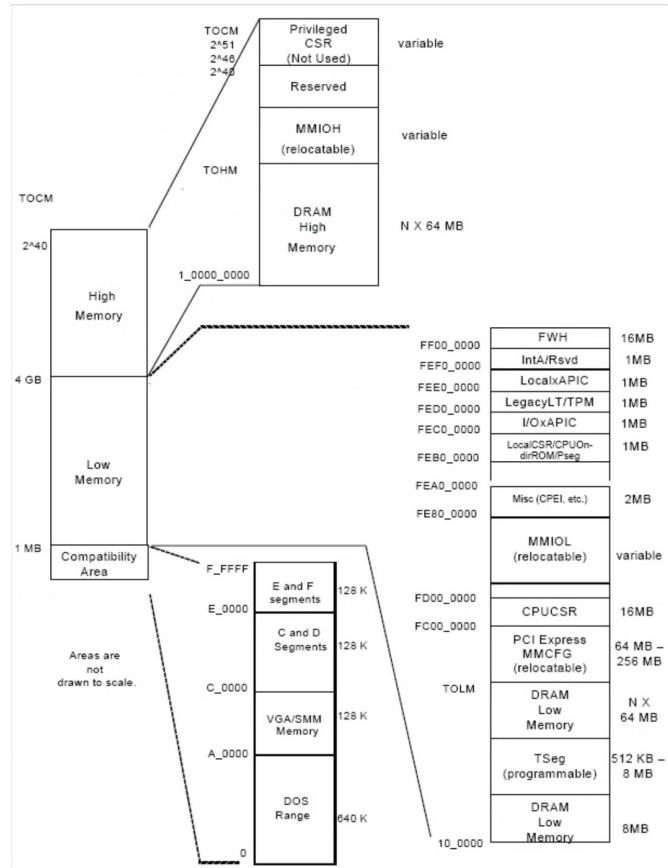
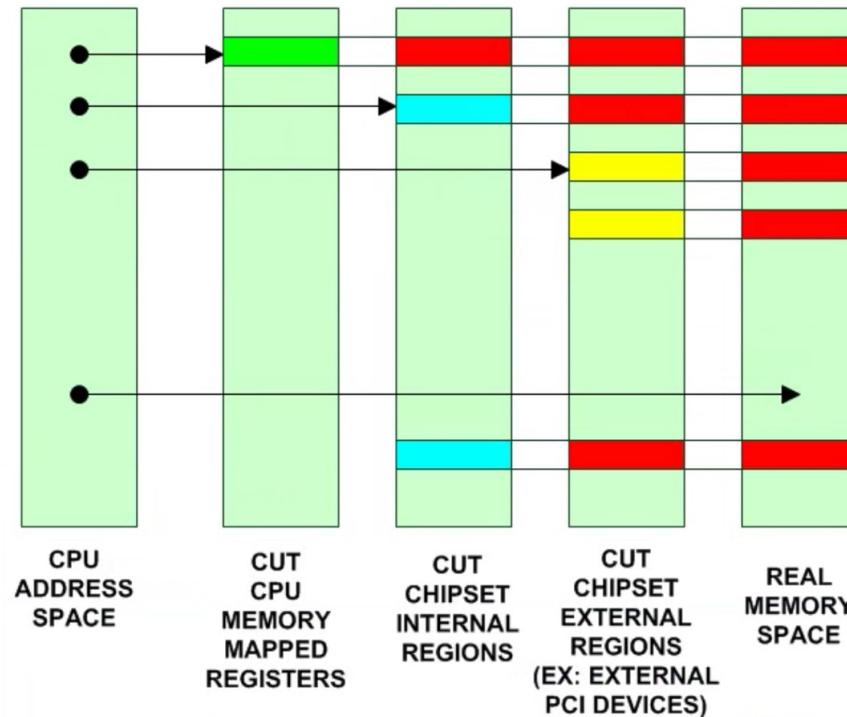


Виртуальная память.
Управление физической памятью.

