The quest for speed

Saulius Gražulis

Vilnius, 2020

Vilnius University, Faculty of Mathematics and Informatics
Institute of Informatics

Long-mode memory management

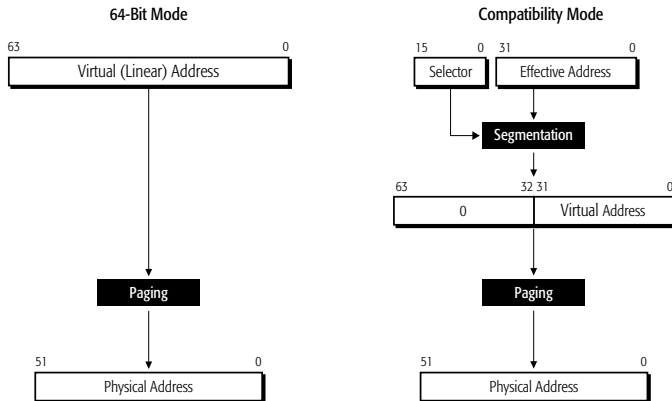*Besser als viel Speicher ist noch mehr Speicher...*

(Beter than much memory is even more memory...)

<div align="right">A German joke</div>
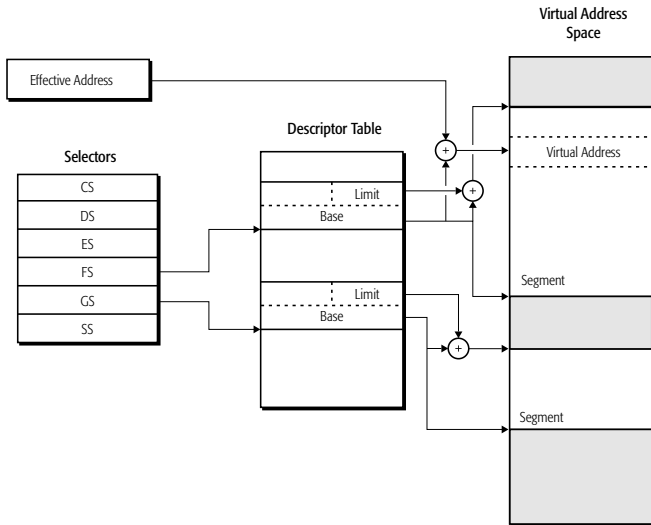
- Paging
- Segmentation

Long-mode memory management

x86, 86_64 CPU

64-Bit Mode

63		0
	Virtual (Linear) Address	

Compatibility Mode

Selector (15 ... 0) | Effective Address (31 ... 0)

Segmentation

63	32	31	0
	0		Virtual Address

Paging (64-Bit Mode)

Paging (Compatibility Mode)

51		0
	Physical Address	

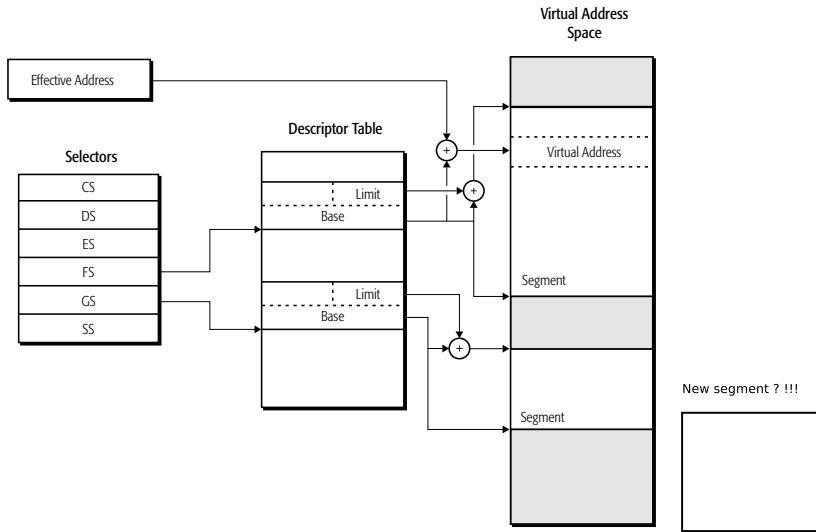| 51		0
	Physical Address	

(AMD 2017)

