Memory management
in protected and 64-bit mode

In operating systems, memory management is the function responsible for managing the computer's main memory.
It determines how memory is allocated among competing processes. When memory is allocated it determines which memory locations will be assigned.

Memory management schemes supported by the Intel processors:

- segmentation
- paging

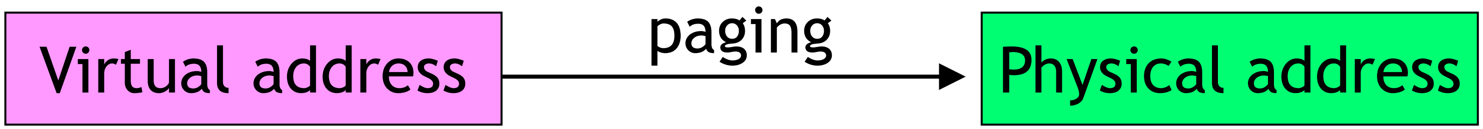
Mode		Operating system required	Type of code being run	Linear address [b]	General purpose registers [b]
Long mode	64-bit mode	64-bit	64-bit code	64	64
	Compatibility mode		32/16-bit protected mode	32	32
Legacy mode	Protected mode	32-bit	32/16-bit protected mode	32	32
	Virtual 8086 mode	32-bit	16-bit real mode	20	32
	Real mode	16-bit (starting mode for 16-, 32- and 64-bit OS)	16-bit real mode	20	16

Paging

Paging mechanism enables system software to create separate address spaces for each process or application. This address space is known as virtual address space.

Each process treats the main memory as a large continuous address space. In fact, its address space can be dispersed in various regions of physical memory or even in the secondary memory (hard disk).

Memory management unit in CPU translates virtual addresses into physical addresses.



