AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Procesory i Architektura Systemów Komputerowych

Virtual Machines & Virtual Memory

IET
Katedra Elektroniki
Kraków 2015
dr inż. Roman Rumian

System virtual machines present the illusion that the users have an entire computer to themselves, including a copy of the operating system. A single computer runs multiple VMs and can support a number of different ***operating systems*** (OSes). On a conventional platform, a single OS "owns" all the hardware resources, but with a VM, multiple OSes all share the hardware resources.

The soft ware that supports VMs is called a ***virtual machine monitor*** (VMM) or ***hypervisor****;* the VMM is the heart of virtual machine technology. The underlying hardware platform is called the ***host***, and its resources are shared among the ***guest*** VMs. The VMM determines how to map virtual resources to physical resources: a physical resource may be time-shared, partitioned, or even emulated in soft ware.

The VMM is much smaller than a traditional OS; the isolation portion of a VMM is perhaps only 10,000 lines of code.

VMs provide two other benefits that are commercially significant:

1. ***Managing software***. VMs provide an abstraction that can run the complete soft ware stack, even including old operating systems like DOS. A typical deployment might be some VMs running legacy OSes, many running the current stable OS release, and a few testing the next OS release.
2. ***Managing hardware***. One reason for multiple servers is to have each application running with the compatible version of the operating system on separate computers, as this separation can improve dependability. VMs allow these separate soft ware stacks to run independently yet share hardware, thereby consolidating the number of servers. Another example is that some VMMs support migration of a running VM to a different computer, either to balance load or to evacuate from failing hardware.

AGH

What must a VM monitor do? It presents a software interface to guest software, it must isolate the state of guests from each other, and it must protect itself from guest software (including guest OSes). The qualitative requirements are:

- Guest soft ware should behave on a VM exactly as if it were running on the native hardware, except for performance-related behavior or limitations of fixed resources shared by multiple VMs.
- Guest soft ware should not be able to change allocation of real system resources directly. To "virtualize" the processor, the VMM must control just about everything—access to privileged state, I/O, exceptions, and interrupts—even though the guest VM and OS currently running are temporarily using them.

To be in charge, **the VMM must be at a higher privilege level than the guest VM**, which generally runs in user mode; this also ensures that the execution of any privileged instruction will be handled by the VMM. The basic requirements of system virtual:

- **At least two processor modes, system and user.**
- **A privileged subset of instructions that is available only in system mode**, resulting in a trap if executed in user mode; all system resources must be controllable only via these instructions.

Virtual Memory

A technique that uses main memory as a “cache” for secondary storage.

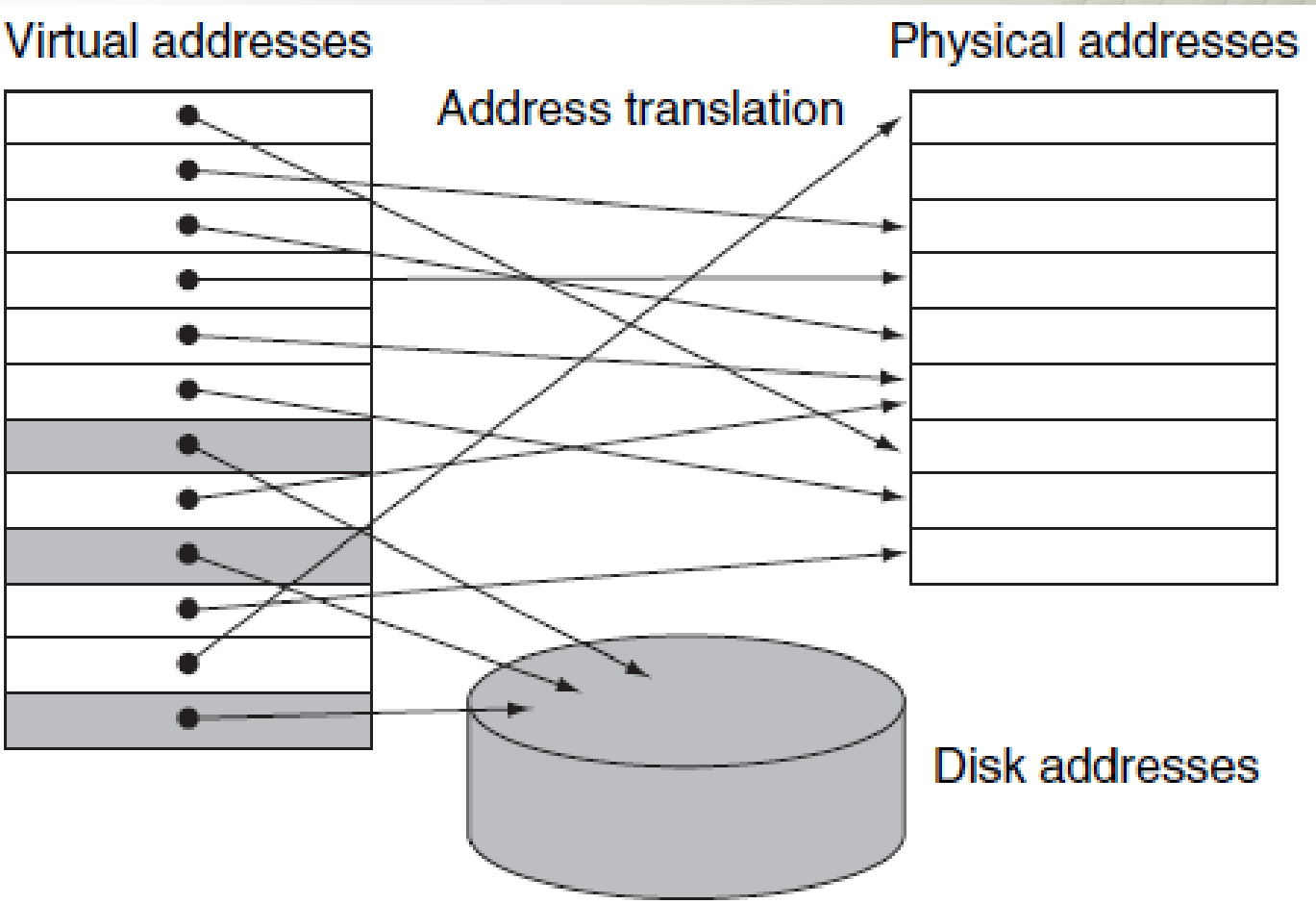
physical address An address in main memory.

protection A set of mechanisms for ensuring that multiple processes sharing the processor, memory, or I/O devices cannot interfere, intentionally or unintentionally, with one another by reading or writing each other’s data.
These mechanisms also isolate the operating system from a user process.