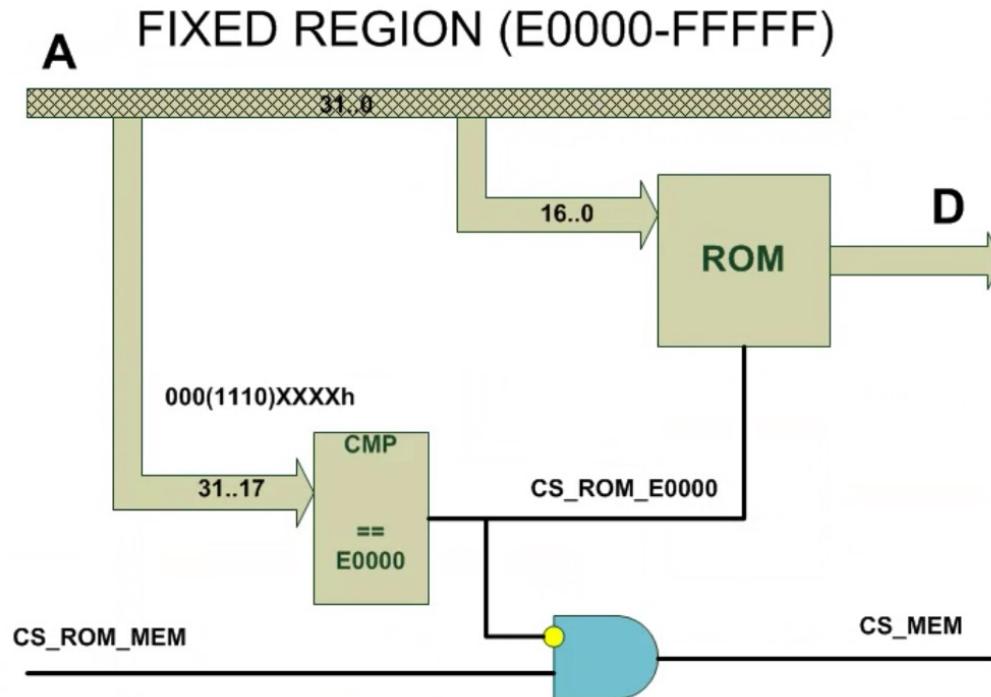
Карта памяти. Различные адресные пространства



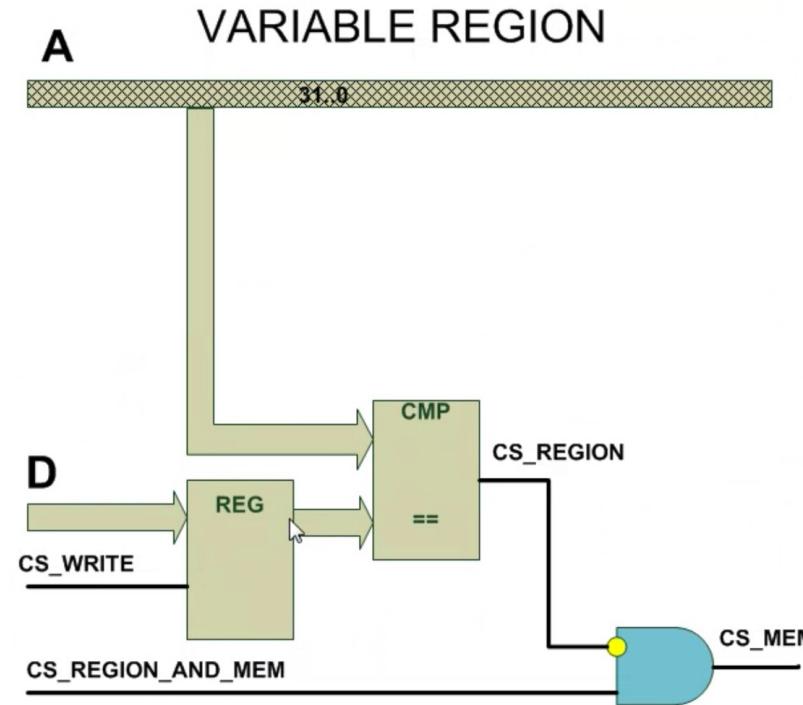
Адресные пространства. Обрезание регионов