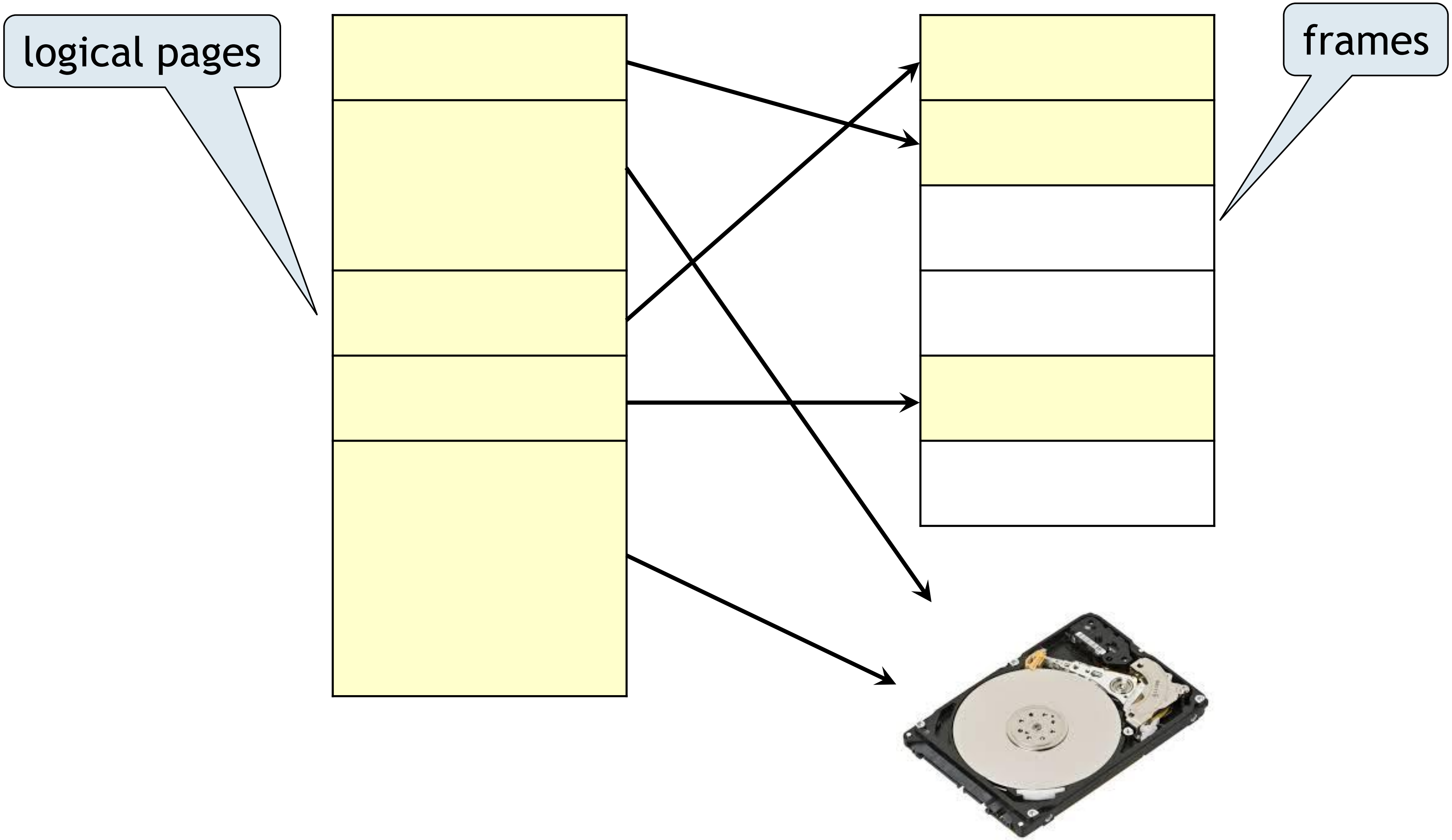
Virtual address

paging

Physical address

Virtual address space of a process

Physical memory

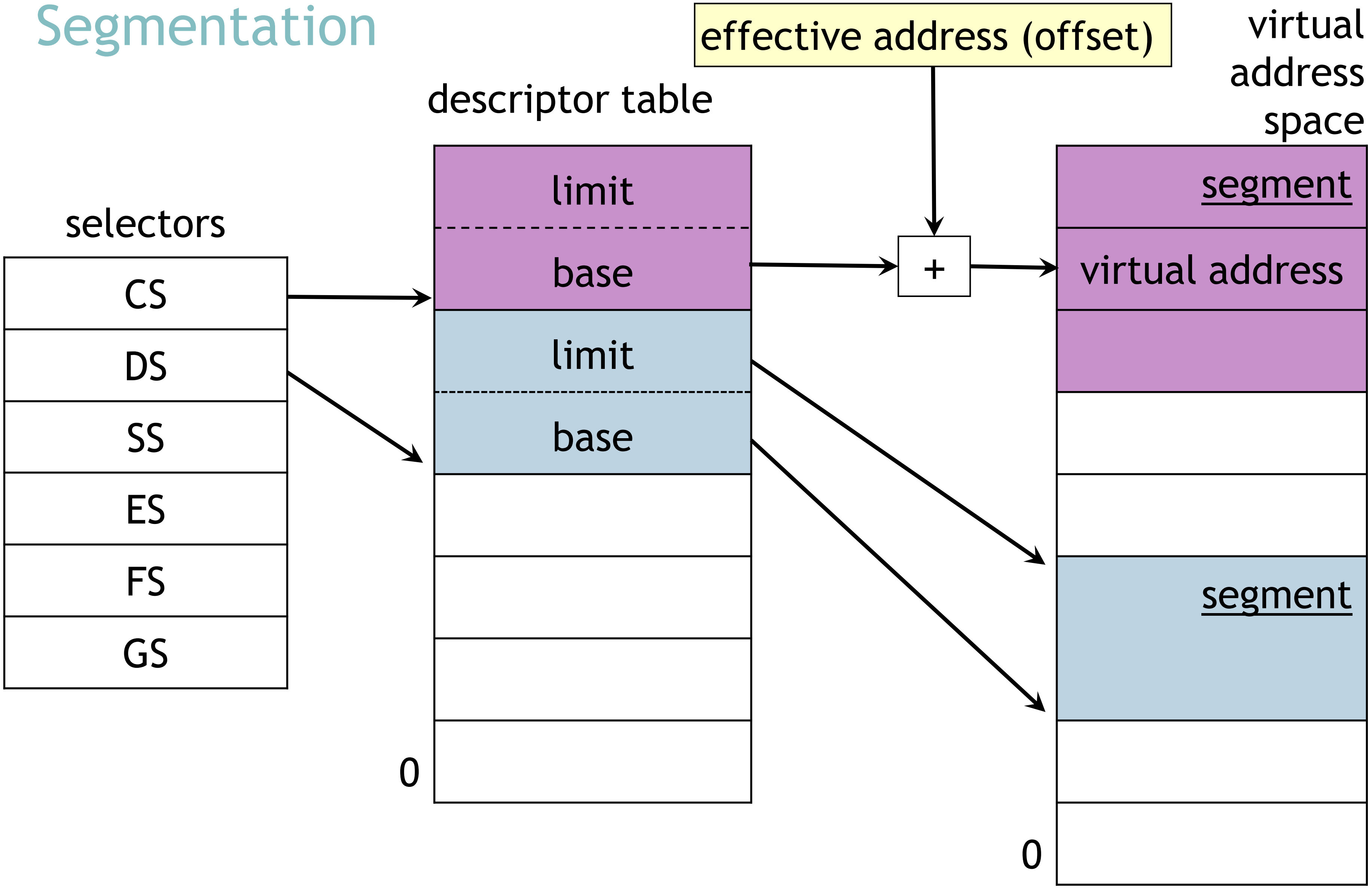


logical pages

frames

If a process uses an address on a logical page that is not in physical memory, the processor generates an exception (page-fault exception) and the operating system must handle it – assign a physical frame to the logical page and load the corresponding part of the code or data from the disk.

Segmentation



Base address - starting address of the segment

Limit - length of the segment (up to 4 GB)

Protected 32-bit mode

Memory model: flat

`.MODEL flat, stdcall` (in the include file SmallWin.inc)

All segment base addresses have a value of 0, the segment limits are fixed at 4 GB =>

- virtual address = effective address (offset)
- data, code and stack segments overlap.

  When running the application, the operating system stores the data and code in such a way so that they do not overwrite (first stack, then code, then data) and sets the EIP and ESP registers in a corresponding way.

In compatibility mode, the application uses the first 4 GB of virtual-memory space. Access to virtual memory above 4 GB requires the use of 64-bit mode.

