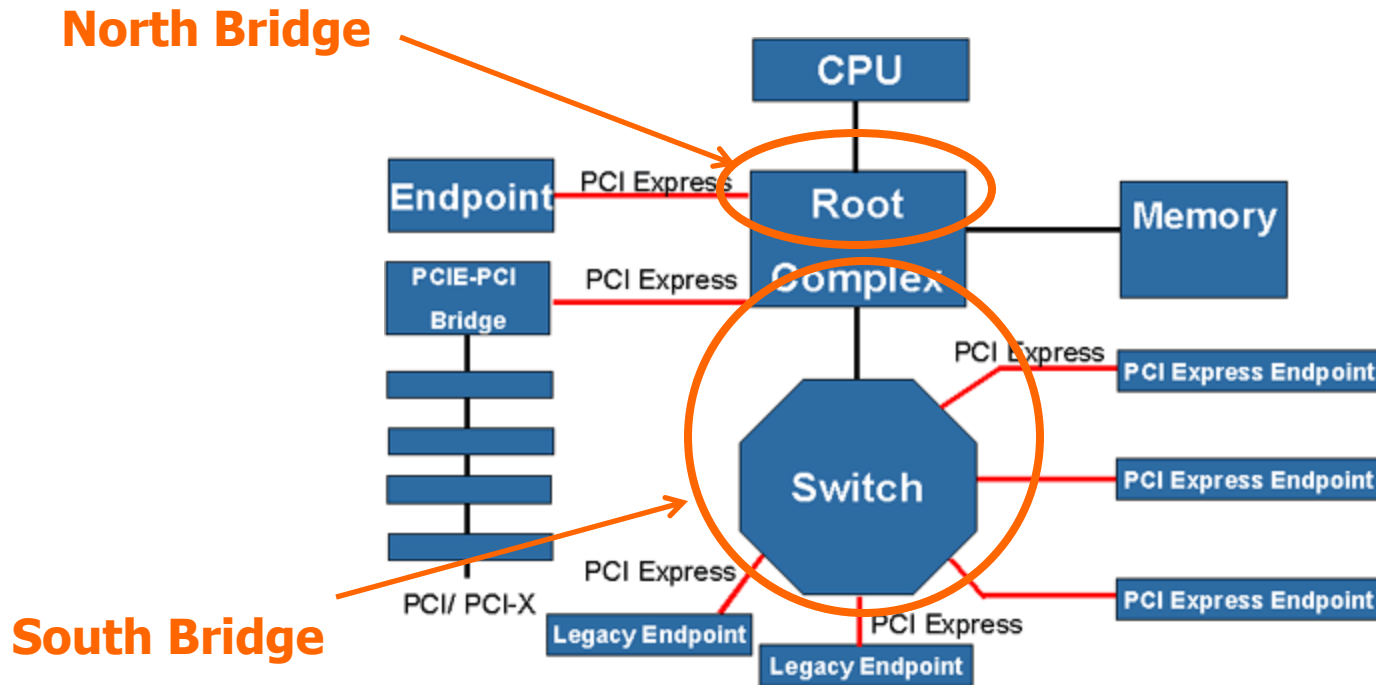


PCI Express Slots in GP Systems

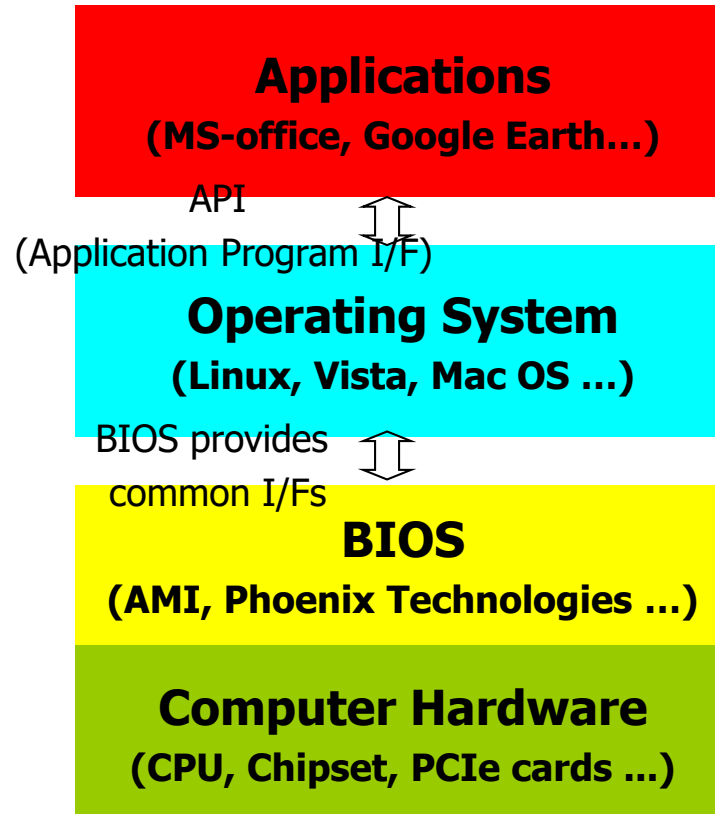
PCI express slot



GP Computer System in terms of PCIe



Software Stack



How the GP Computer System Works?