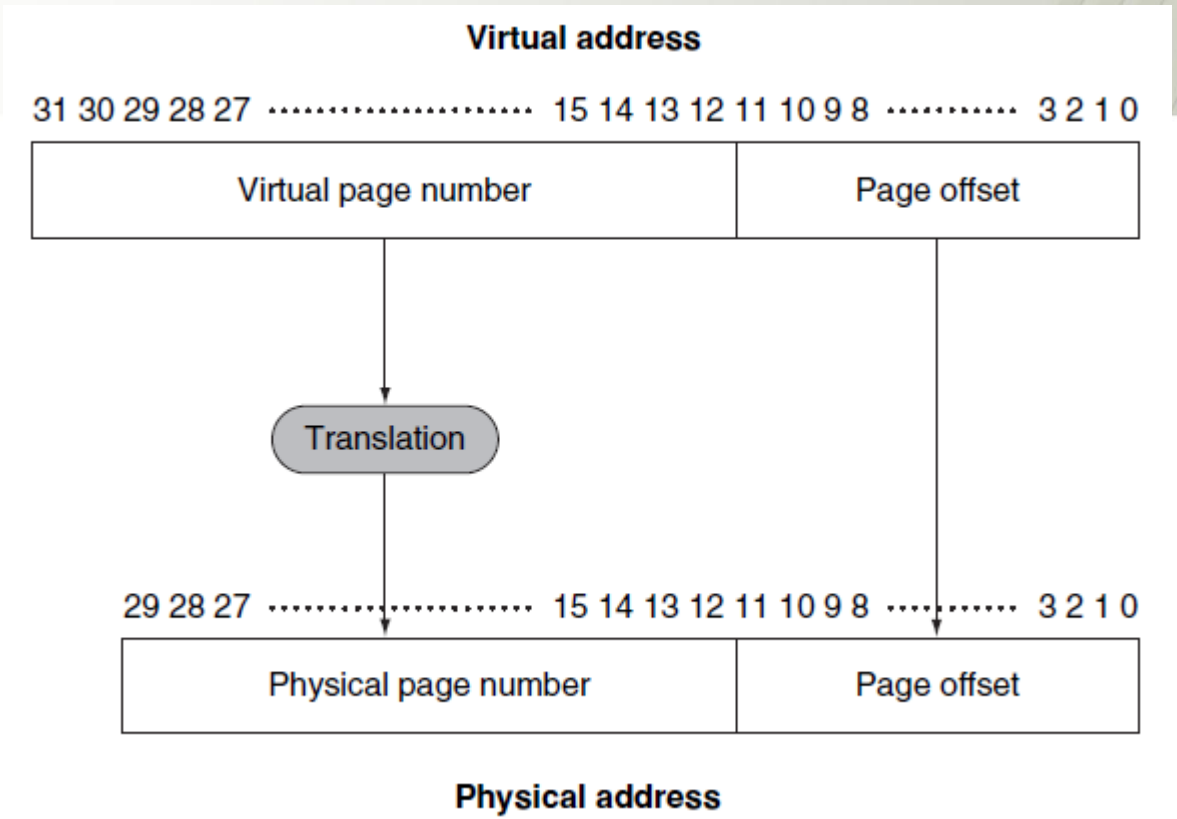
page fault An event that occurs when an accessed page is not present in main memory.

virtual address An address that corresponds to a location in virtual space and is translated by address mapping to a physical address when memory is accessed.

address translation Also called **address mapping**. The process by which a virtual address is mapped to an address used to access memory.



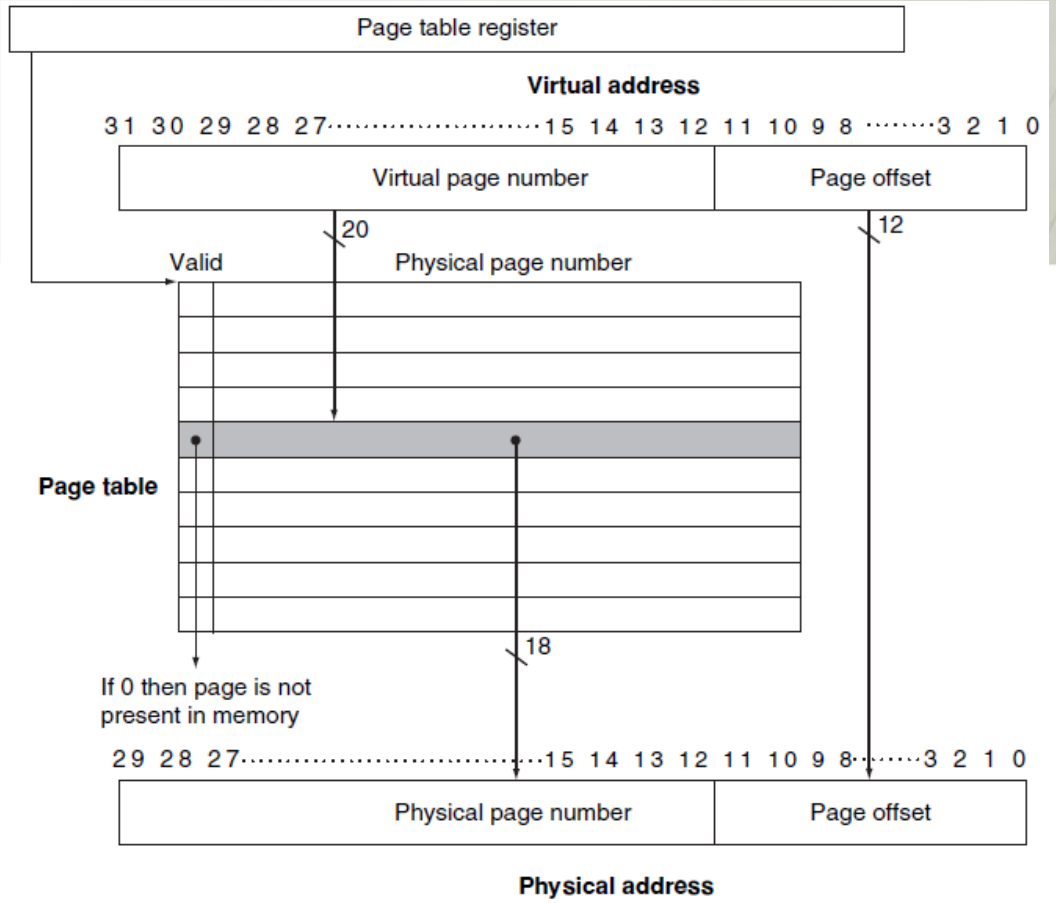
In virtual memory, blocks of memory (called *pages*) are mapped from one set of addresses (called *virtual addresses*) to another set (called *physical addresses*).