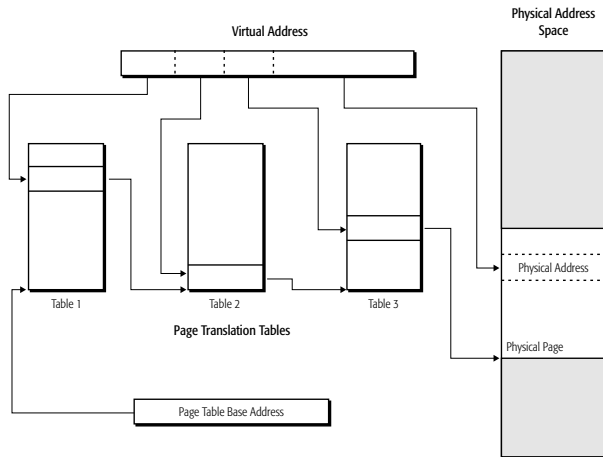# Virtual memory. Segmentation

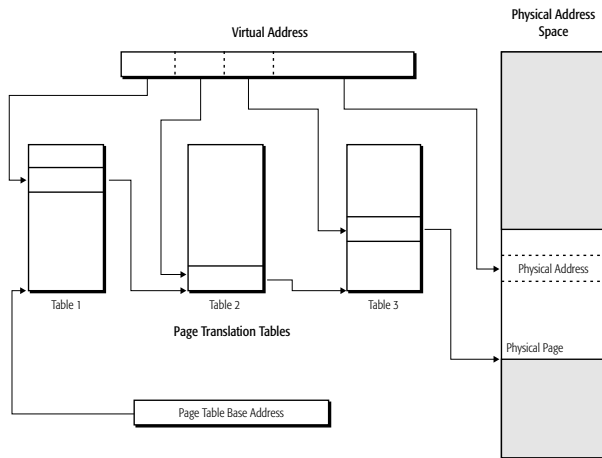

(Жмакин 2006; AMD 2012)

Virtual memory. Segmentation
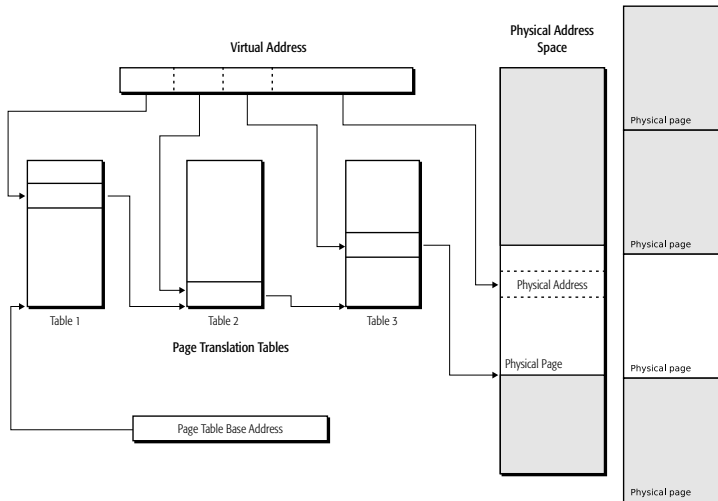

(Жмакин 2006; AMD 2012)

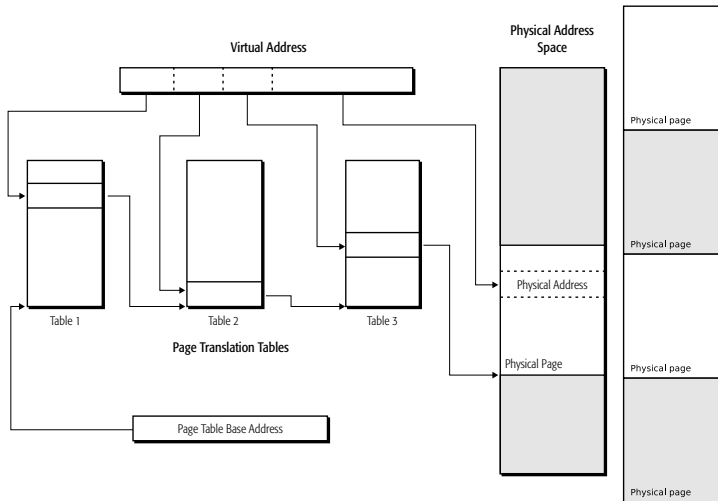# Virtual memory. Paging

(AMD 2012)