- x86-based system starts to execute from the **reset address 0xFFFF_FFF0**
 - The first instruction is "**jmp xxx**" off from BIOS ROM
- BIOS (Basic Input/Output System)
 - Detect and initialize all the devices (including PCI devices via PCI enumeration) on the system
 - Provide common interfaces to OS
 - Hand over the control to OS
- OS
 - Manage the system resources such as main memory
 - Control and coordinate the use of the hardware among various application programs for the various users
 - Provide APIs for system and application programming

So... What?

- How is it different from embedded systems?
 - General-purpose computer systems provide **programmability** to end-users
 - You can do any kinds of programming on your PC
 - C, C++, C#, Java etc
 - General-purpose systems should provide **backward compatibility**
 - A new system should be able to run legacy software, which could be in the form of binaries with no source codes written 30 years ago
 - So, general purpose computer system becomes messy and complicated, still containing all legacy hardware functionalities

x86 Operation Modes

- **Real Mode** (= real address mode)
 - Programming environment of the 8086 processor
 - 8086 is a 16-bit processor from Intel
- **Protected Mode**
 - Native state of the 32-bit Intel processor
 - For example, Windows is running in protected mode if 32-bit Windows is installed on your PC
 - 32-bit mode
- **IA-32e mode (IA-32 Extended Mode)**
 - There are 2 sub modes
 - Compatibility mode
 - 64-bit mode



Registers in 8086

- Registers inside the 8086
 - 16-bit segment registers
 - CS, DS, SS, ES
 - General-purpose registers
 - all 16-bits
 - AX, BX, CX, DX, SP, BP, SI, DI

```

MS-DOS Prompt - debug hello.exe
192B:0040  F7 86 E1 E8 15 E3 75 21-80 CF 04 80 3E 8A 99 00  .....u!....>...
192B:0050  74 05 F6 C7 02 75 48 89-3E 4E 99 FF 06 4E 99 C6  t....uH.>N...N..
192B:0060  06 50 99 FF C6 06 51 99-00 E8 99 00 AC E8 61 E2  .P....Q.....a.
192B:0070  74 38 3C 0D 74 34 3A 06-1E 96 74 2E 3A C3 74 2A  t8<.t4:...t.:.t*
-t
AX=192B  BX=0000  CX=0020  DX=0000  SP=0100  BP=0000  SI=0000  DI=0000
DS=191A  ES=191A  SS=192C  CS=192A  IP=0003  NU UP EI PL NZ NA PO NC
192A:0003  8ED8                MOV     DS,AX
-t
AX=192B  BX=0000  CX=0020  DX=0000  SP=0100  BP=0000  SI=0000  DI=0000
DS=192B  ES=191A  SS=192C  CS=192A  IP=0005  NU UP EI PL NZ NA PO NC
192A:0005  BA0200              MOV     DX,0002
-t
AX=192B  BX=0000  CX=0020  DX=0002  SP=0100  BP=0000  SI=0000  DI=0000
DS=192B  ES=191A  SS=192C  CS=192A  IP=0008  NU UP EI PL NZ NA PO NC
192A:0008  B409                MOV     AH,09
-t
AX=092B  BX=0000  CX=0020  DX=0002  SP=0100  BP=0000  SI=0000  DI=0000
DS=192B  ES=191A  SS=192C  CS=192A  IP=000A  NU UP EI PL NZ NA PO NC
192A:000A  CD21                INT     21

```

- Registers in x86-32



General-Purpose Registers							
31	16	15	8	7	0	16-bit	32-bit
	AH		AL			AX	EAX
	BH		BL			BX	EBX
	CH		CL			CX	ECX
	DH		DL			DX	EDX
	BP						EBP
	SI						ESI
	DI						EDI
	SP						ESP

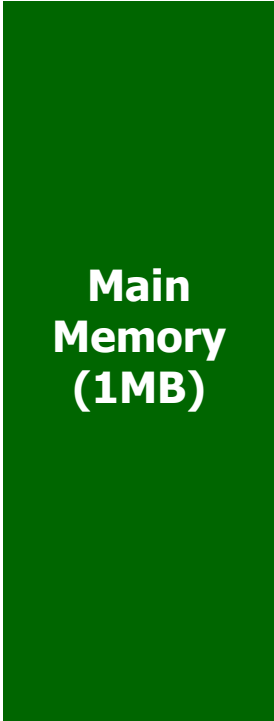
Figure 3-5. Alternate General-Purpose Register Names

Real Mode Addressing

- In real mode (8086), general purpose registers are all 16-bit wide
- Real model
 - Segment registers specify the base address of each segment
 - Segment registers
 - CS: Code Segment -> used to access instructions
 - DS: Data Segment -> used to store data
 - SS: Stack Segment -> Stack
 - ES: Extra Segment -> could be used to store more data
 - Addressing method
 - Segment $\ll 4$ + offset = physical address
 - Example:

```
mov ax, 2000h
mov ds, ax
```