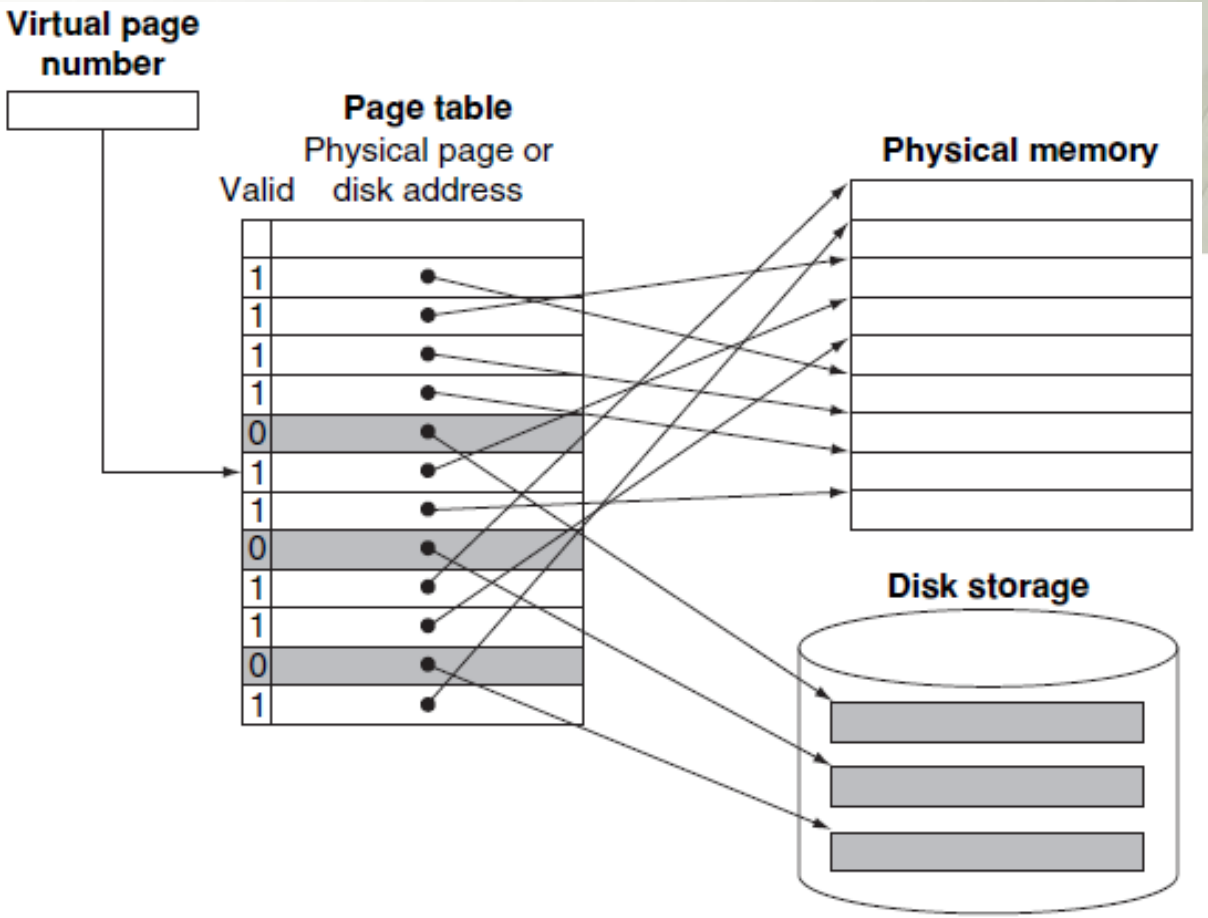
Mapping from a virtual to a physical address. The page size is 212 4 KiB. The number of physical pages allowed in memory is 218, since the physical page number has 18 bits in it. Thus, main memory can have at most 1 GiB, while the virtual address space is 4 GiB.

■ Pages should be large enough to try to amortize the high access time. Sizes from 4 KiB to 16 KiB are typical today. New desktop and server systems are being developed to support 32 KiB and 64 KiB pages, but new embedded systems are going in the other direction, to 1 KiB pages.
■ Organizations that reduce the page fault rate are attractive. The primary technique used here is to allow fully associative placement of pages in memory.
■ Page faults can be handled in software because the overhead will be small compared to the disk access time. In addition, soft ware can afford to use clever algorithms for choosing how to place pages because even small reductions in the miss rate will pay for the cost of such algorithms.
■ Write-through will not work for virtual memory, since writes take too long. Instead, virtual memory systems use write-back.

Segmentation A variable-size address mapping scheme in which an address consists of two parts: a segment number, which is mapped to a physical address, and a segment off set.
page table The table containing the virtual to physical address translations in a virtual memory system. The table, which is stored in memory, is typically indexed by the virtual page number; each entry in the table contains the physical page number for that virtual page if the page is currently in memory.
swap space The space on the disk reserved for the full virtual memory space of a process.
reference bit Also called **use bit**. A field that is set whenever a page is accessed and that is used to implement LRU or other replacement schemes.

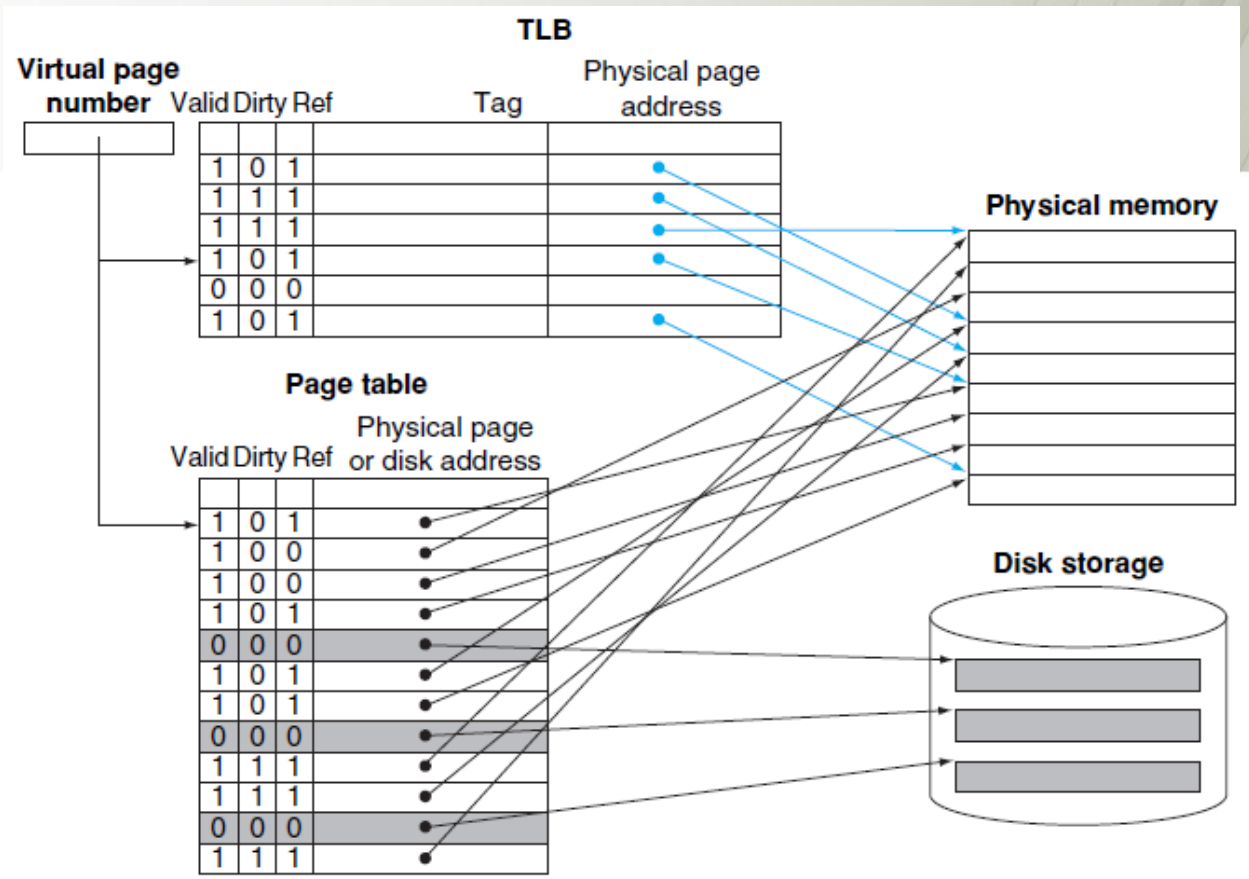


The page table is indexed with the virtual page number to obtain the corresponding portion of the physical address. We assume a 32-bit address. The page table pointer gives the starting address of the page table. In this figure, the page size is 212 bytes, or 4 KiB. The virtual address space is 232 bytes, or 4 GiB, and the physical address space is 230 bytes, which allows main memory of up to 1 GiB. Th e number of entries in the page table is 220, or 1 million entries. The valid bit for each entry indicates whether the mapping is legal. If it is off , then the page is not present in memory. Although the page table entry shown here need only be 19 bits wide, it would typically be rounded up to 32 bits for ease of indexing. Th e extra bits would be used to store additional information that needs to be kept on a per-page basis, such as protection.