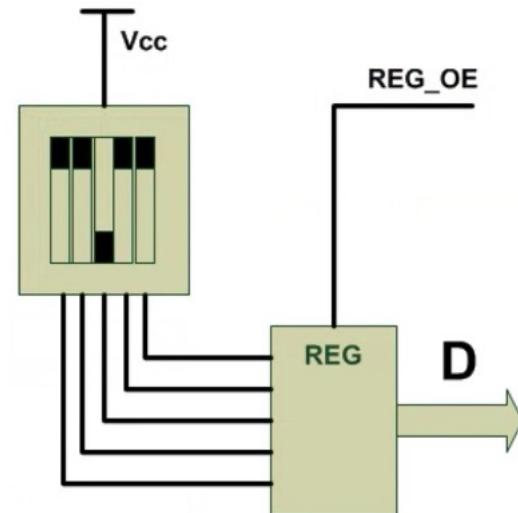
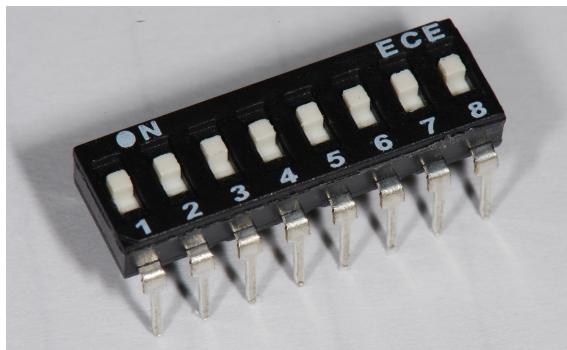
Обрезание фиксированного региона



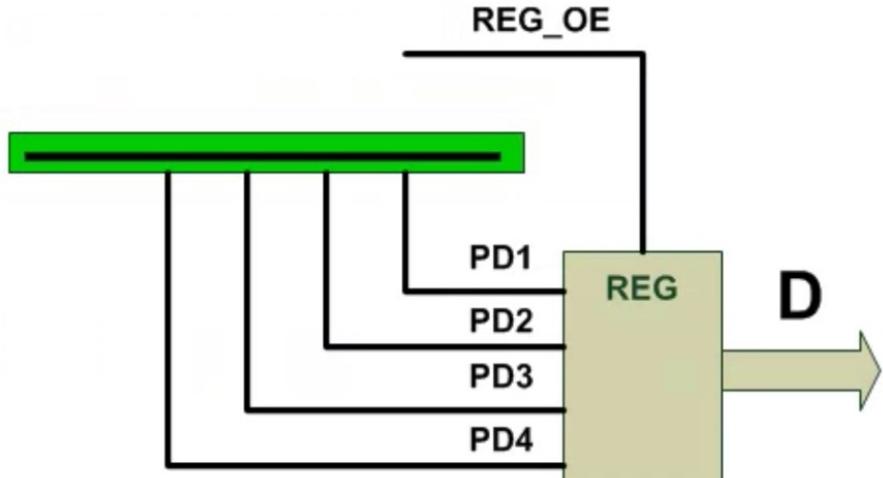
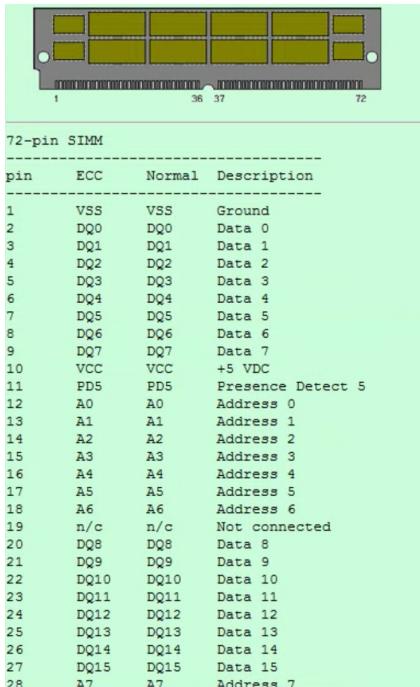
Обрезание перемещаемого региона