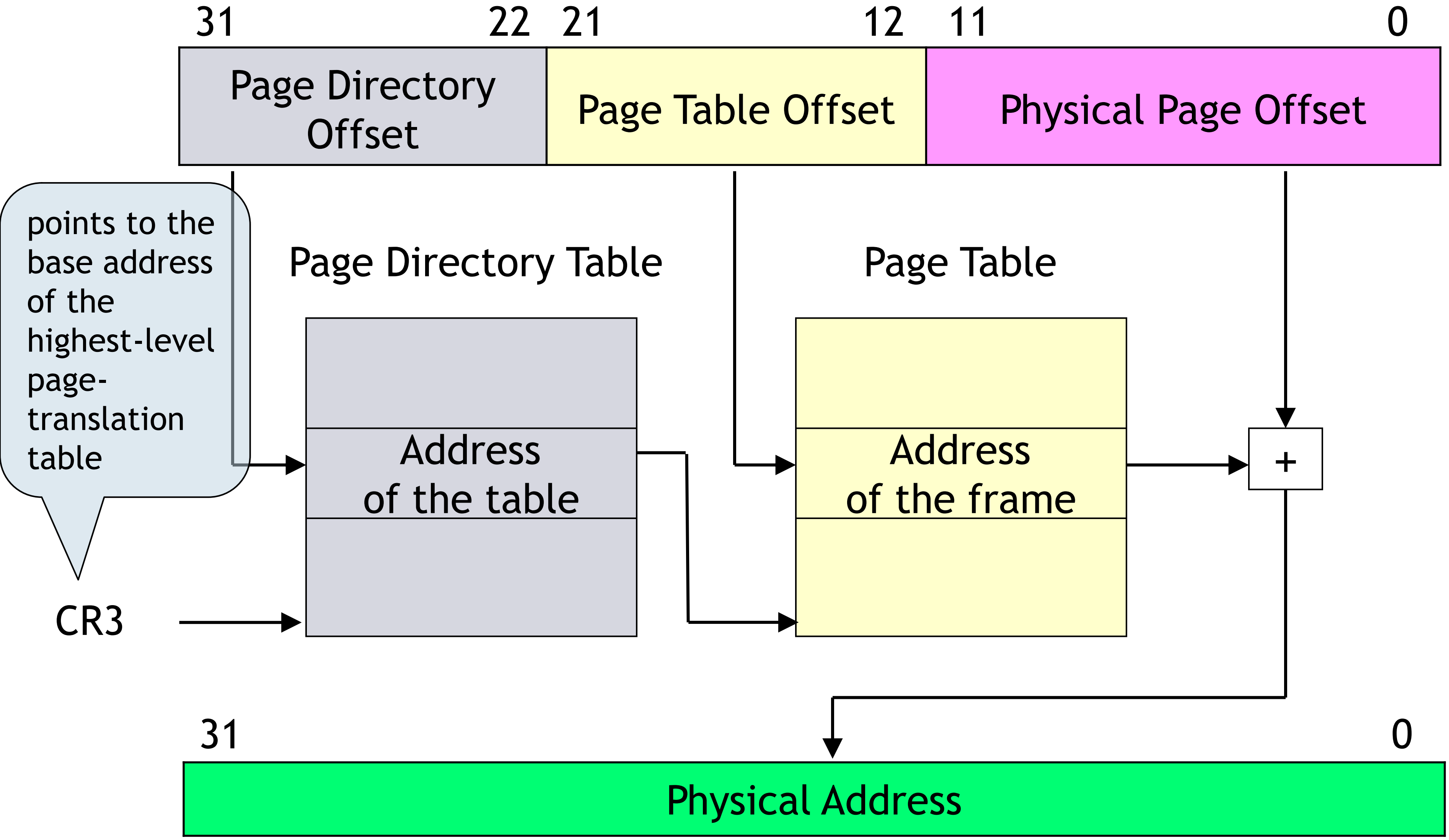
64-bit mode

Segmentation is disabled.
The segment base is treated as if it were 0, and the segment limit is ignored.
This allows an effective addresses to access the full virtual-address space supported by the processor.

Virtual to physical address translation - 32-bit mode

Page size: 2^{12} B = 4 kB.

Virtual address:



31 22 21 12 11 0

Page Directory Offset

Page Table Offset

Physical Page Offset

points to the base address of the highest-level page-translation table

Page Directory Table

Page Table

Address of the table

Address of the frame

+

CR3

31 0

Physical Address

Translation Lookaside Buffer (TLB)

- a special on-chip cache
- is used to reduce the time taken to access a memory location
- contains recently translated physical addresses. Every memory reference is first checked in the TLB. If the requested address is not in the TLB, the translation proceeds through page tables.

Page table entry

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Physical Page Base Address															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				AVL			G	PAT	D	A	PCD	PWT	U/S	R/W	P

Available to Software (AVL) Bit. These bits are not interpreted by the processor and are available for use by system software.

Bit		Meaning
P	Present	1 – page is loaded in physical memory, 0 – is not
R/W	Read/Write	0 – page is read only
U/S	User/Supervisor	0 – access is restricted to supervisor level (CPL 0, 1, 2) 1 – user access is allowed (also CPL 3)
PWT	Page-Level Writethrough	0 – physical page has a writeback caching policy 1 – writethrough
PCD	Page-Level Cache Disable	0 – physical page may be in the cache memory 1 – not cacheable
A	Accessed	0/1 – page has not/has been accessed (read from or written to)
D	Dirty	0/1 – page has not/has been written
PAT	Page Attribute Table	Cache memory management together with the PWT and PCD bits.
G	Global Page	1 – the TLB entry for the global page is not invalidated during a task switch

Current privilege level (CPL):

0 – BIOS, memory management, interrupt service routines

1 – other parts of the OS (device drivers)

2 – development tools (compilers, ...)