The page table maps each page in virtual memory to either a page in main memory or a page stored on disk, which is the next level in the hierarchy. The virtual page number is used to index the page table. If the valid bit is on, the page table supplies the physical page number (i.e., the starting address of the page in memory) corresponding to the virtual page. If the valid bit is off , the page currently resides only on disk, at a specifi ed disk address. In many systems, the table of physical page addresses and disk page addresses, while logically one table, is stored in two separate data structures. Dual tables are justifi ed in part because we must keep the disk addresses of all the pages, even if they are currently in main memory. Remember that the pages in main memory and the pages on disk are the same size.

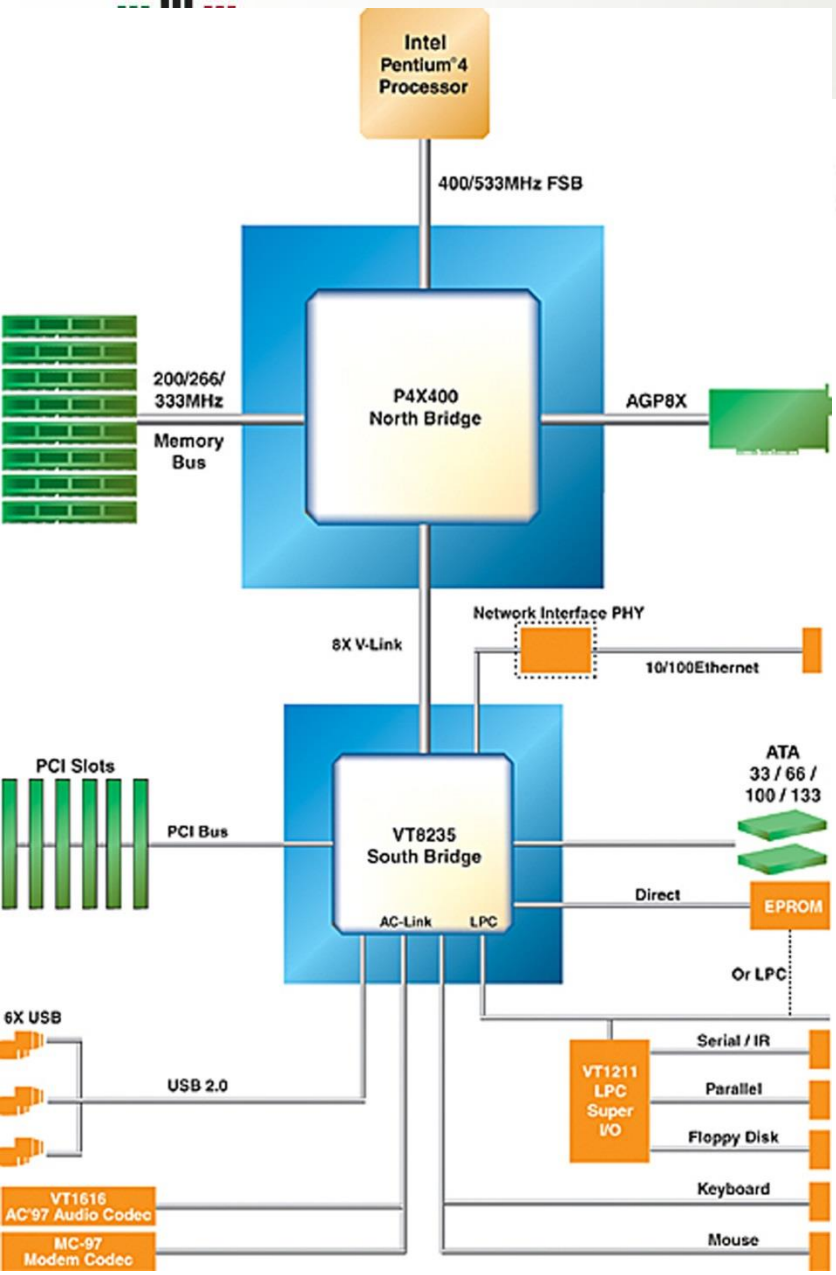
translation-lookaside buffer (TLB) A cache that keeps track of recently used address mappings to try to avoid an access to the page table.

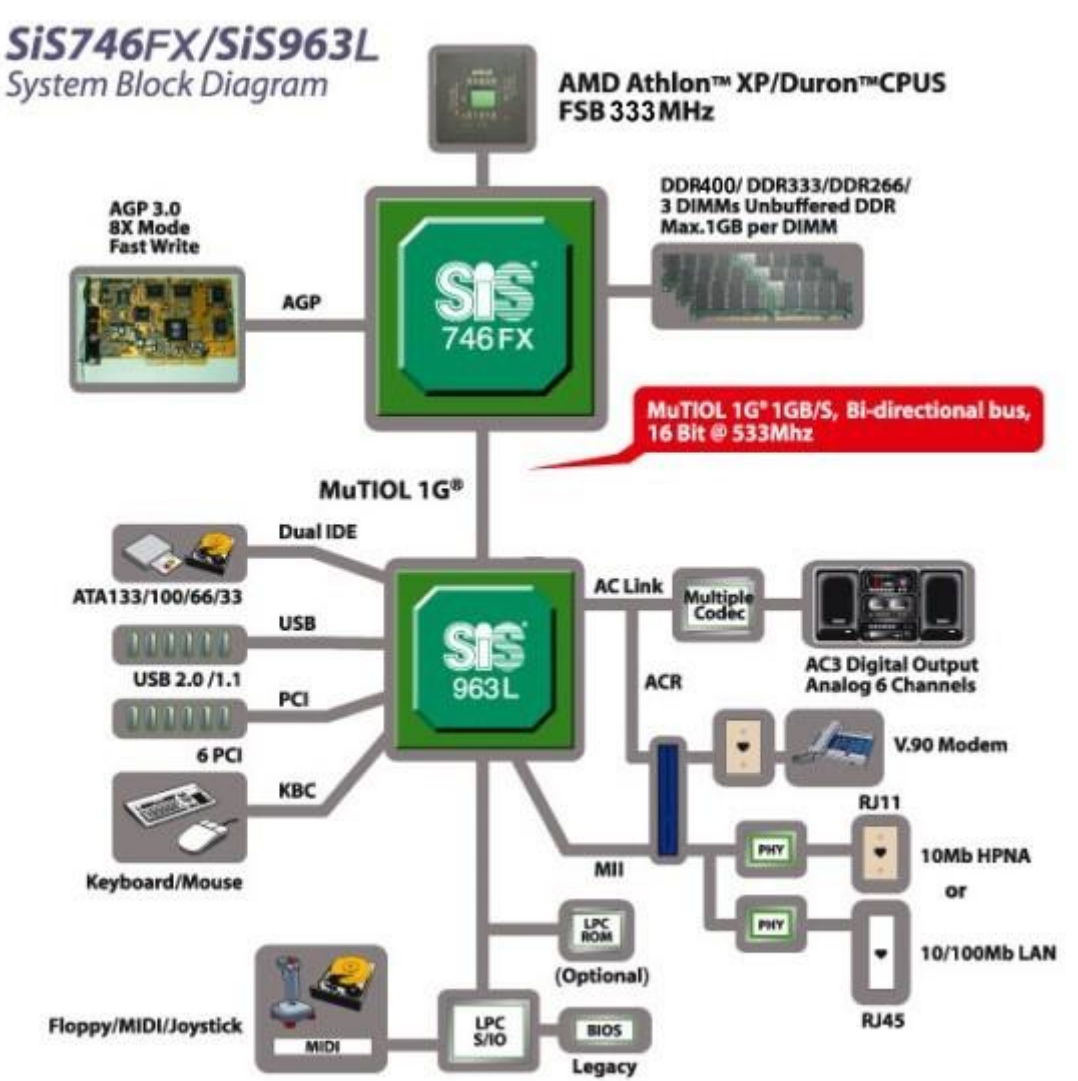


The TLB acts as a cache of the page table for the entries that map to physical pages only. The TLB contains a subset of the virtual-to-physical page mappings that are in the page table. The TLB mappings are shown in color. Because the TLB is a cache, it must have a tag field. If there is no matching entry in the TLB for a page, the page table must be examined. The page table either supplies a physical page number for the page (which can then be used to build a TLB entry) or indicates that the page resides on disk, in which case a page fault occurs. Since the page table has an entry for every virtual page, no tag field is needed; in other words, unlike a TLB, a page table is *not* a cache.