Настройка памяти. DIP-switches



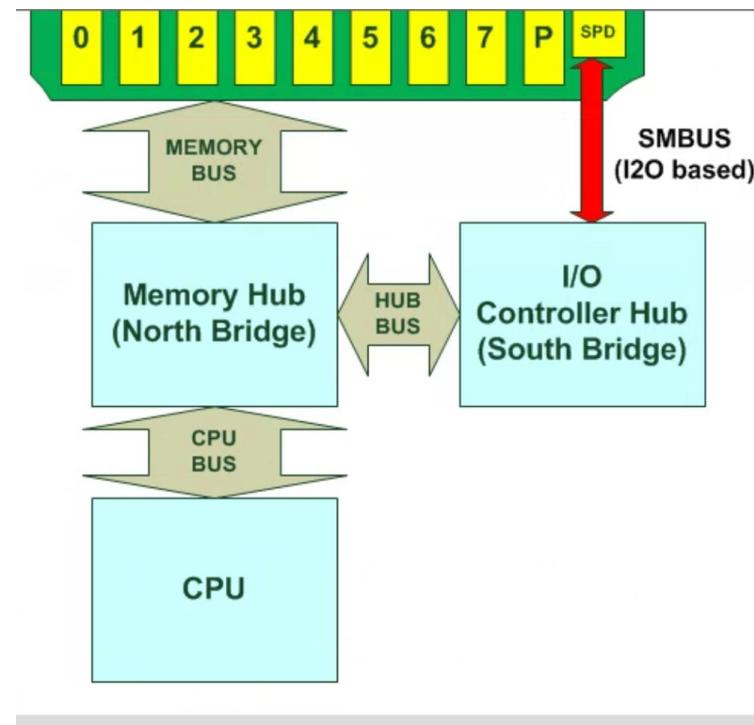
Serial Presense Detection SIMM 1



Serial Presense Detection SIMM 1

SDA	- Serial presents detect Data	
SA[3:0]	- Serial presents detect address	
SCL	- Serial presents clock	
WP	- Write protected	
SPD (Serial Presents Detect)		
на примере Micron DDR 256MB module		
Byte	Description	Value
0	Number of bytes used in SPD	80 = 128
1	Total number of SPD bytes	08 = 256
2	Memory type	07 = DDR
3	Number of ROW addresses	12
4	Number of column adderesses	13
5	Number of banks	2
6	Module data width	72
7	Module data width (continued)	0
8	Module interface voltage level	04 = SSTL 2.5V
9	SDRAM cycle time	70
10	SDRAM access from clock	75
11	Module configuration type	02 = ECC
12	Refresh rate/type	80
13	Refresh width	08
14	Error checking SDRAM data width	08
15	Minimum clock delay random col access	01
16	Burst length supported	0E = 2,4,8
17	Number of banks on SDRAM device	04
18	CAS latencies supported	0C = 2, 2.5
19	CS latency	01 = 0
20	WE latency	02 = 1
21	SDRAM Module attributes	20
22	SDRAM Device attribute	00
23	SDRAM cycle time (CAS latency = 2)	75
24	SDRAM cycle time (CAS latency = 2)	75
25	SDRAM cycle time (CAS latency = 1)	00
26	SDRAM cycle time from CK (CAS lat = 1)	00
27	Minimum ROW precharge time	50
28	Minimum ROW active to ROW active	3C
29	Minimum RAS# to CAS# delay	50
30	Minumum RAS# pulse width	28
31	Module bank density	20 = 256K
32	Address and command setup time	A0