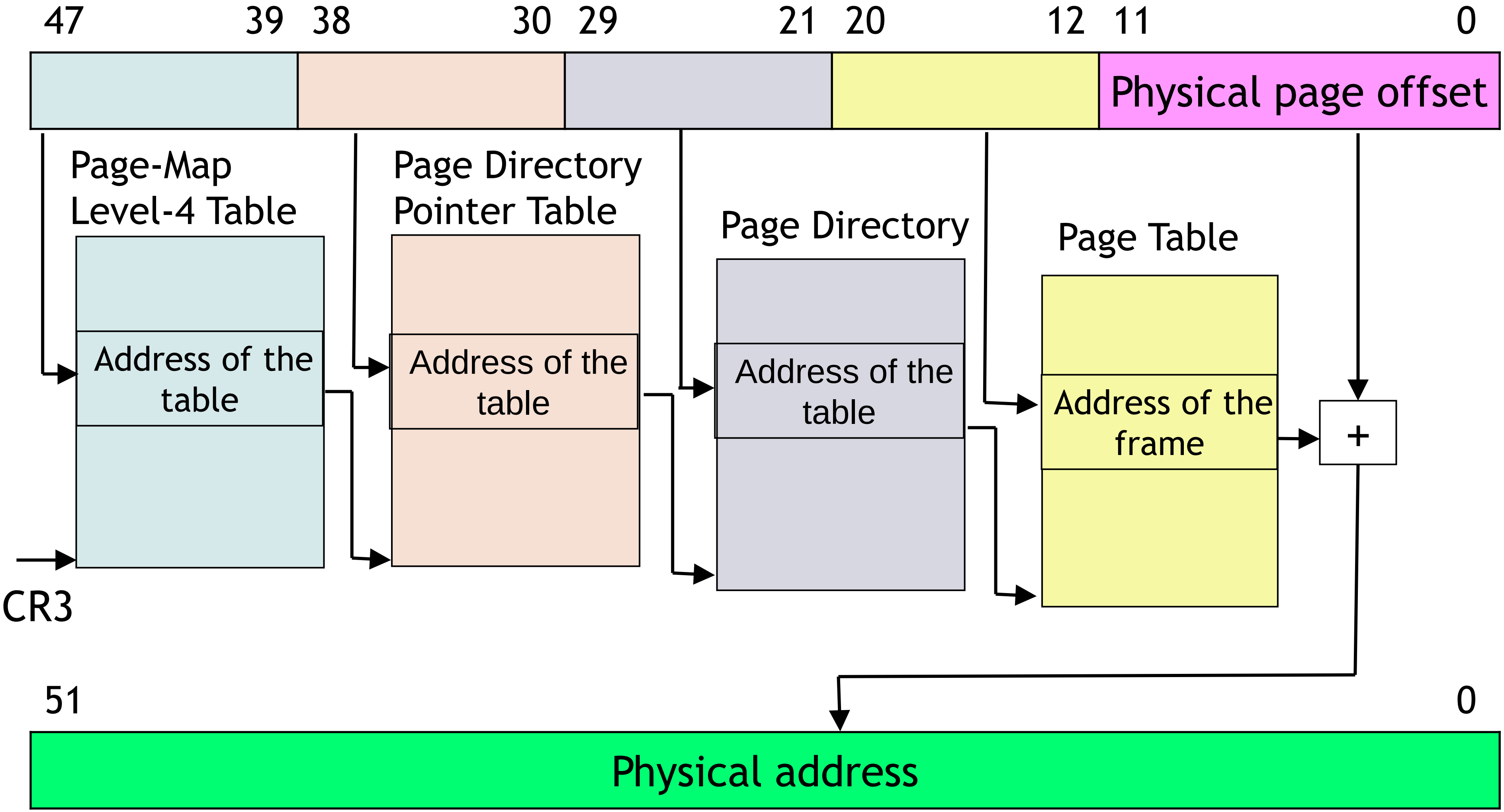
3 – application programs

The A bit (Accessed) is set to 1 by the processor the first time the physical page is either read from or written to. The A bit is never cleared by the processor. Instead, software must clear this bit to 0 when it needs to track the frequency of physical-page accesses.

Page translation scheme in 64-bit mode

Page size: 2^{12} B = 4 kB.

Bits 48 – 63 of the virtual address copy bit 47.



Paging mechanism:

- provides each process with its own private region of physical memory for storing its code and data;
- allows physical pages to be shared by multiple processes and applications. The physical pages can be configured by the page tables to allow read-only access. This prevents applications from altering the pages and ensures their integrity for use by all applications.
- the memory access rights can be controlled, much as they can for segments;
- allows to map multiple, large virtual-address space to a limited physical memory;
- the consolidation of free memory is not needed.