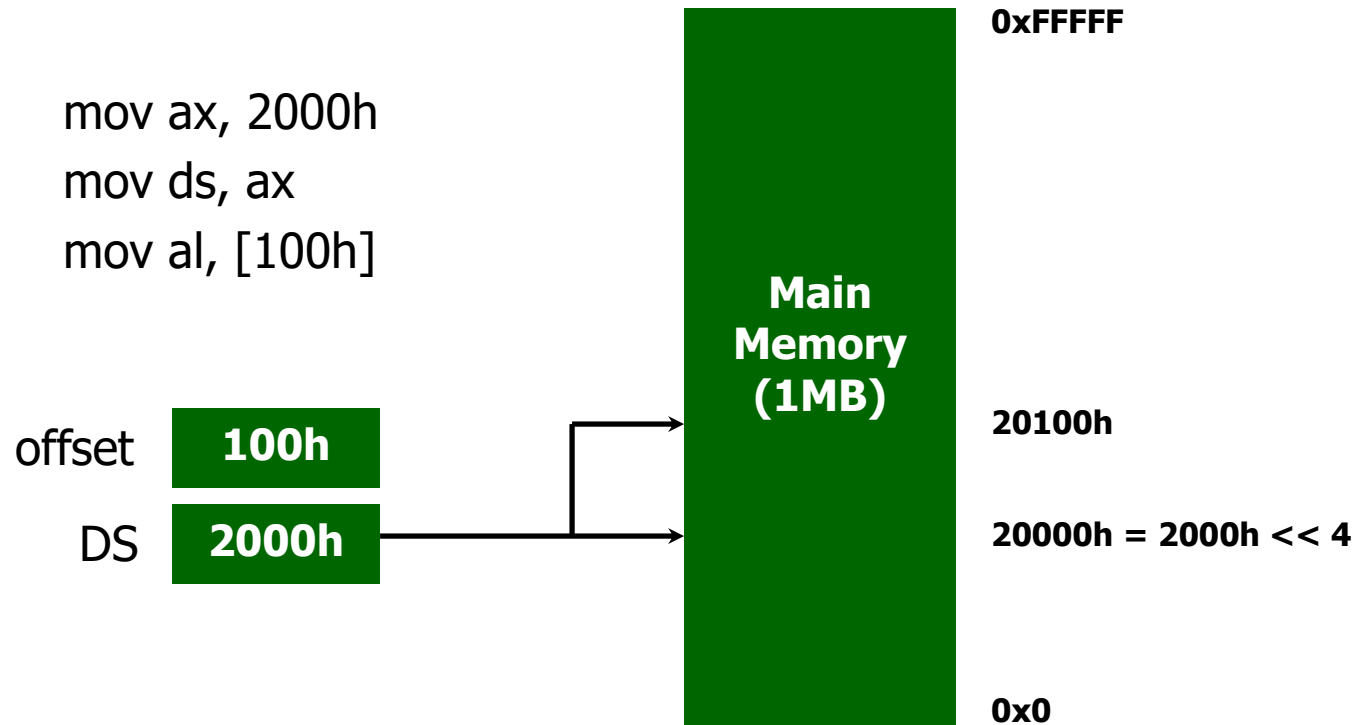
→ Data segment starts from 20000h ($2000h \ll 4$)



**Main
Memory
(1MB)**

Data Segment in Real Mode

- Memory addressing in real mode (8086)



A20M

- 8088/8086 allows only 1MB memory access since they have only 20-bit physical address lines
 - $2^{20} = 1\text{MB}$
- Memory is accessed with **segment:offset** in 8086/8088 (still the same though)
 - What if CS=0xFFFF, IP=0x0020?
 - $\text{CS} \ll 4 + \text{IP} = 0x100010$
 - But, we have only 20 address lines. So, 8088 ends up accessing 0x00010 ignoring the "1" in A20
- Some (weird?) programmers took advantage of this mechanism

A20M (Cont)

- How about now?
 - Your Core 2 Duo has 48-bit physical address lines
 - What happens if there is no protection in the previous case
 - Processor will access 0x100010, breaking the legacy code
 - So, x86 provides a mechanism called **A20M (A20 Mask)** to make it compatible with the old generations