Virtual memory. Paging

Translation-Lookaside Buffer (TLB) ⇔ page-translation cache
(AMD 2012)

Virtual memory. Paging

Translation-Lookaside Buffer (TLB) ⇔ page-translation cache
(AMD 2012)

Virtual memory. Paging

Translation-Lookaside Buffer (TLB) ⇔ page-translation cache
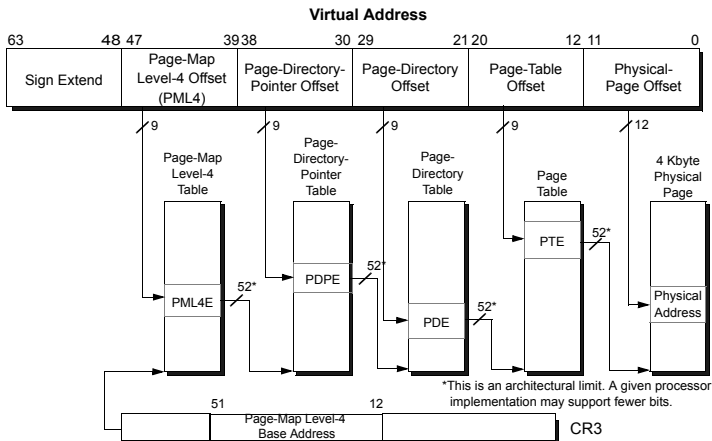(AMD 2012)

Virtual memory. Various page sizes

Figure 5-17.   4-Kbyte Page Translation—Long Mode

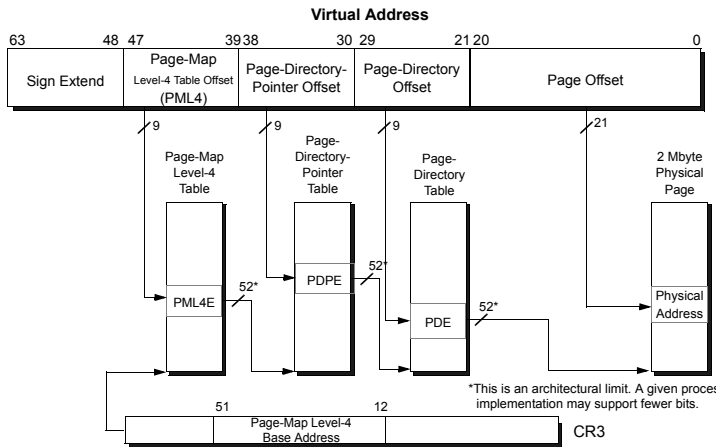(AMD 2012)

Virtual memory. Various page sizes

Figure 5-22.   2-Mbyte Page Translation—Long Mode

(AMD 2012)

Virtual memory. Various page sizes
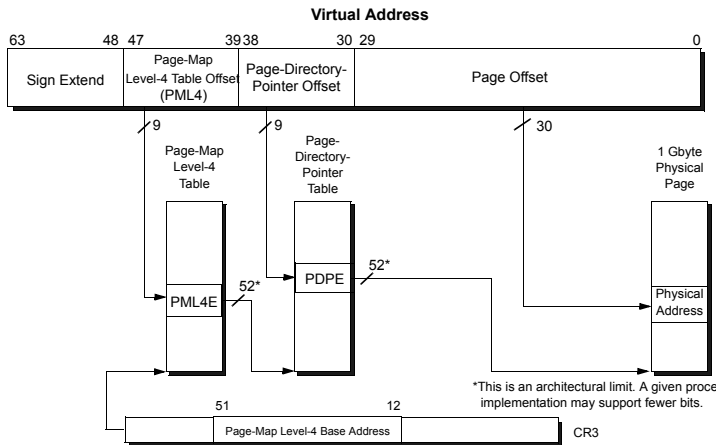
Figure 5-26.   1-Gbyte Page Translation—Long Mode

(AMD 2012)

Virtual memory. Page attributes

63	62	52	51	32
N X	Available		Page-Table Base Address (This is an architectural limit. A given implementation may support fewer bits.)	