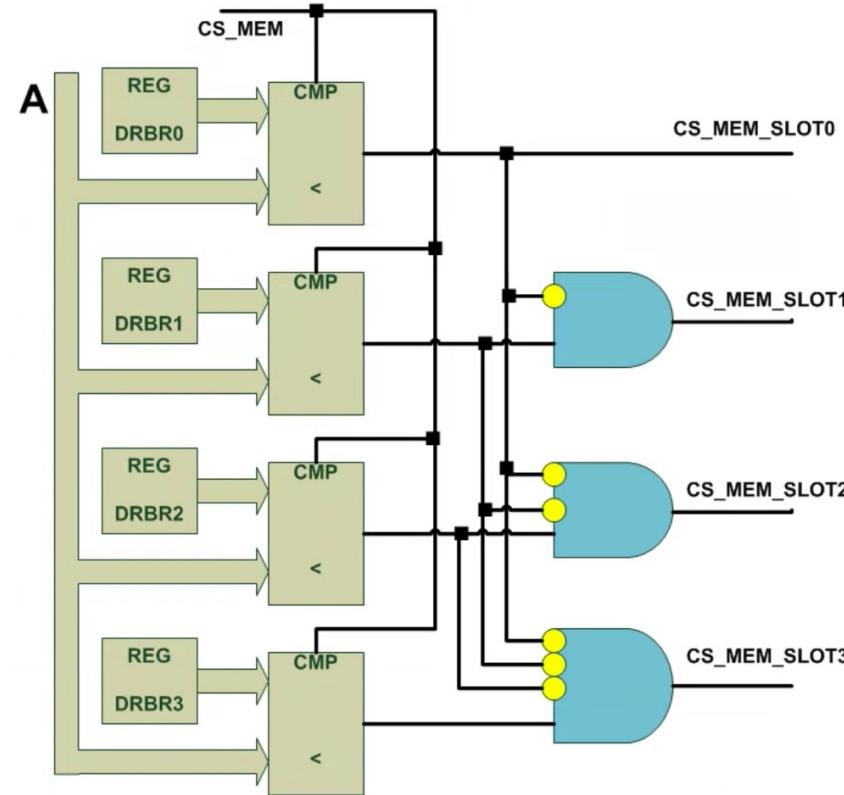
SMBUS



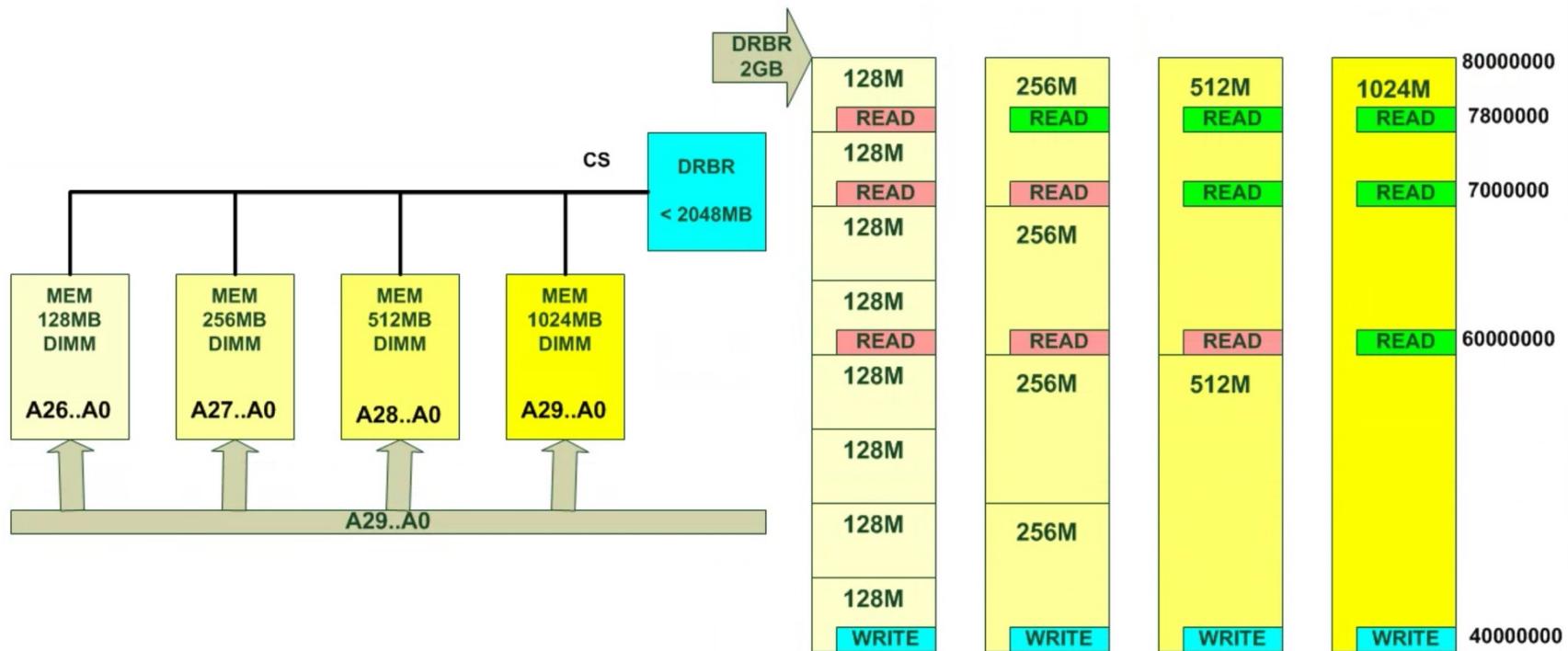
Настройки памяти intel 845PE

```
-----  
For each row: DRAM Row Boundary Register (DRBR)  
define upper boundary address of DRAM raw in 32-MB units  
DRB0 = row 0 size  
DRB1 = row 0 + row 1 size  
DRB2 = row 0 + row 1 + row 2 size  
DRB3 = row 0 + row 1 + row 2 size + row 3 size  
-----  
For each row: DRAW Row Attribute register (DRAR)  
size of page  
000 = 2K  
001 = 4K  
010 = 8K  
011 = 16K  
-----  
One DRAM Timing register  
DRAM Idle timer (how much idle DRAM cycles before precharge)  
000 = infinite  
001 = 0  
010 = 8 DRAM clock  
011 = 16 DRAM clock  
100 = 64 DRAM clock  
Activate to precharge delay (tRAS) MAX  
0 = 120 ns  
1 = reserved  
Activate to precharge delay (tRAS) MIN  
00 = 8 clocks  
01 = 7 clocks  
10 = 6 clocks  
11 = 5 clocks  
CAS# latency  
00 = 2.5 clocks  
01 = 2 clocks  
10 = 1.5 clocks  
11 = reserved  
RAS# to CAS# delay  
01 = 3 DRAM clocks  
10 = 2 DRAM clocks  
DRAM RAS# Precharge  
01 = 3 DRAM clocks  
10 = 2 DRAM clocks  
-----  
One DRAM Controller mode register  
Initialization complete  
Dynamic power-down mode enable  
Refresh mode  
Mode  
DRAM type (example: DDR)
```