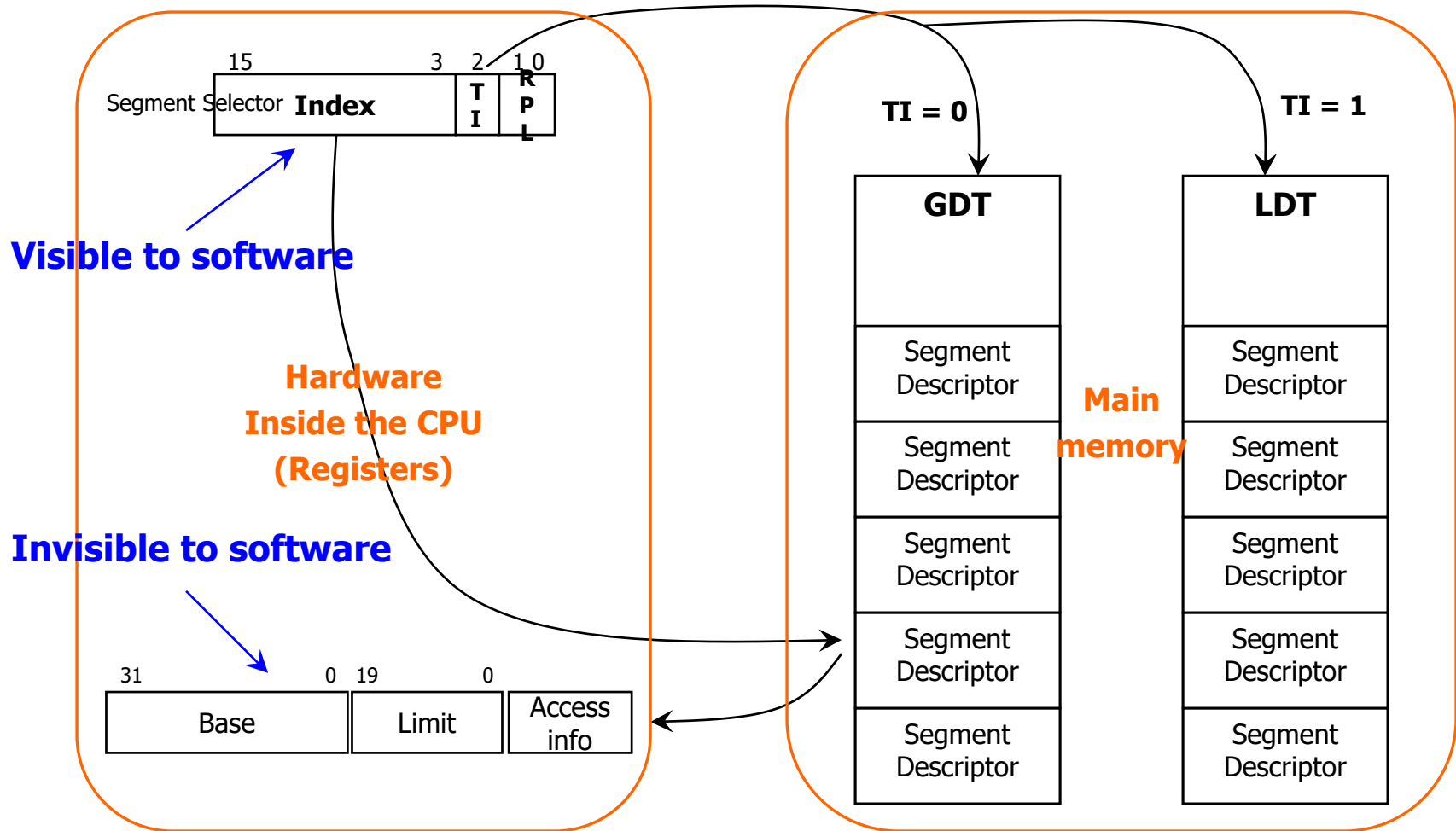
Korea Univ



Another Example

- Protected mode addressing (32-bit)
 - As application programs become larger, 1MB main memory is too small
 - Intel introduced protected mode to address a larger memory (up to 4GB)
 - But, Intel still wants to use 16-bit segment registers for the backward compatibility
 - How to access a 4GB space with a 16-bit register?

Protected Mode Addressing



Segment Descriptor Format

- Software (OS) creates descriptor tables (GDT, LDT)