31	12	11	9	8	7	6	5	4	3	2	1	0
Page-Table Base Address		AVL		IGN	0	IGN	A	PCD	PWT	U/S	R/W	P
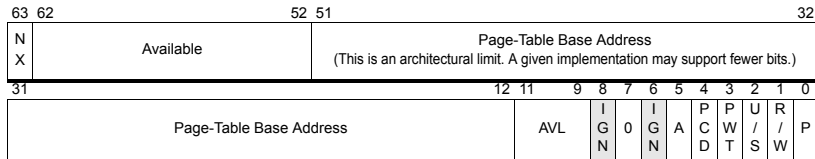
Figure 5-20.   4-Kbyte PDE—Long Mode

63	62	52	51	32
N X	Available		Physical-Page Base Address (This is an architectural limit. A given implementation may support fewer bits.)	

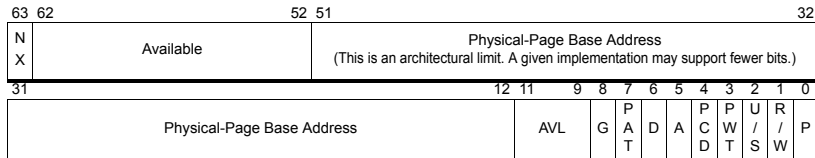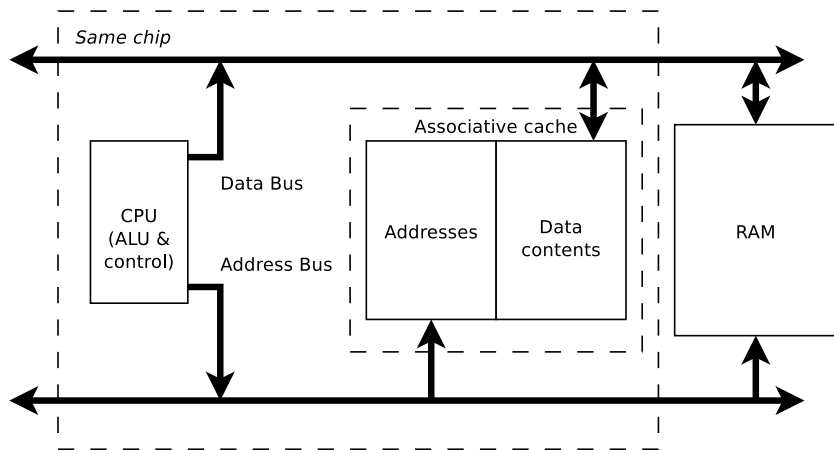| 31	12	11	9	8	7	6	5	4	3	2	1	0
Physical-Page Base Address		AVL		G	PAT	D	A	PCD	PWT	U/S	R/W	P

Figure 5-21.   4-Kbyte PTE—Long Mode

(AMD 2012)

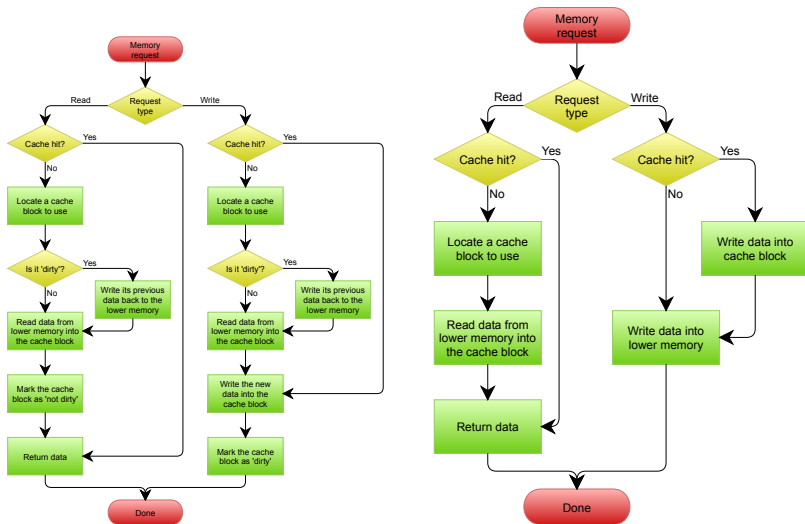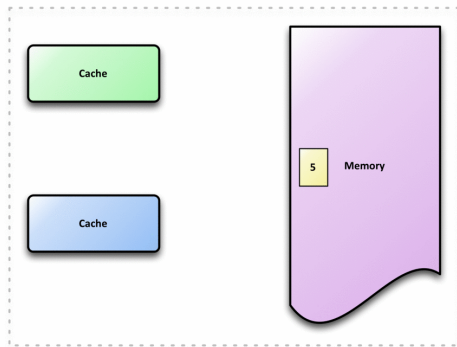# Cache memory
Associative cache