Some typical values for a TLB might be:

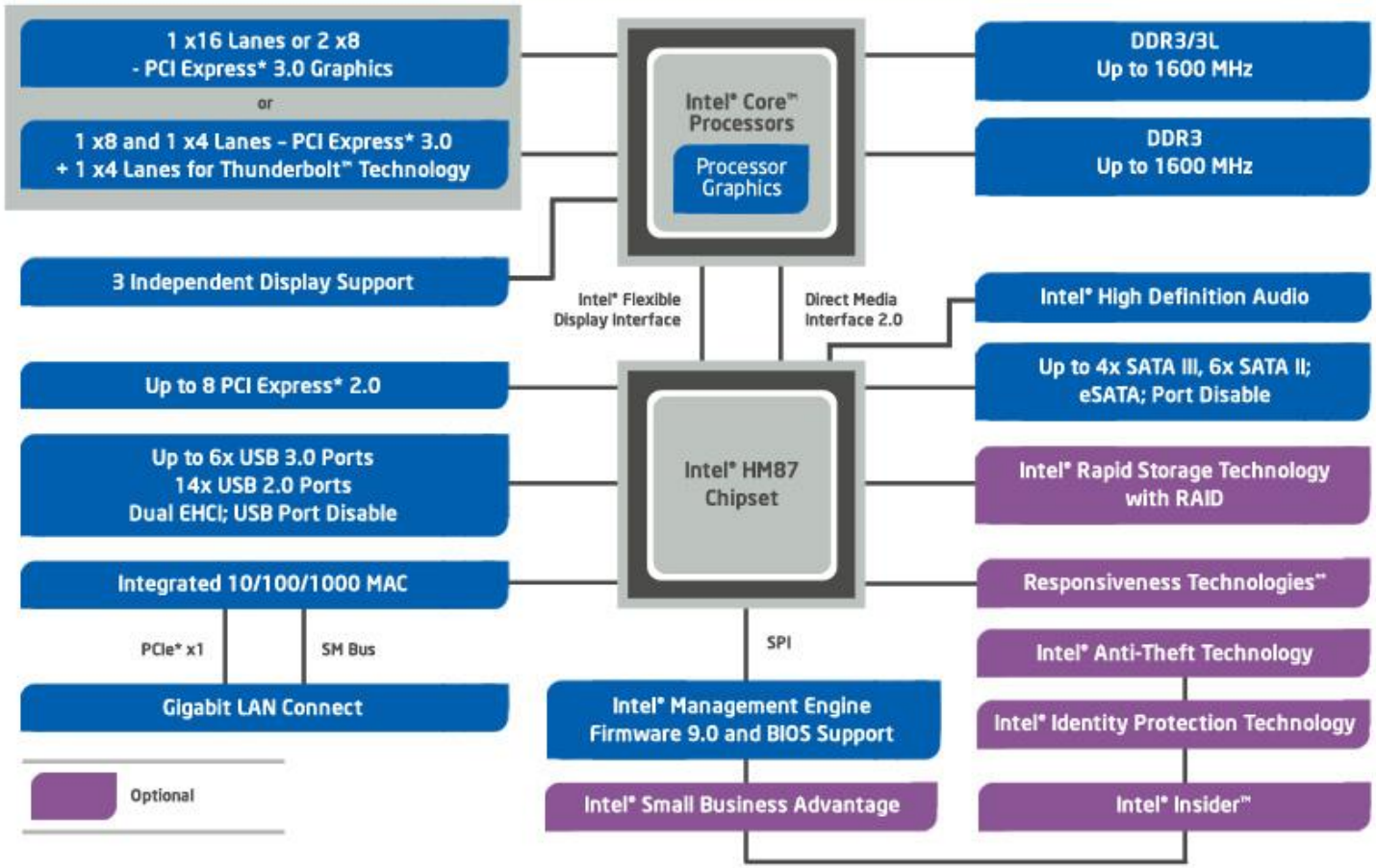
- TLB size: 16–512 entries
- Block size: 1–2 page table entries (typically 4–8 bytes each)
- Hit time: 0.5–1 clock cycle
- Miss penalty: 10–100 clock cycles
- Miss rate: 0.01%–1%


Intel
Pentium 4
Processor
400/533MHz FSB
P4X400
North Bridge
200/266/
333MHz
Memory
Bus
AGP8X
8X V-Link
Network Interface PHY
10/100Ethernet
PCI Slots
PCI Bus
VT8235
South Bridge
AC-Link
LPC
ATA
33 / 66 /
100 / 133
Direct
EPROM
Or LPC
6X USB
USB 2.0
VT1211
LPC
Super
I/O
Serial / IR
Parallel
Floppy Disk
Keyboard
Mouse
VT1616
AC'97 Audio Codec
MC-97
Modem Codec