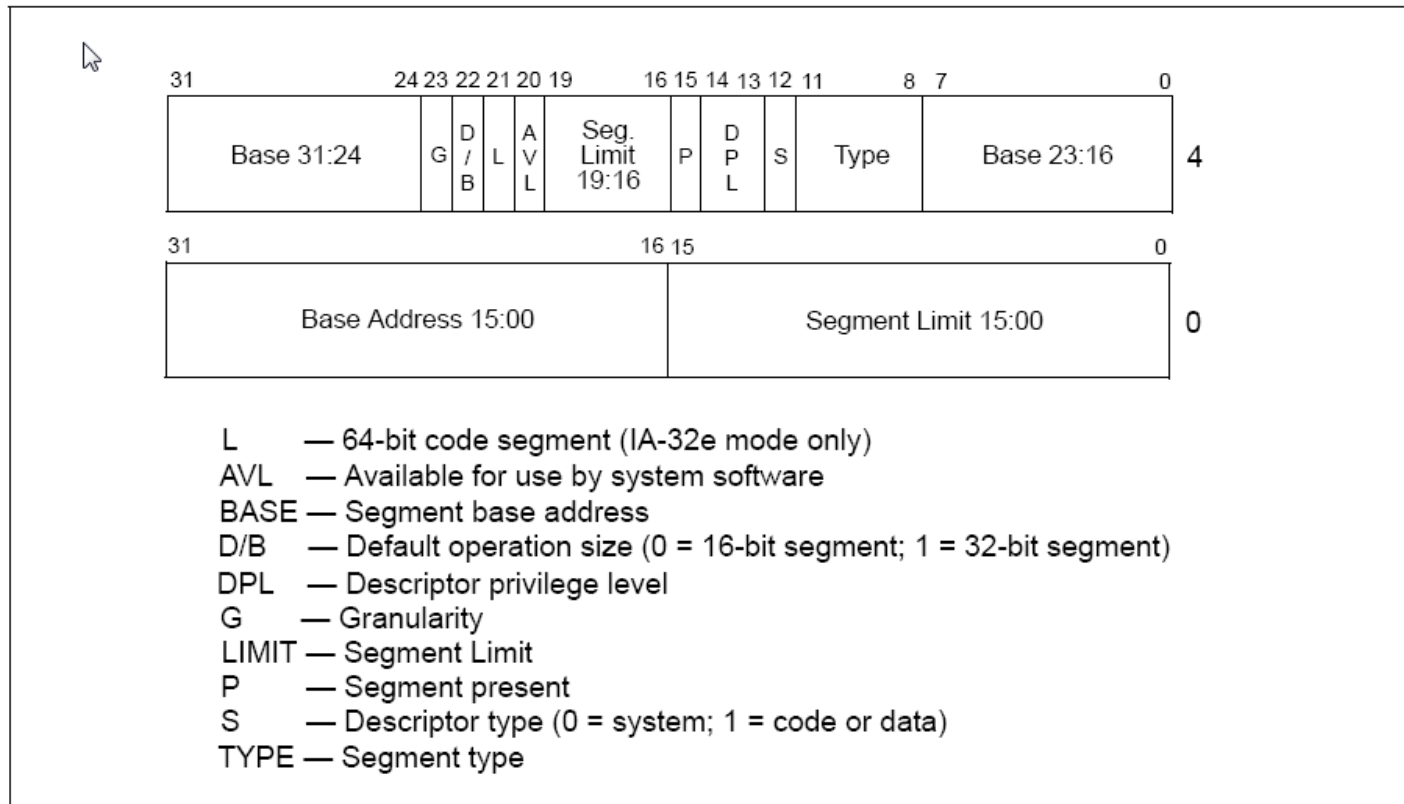


Figure 3-8. Segment Descriptor

Address Translation in Protected Mode

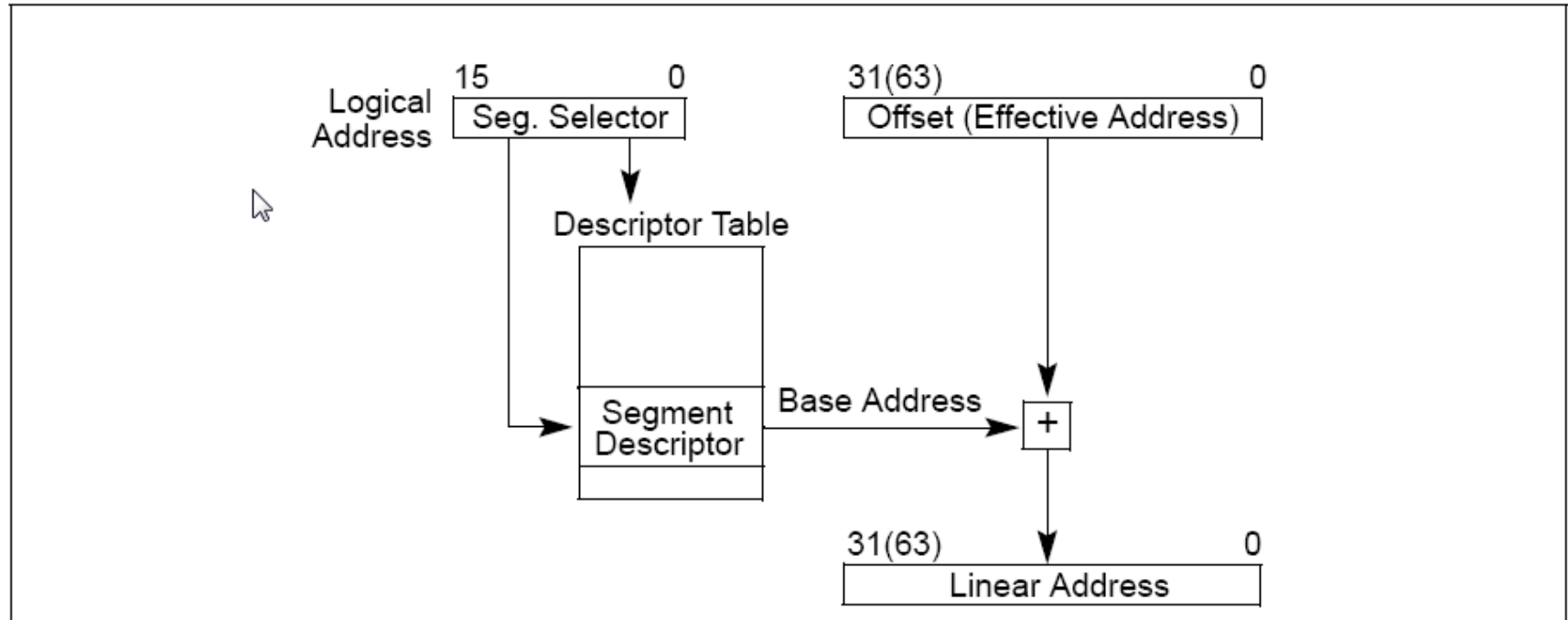