(Жмакин 2006)

Cache updates

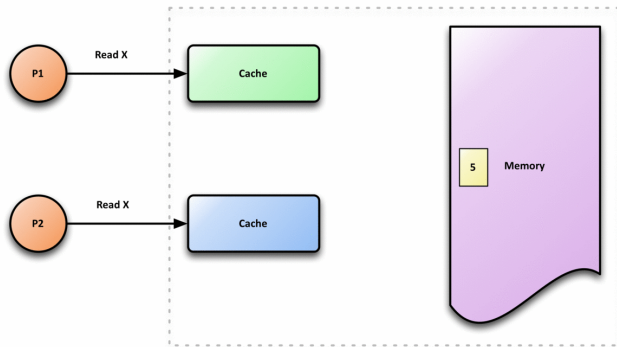Flin00, CC0, via Wikimedia Commons
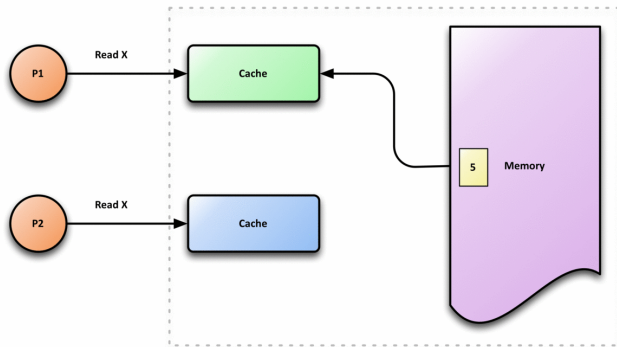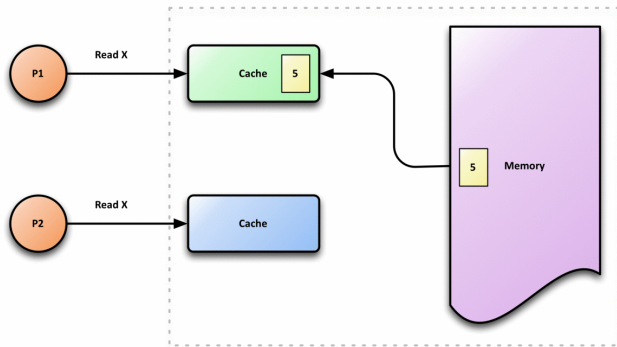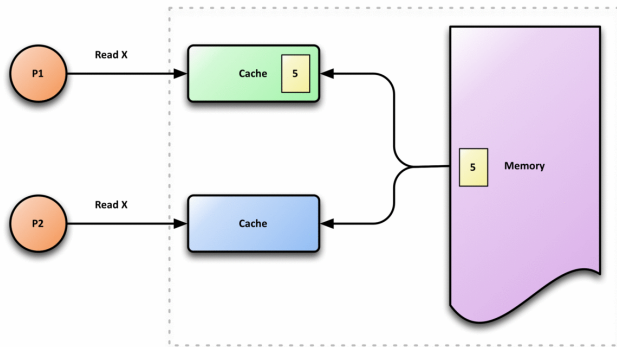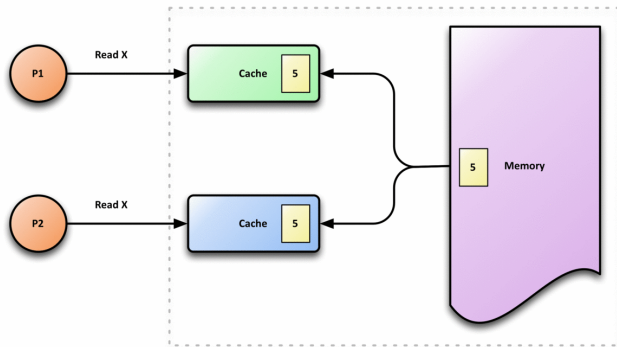
Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0
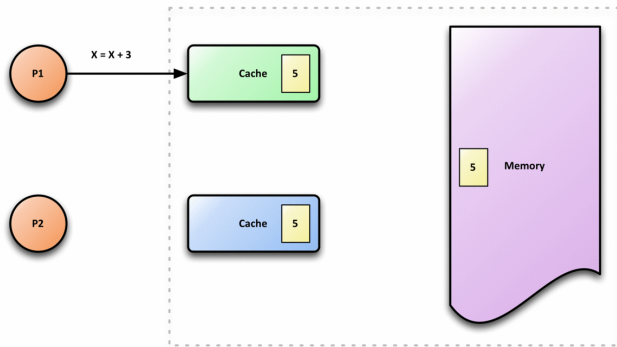
Cache coherence
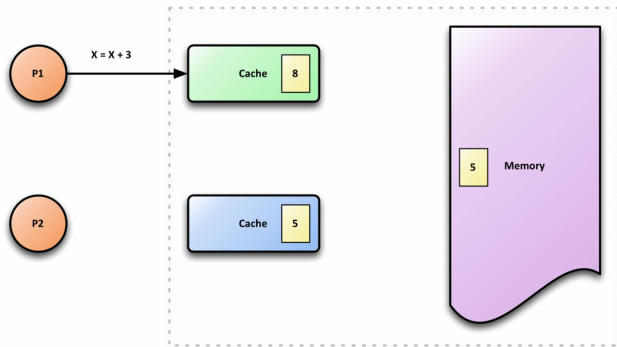
Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