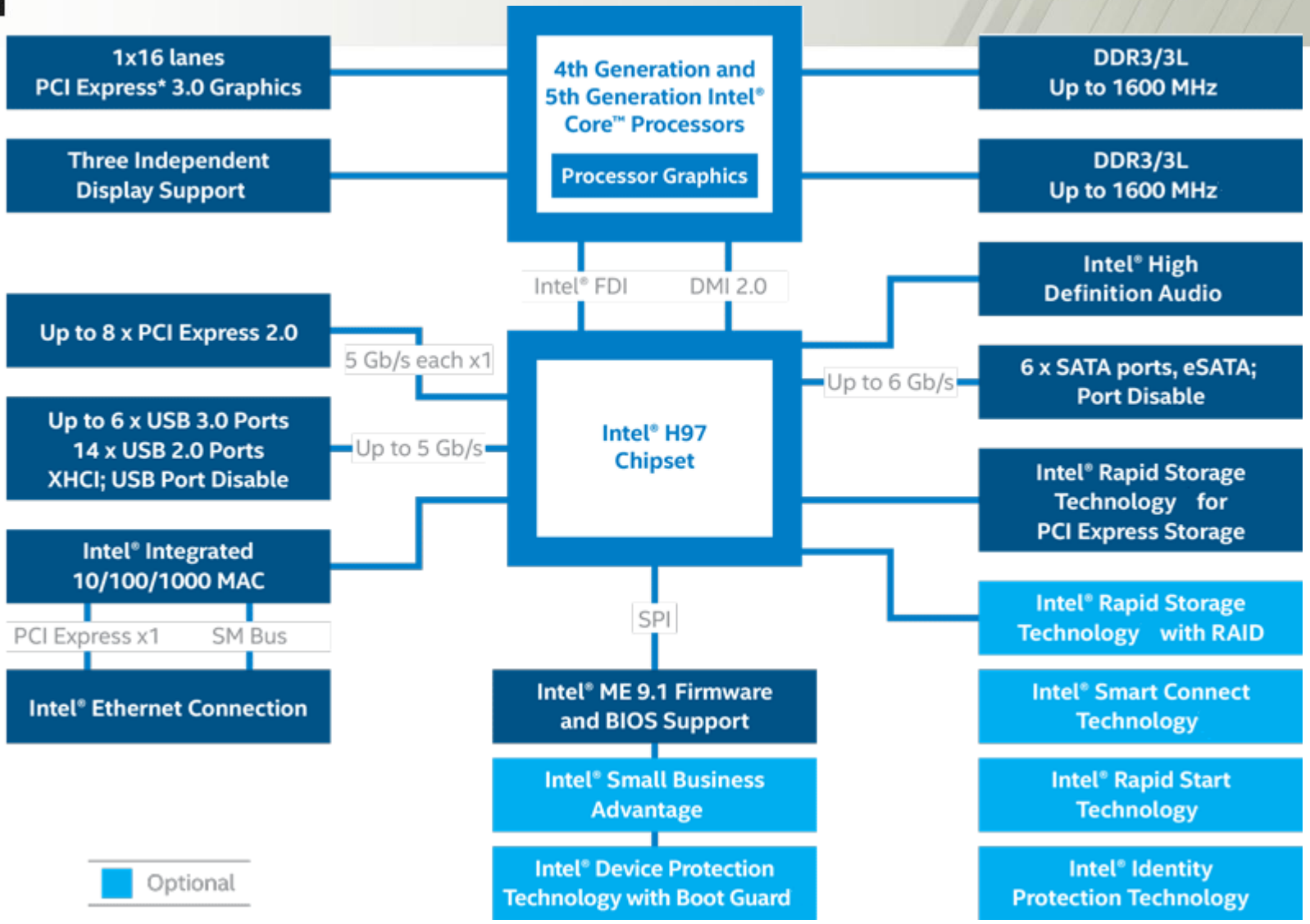

SiS746FX/SiS963L
System Block Diagram
AMD Athlon™ XP/Duron™CPUS
FSB 333MHz
AGP 3.0
8X Mode
Fast Write
AGP
SiS
746FX
DDR400/ DDR333/DDR266/
3 DIMMs Unbuffered DDR
Max.1GB per DIMM
MuTIOL 1G® 1GB/S, Bi-directional bus,
16 Bit @ 533Mhz
MuTIOL 1G®
Dual IDE
ATA133/100/66/33
USB
USB 2.0 /1.1
PCI
6 PCI
KBC
Keyboard/Mouse
SiS
963L
AC Link
Multiple
Codec
AC3 Digital Output
Analog 6 Channels
ACR
V.90 Modem
RJ11
MII
PHY
PHY
10Mb HPNA
or
10/100Mb LAN
RJ45
LPC
ROM
(Optional)
Floppy/MIDI/Joystick
MIDI
LPC
S/IO
BIOS
Legacy

AGH

Mobile Intel® HM87 Chipset Block Diagram

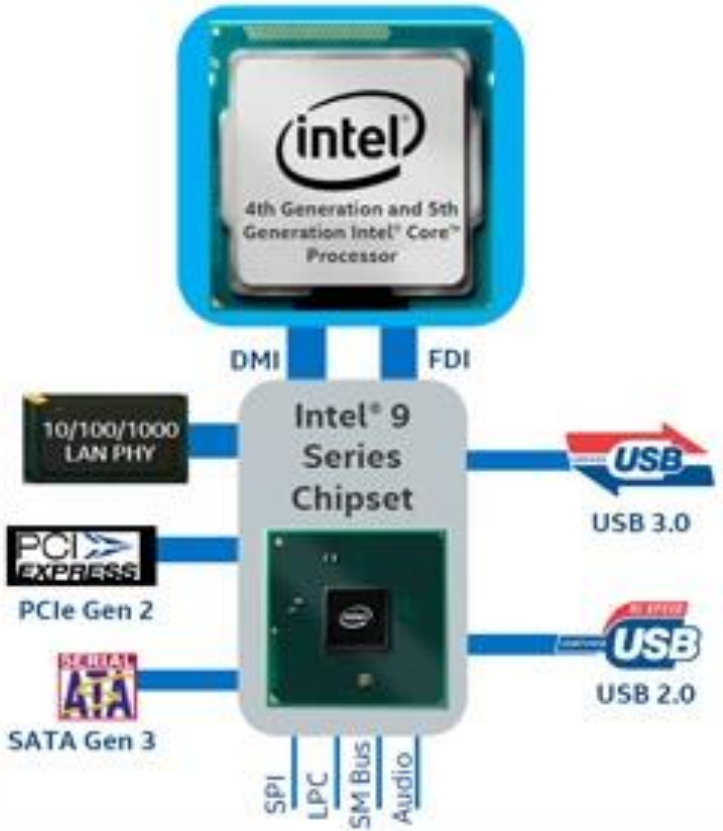


AGH


1x16 lanes
PCI Express* 3.0 Graphics
Three Independent
Display Support
4th Generation and
5th Generation Intel®
Core™ Processors
Processor Graphics
DDR3/3L
Up to 1600 MHz
DDR3/3L
Up to 1600 MHz
Intel® FDI
DMI 2.0
Intel® High
Definition Audio
Up to 8 x PCI Express 2.0
5 Gb/s each x1
6 x SATA ports, eSATA;
Port Disable
Up to 6 Gb/s
Up to 6 x USB 3.0 Ports
14 x USB 2.0 Ports
XHCI; USB Port Disable
Up to 5 Gb/s
Intel® H97
Chipset
Intel® Rapid Storage
Technology for
PCI Express Storage
Intel® Integrated
10/100/1000 MAC
PCI Express x1
SM Bus
SPI
Intel® Rapid Storage
Technology with RAID
Intel® Ethernet Connection
Intel® ME 9.1 Firmware
and BIOS Support
Intel® Smart Connect
Technology
Intel® Small Business
Advantage
Intel® Rapid Start
Technology
Intel® Device Protection
Technology with Boot Guard
Intel® Identity
Protection Technology
Optional