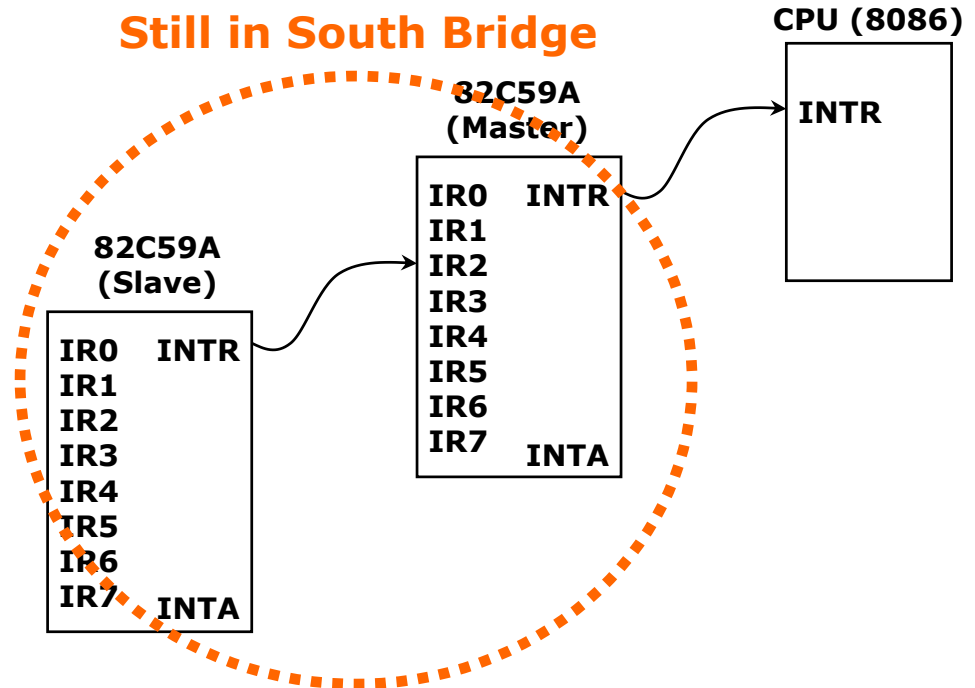
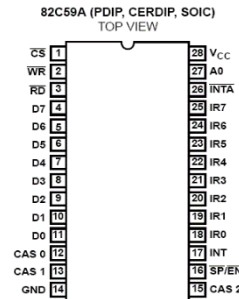
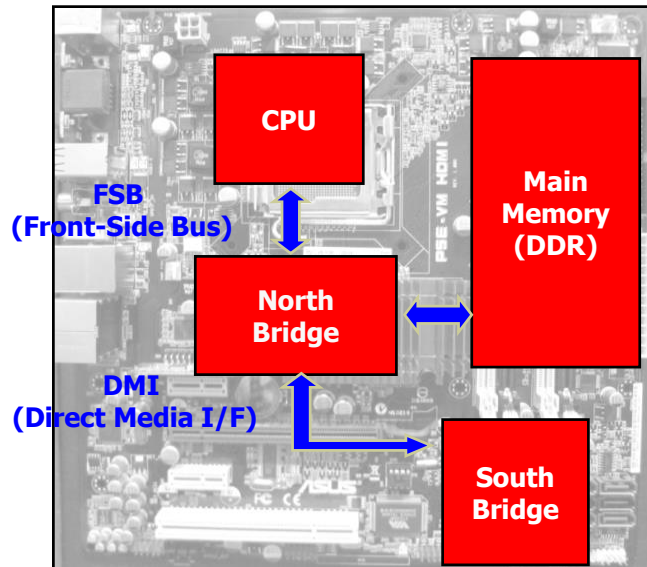