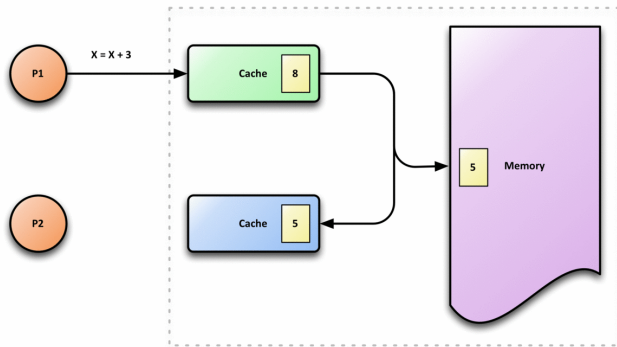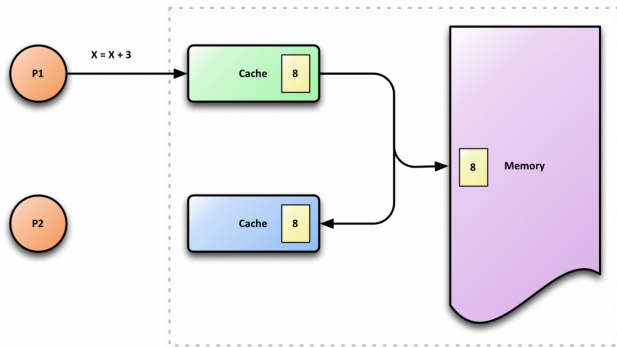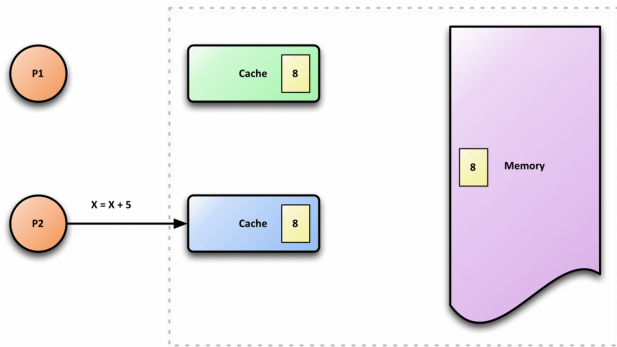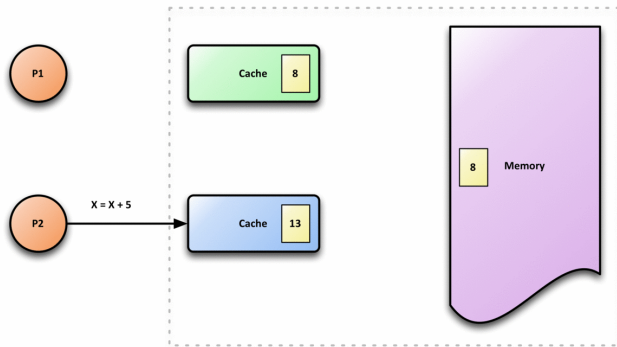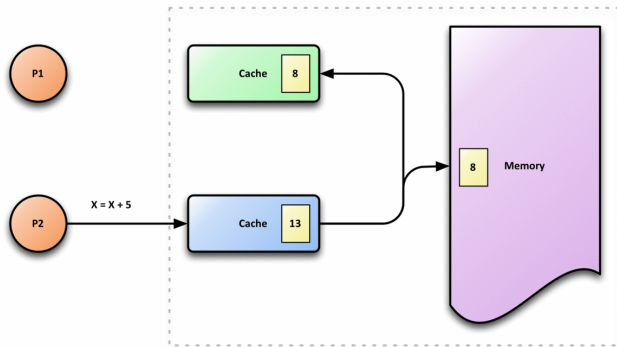By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence
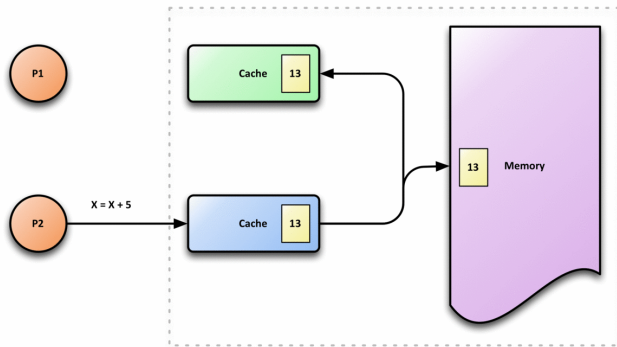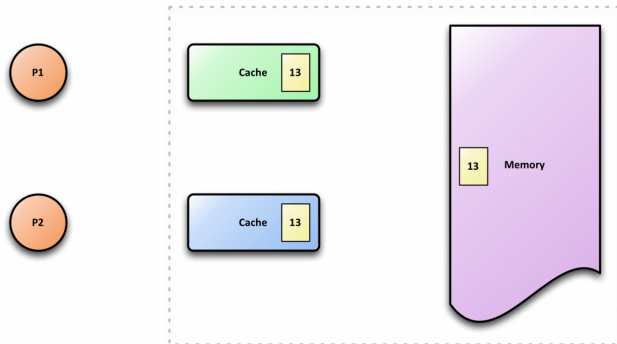