Intel® 9 Series Chipset Overview

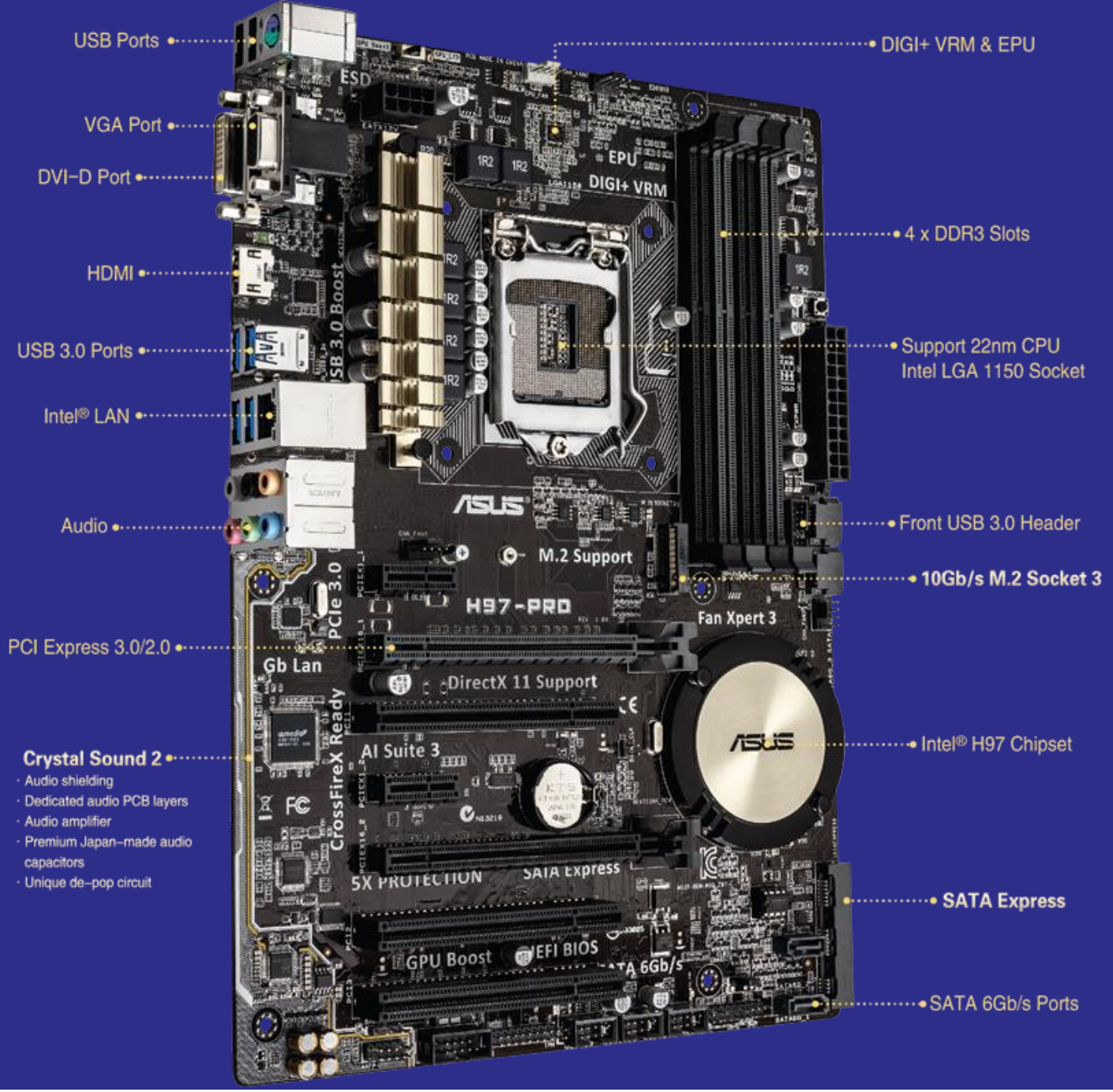
- 4th Generation and 5th Generation Intel® Core™ Processor Support
- Intel® RST for PCIe Storage
- Flexible I/O Port Selection
- High Speed SATA 6 Gb/s support for SSD/HDDs
- Intel® Rapid Storage Technology 13



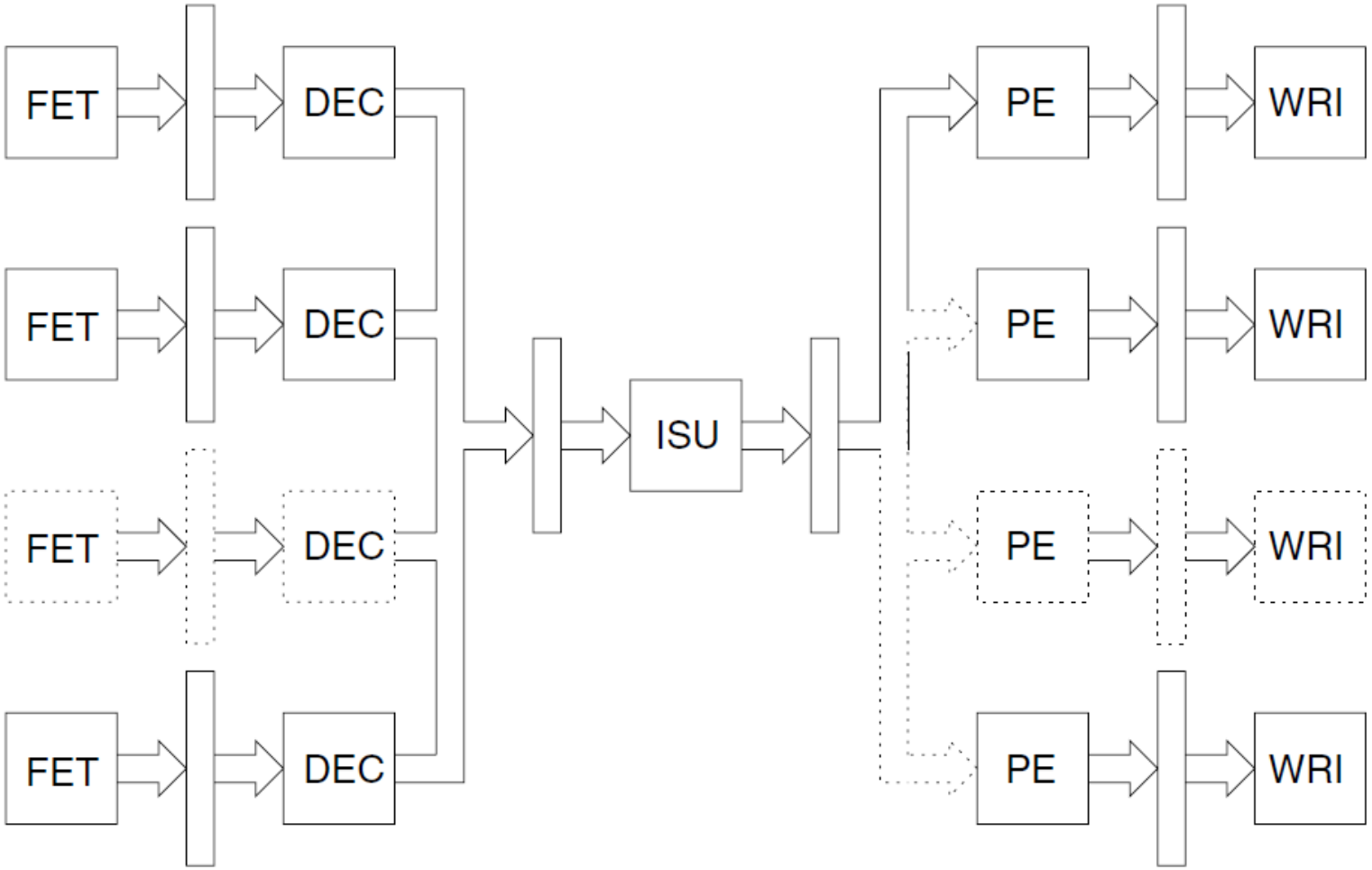
- Intel® Device Protection Technology with Boot Guard
- Intel® Small Business Advantage
- Integrated USB 3.0 (with Streams support)
- Storage Power Savings
- PCI Express 2.0 at 5 GT/s

New Intel Chipset with Updated Platform Capabilities

New with Intel® 9 Series


USB Ports
VGA Port
DVI-D Port
HDMI
USB 3.0 Ports
Intel® LAN
Audio
PCI Express 3.0/2.0
Crystal Sound 2
· Audio shielding
· Dedicated audio PCB layers
· Audio amplifier
· Premium Japan-made audio capacitors
· Unique de-pop circuit
DIGI+ VRM & EPU
4 x DDR3 Slots
Support 22nm CPU
Intel LGA 1150 Socket
Front USB 3.0 Header
10Gb/s M.2 Socket 3
Intel® H97 Chipset
SATA Express
SATA 6Gb/s Ports
ESD
EPU
DIGI+ VRM
USB 3.0 Boost
M.2 Support
H97-PRO
Fan Xpert 3
PCIe 3.0
Gb Lan
DirectX 11 Support
AI Suite 3
CrossFireX Ready
5X PROTECTION
SATA Express
GPU Boost
EFI BIOS
SATA 6Gb/s