Figure 3-5. Logical Address to Linear Address Translation

One More Example

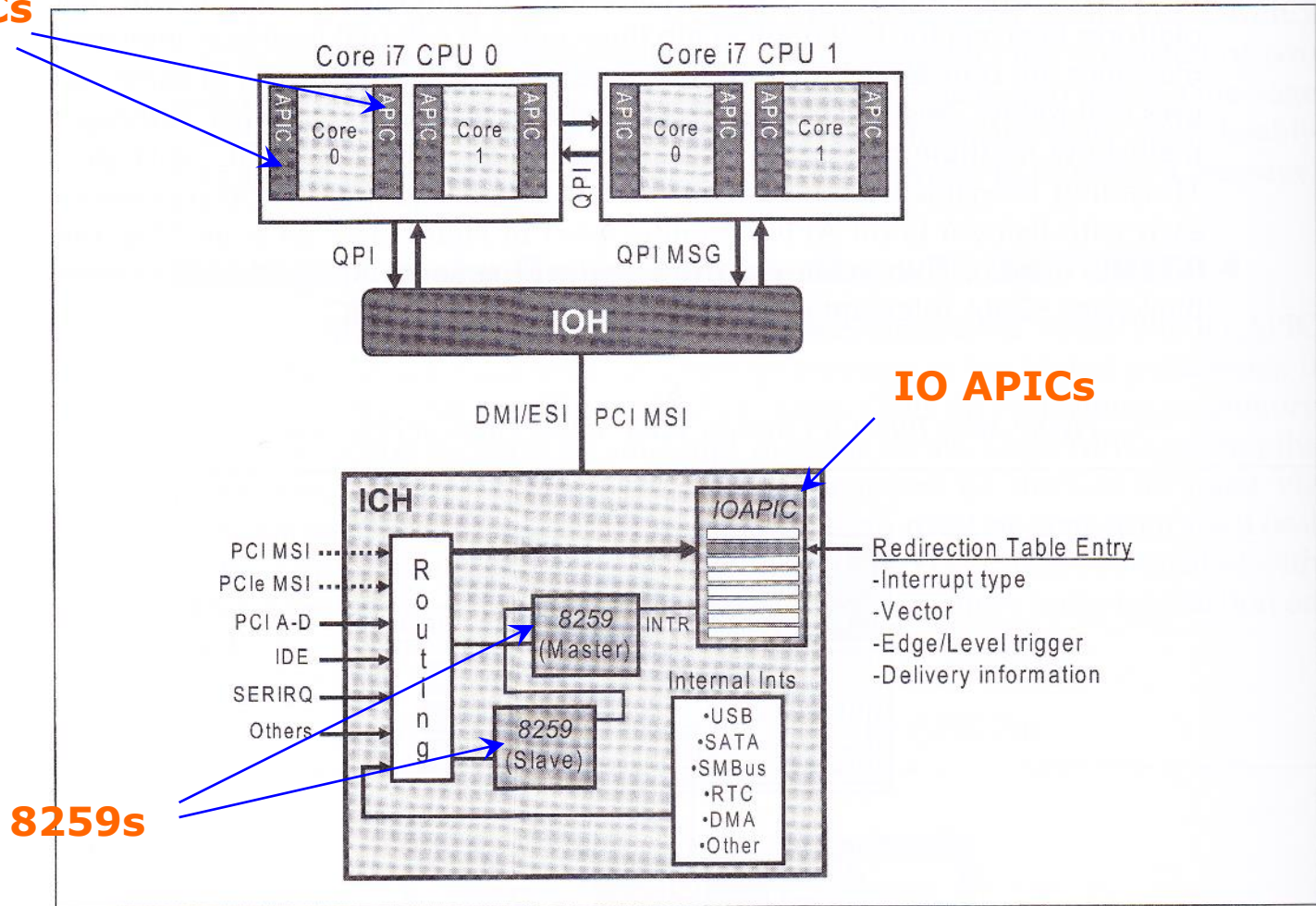
- 8259 Interrupt Controller



APICs and 8259s

Figure 34-7: Intel QPI System With Dual Physical Processors

Local APICs



IO APICs

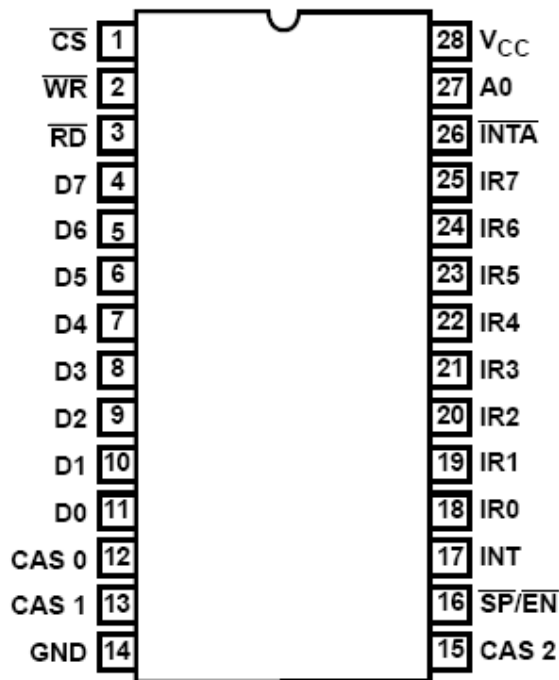
APIC: Advanced Programmable Interrupt Controller

Backup Slides

8259 in Prehistoric Era

82C59A (PDIP, Cerdip, SOIC)

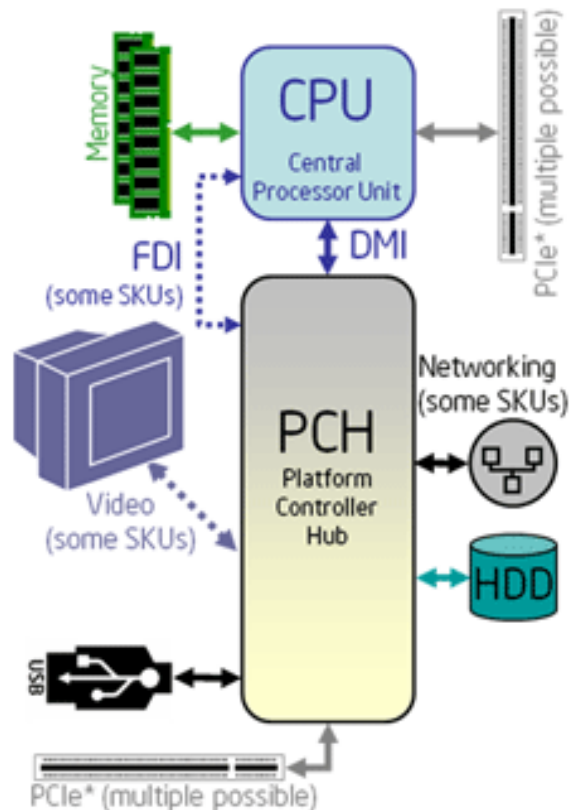
TOP VIEW



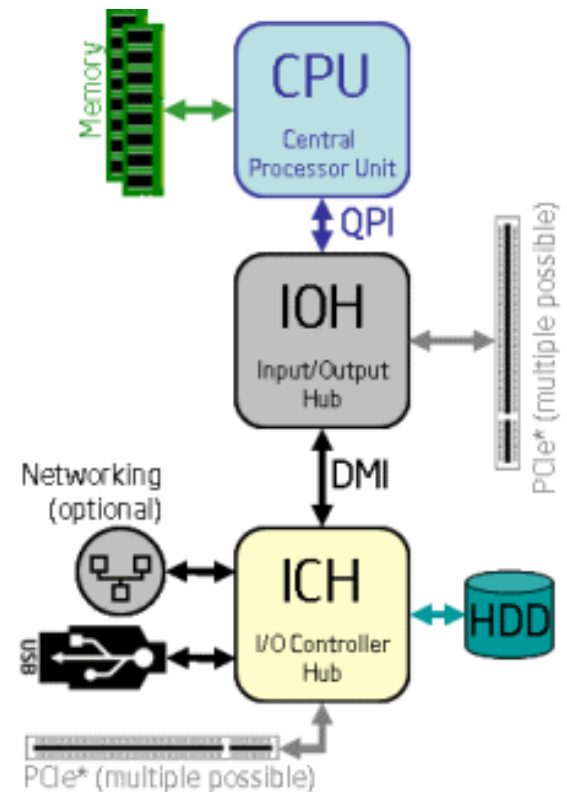
PIN	DESCRIPTION
D7 - D0	Data Bus (Bidirectional)
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A0	Command Select Address
$\overline{CS}$	Chip Select
CAS 2 - CAS 0	Cascade Lines
$\overline{SP/EN}$	Slave Program Input Enable
INT	Interrupt Output
$\overline{INTA}$	Interrupt Acknowledge Input
IR0 - IR7	Interrupt Request Inputs