By M3tainfo - Own work, CC BY-SA 4.0

By M3tainfo - Own work, CC BY-SA 4.0

Cache coherence

By M3tainfo - Own work, CC BY-SA 4.0

Scientific number notation

$$602\underbrace{00\ldots0}_{21\text{ times}} = 6.02 \times 10^{23}$$

$$\pm d_0.d_1 d_2 \ldots d_{p-1} \times \beta^e = \sum_{i=0}^{p-1} d_i \beta^{-i} \times \beta^e, (0 \leq d_i < \beta)$$

Floating point numbers

$$\pm d_0.d_1 d_2 \ldots d_{p-1} \times \beta^e = \sum_{i=0}^{p-1} d_i \beta^{-i} \times \beta^e, (0 \le d_i < \beta)$$

$$\beta \ = 2$$

$$0.1 \approx 1.10011001100110011001101 \times 2^{-4}$$

IEEE 754 Standard

- Signed magnitude (significand)
- Biased exponent
- Hidden (assumed) bit

$$0.1 \approx 1.1001100110011001101 \times 2^{-4}$$

p: 23+1 bit e: $-126 - 127$ (8 bits)
$f = 1.10011001100110011001101$
$e = 127 + (-4) = 123_{10} = 01111011_2$

0 01111011 10011001100110011001101

References

AMD (Sept. 2012). *AMD64 Architecture Programmer's Manual, Volume 2: System Programming*. Vol. 2. AMD. URL: http://developer.amd.com/wordpress/media/2012/10/24593%5C_APM%5C_v2.pdf.

— (Dec. 2017). *AMD64 Architecture Programmer's Manual, Volume 1: Application Programming, revision 3.22*. AMD. URL: https://www.amd.com/system/files/TechDocs/24592.pdf.

Жмакин, А. П. (2006). *Архитектура ЭВМ*. БХВ-Петербург. ISBN: 5-94157-719-2.