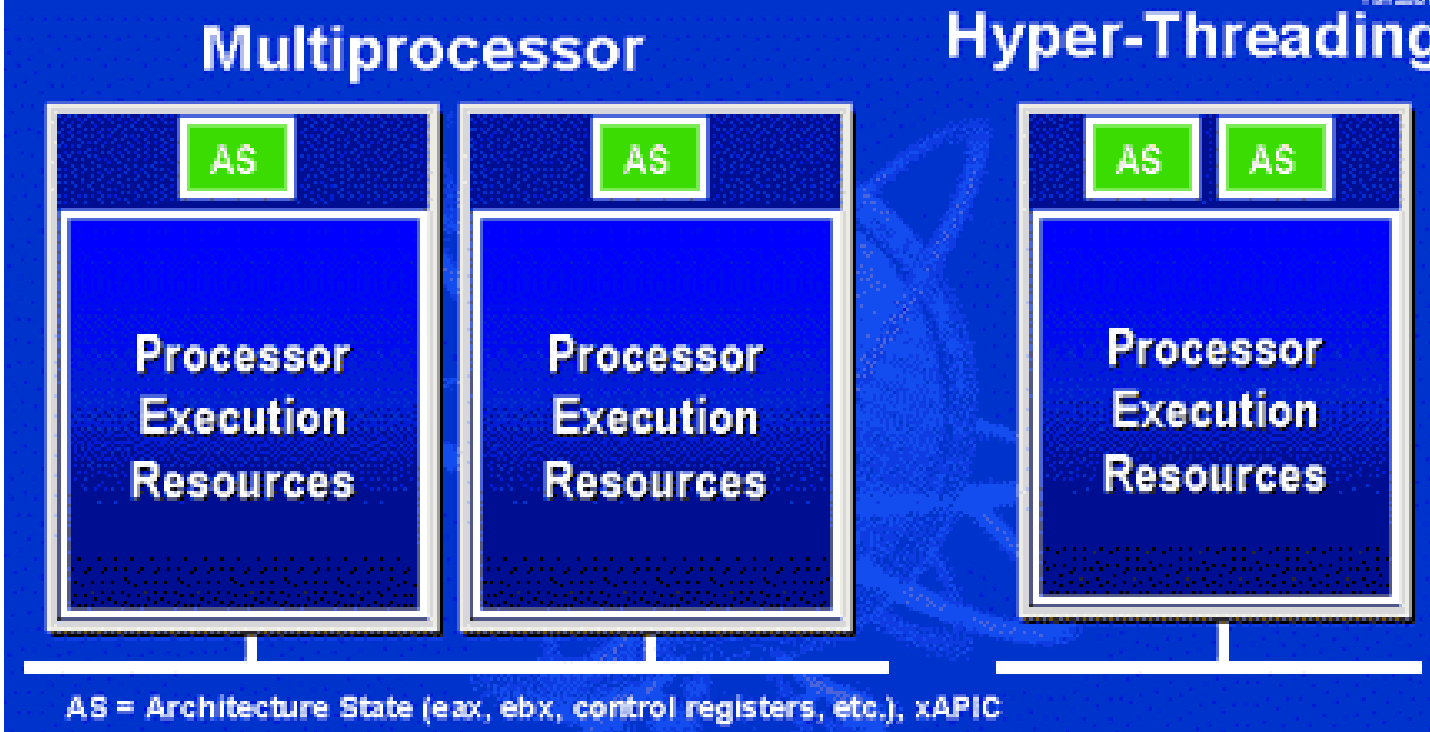
A conceptual diagram of a superscalar processor with hyperthreading



AGH

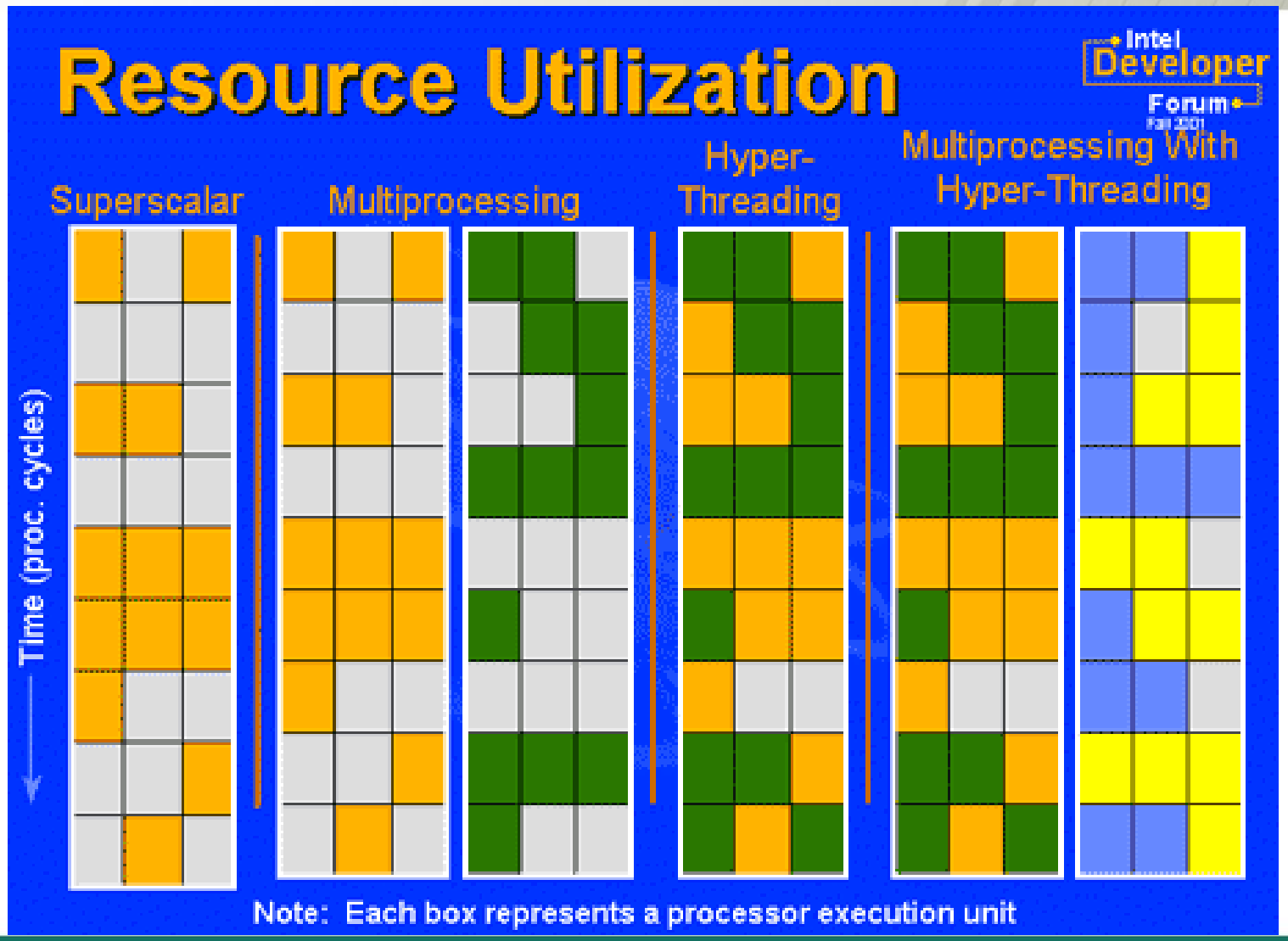
Hyper-Threading Technology

Intel Developer Forum Fall 2001



Hyper-Threading Technology looks like two processors to software

AGH


Resource Utilization
Intel Developer Forum
Superscalar
Multiprocessing
Hyper-Threading
Multiprocessing With Hyper-Threading
Time (proc. cycles)
Note: Each box represents a processor execution unit