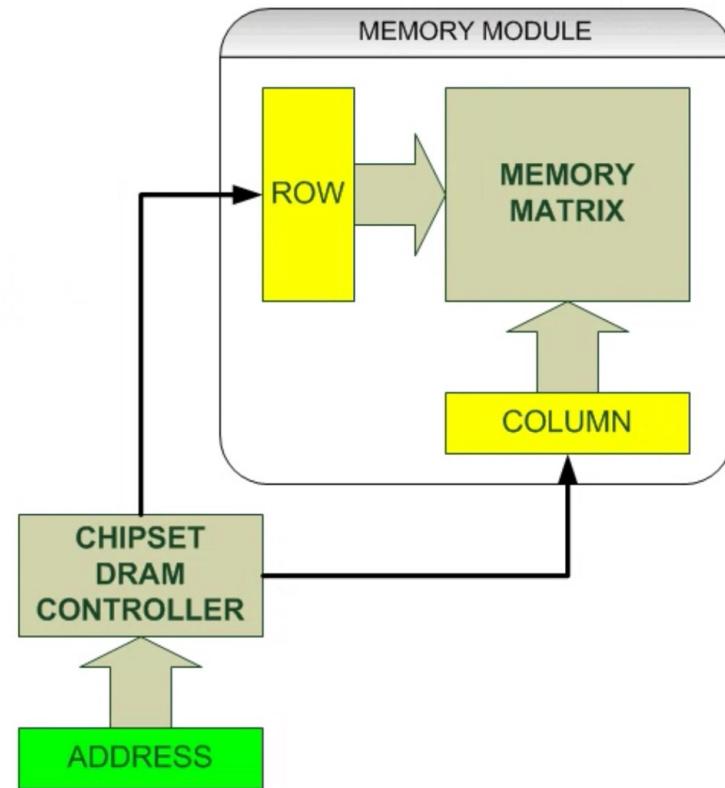
Компаратор для выбора типа памяти



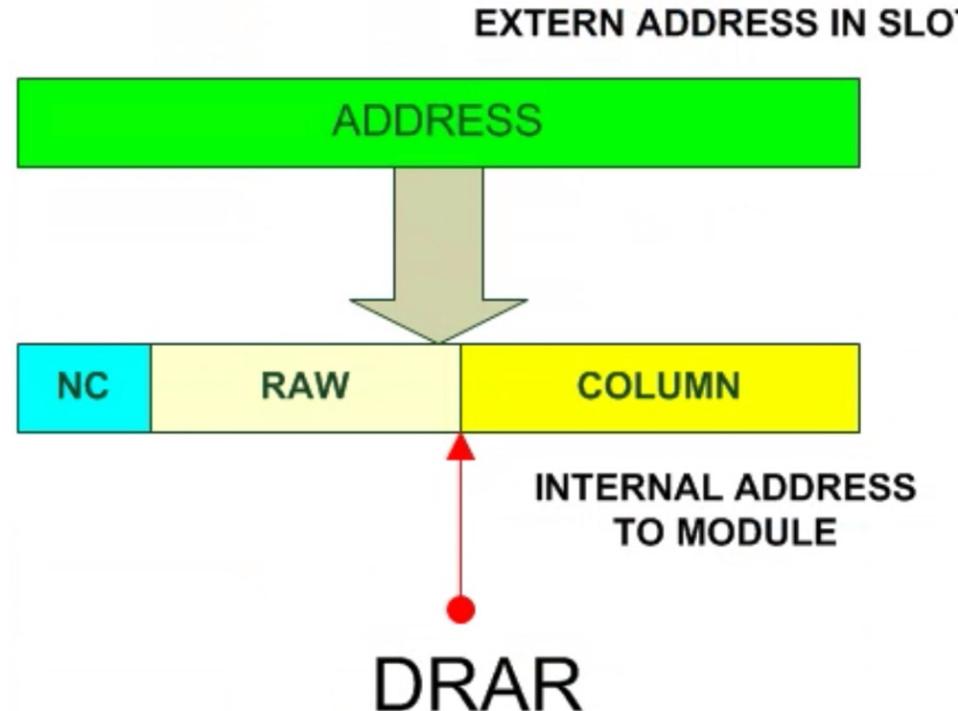
Сканирование памяти. Проверка паттернов



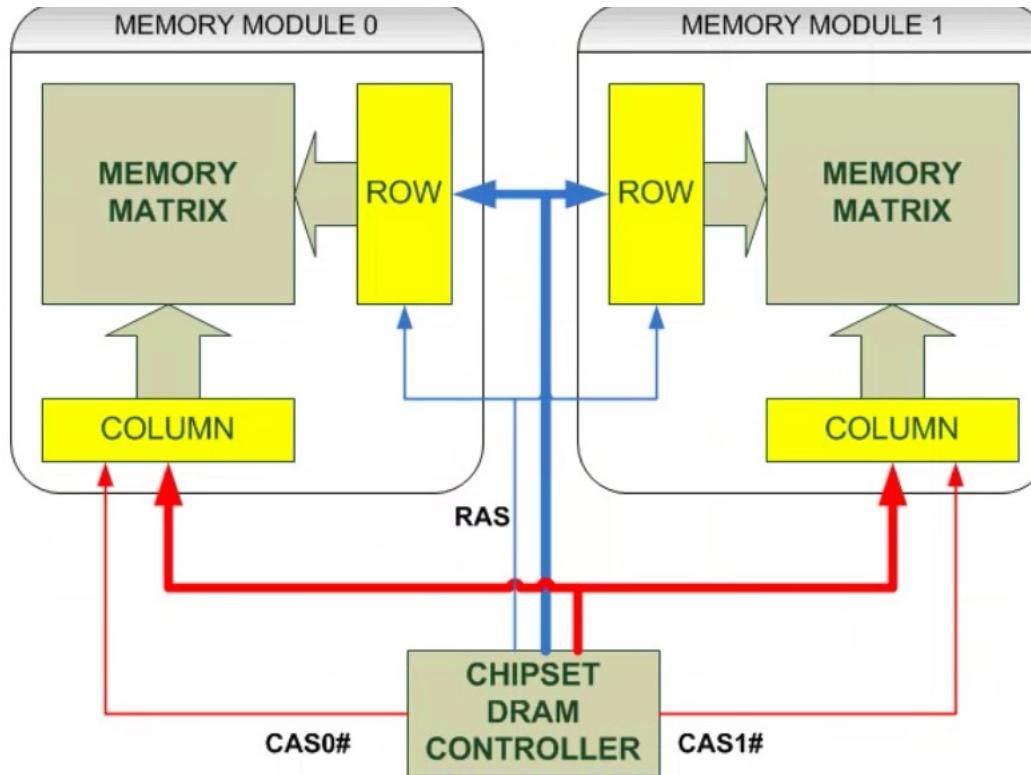
Организация памяти DRAM.