Core i7-based Systems

- Core i7 860 (Lynnfield)
– based system

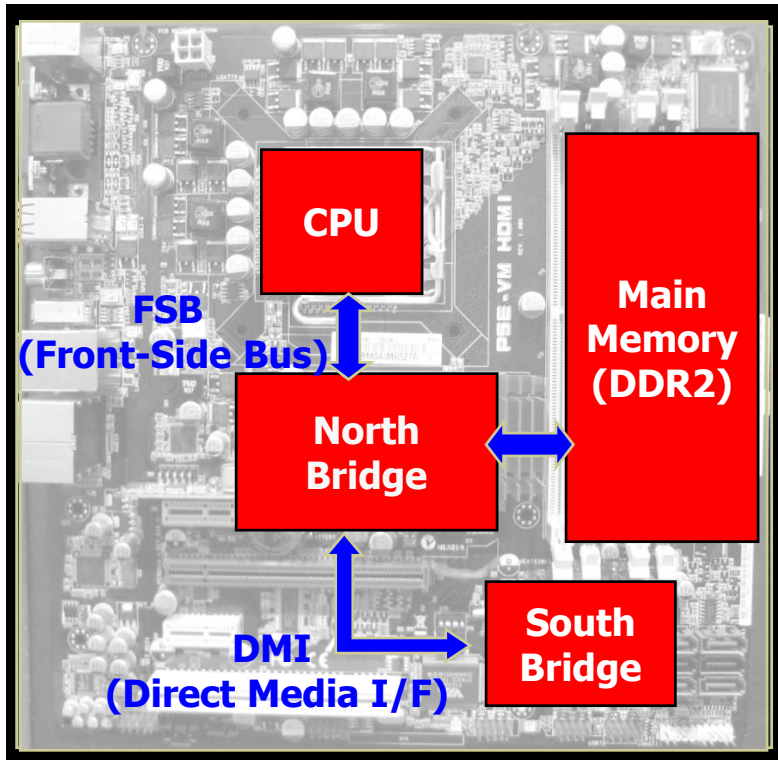


- Core i7 920 (Bloomfield)
– based system

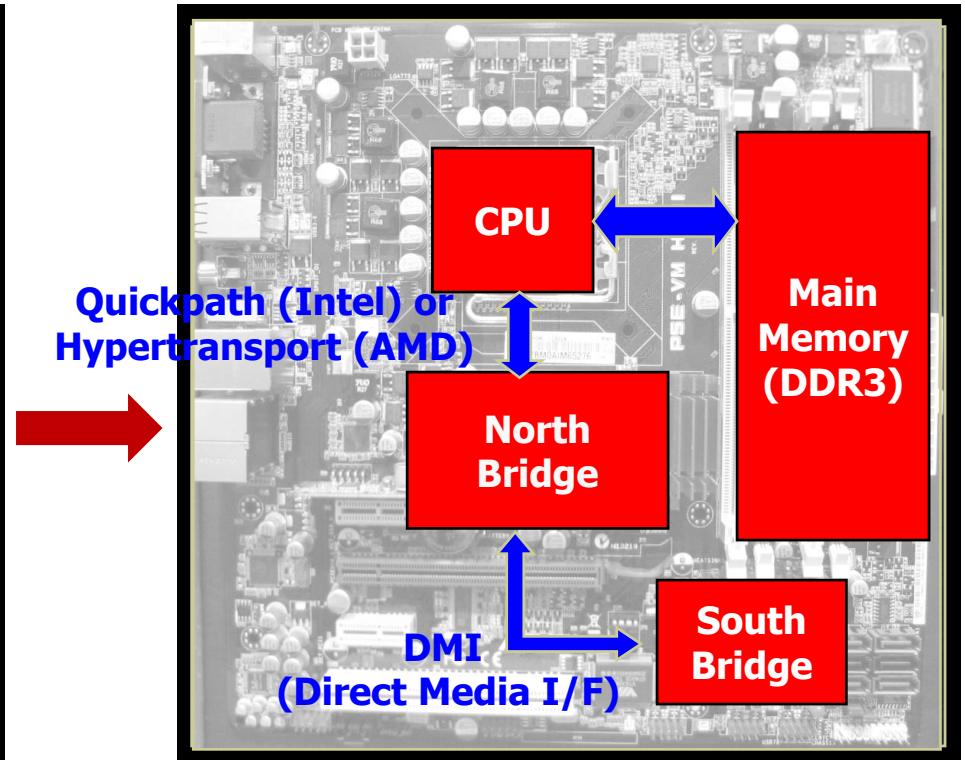


Present and More...

- Core 2 Duo – based Systems



- Core i7– based Systems



Keep in mind that CPU and computer systems are evolving at a fast pace