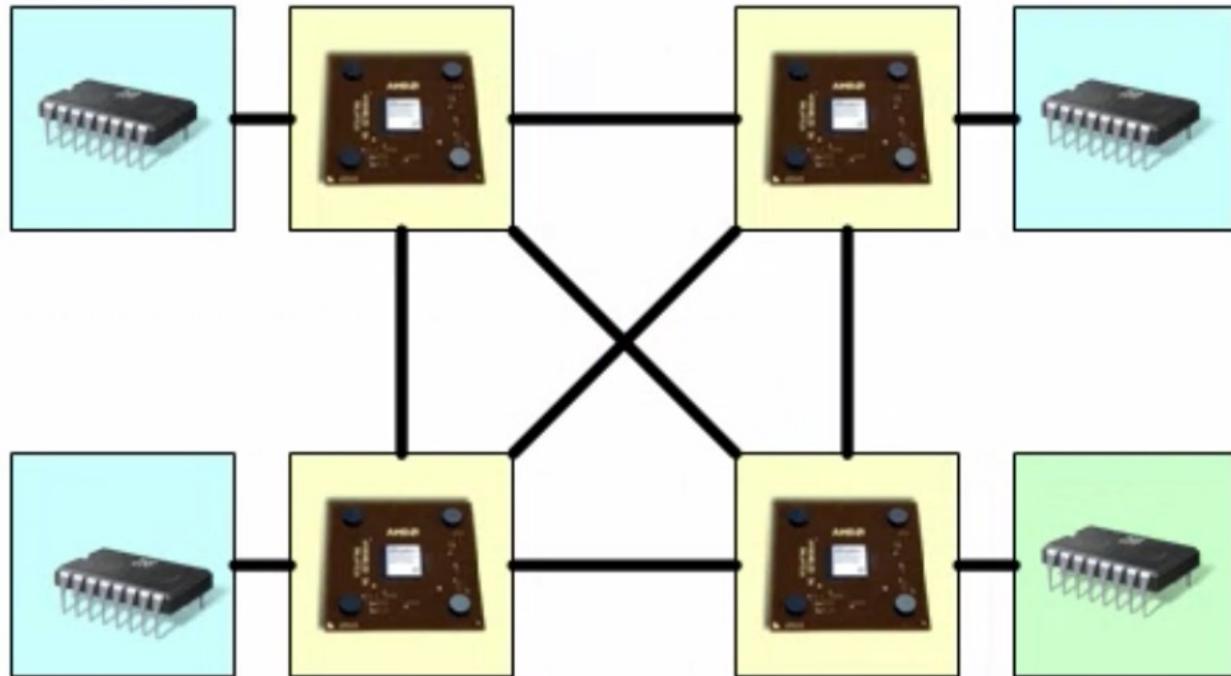
Отображение внешнего адреса на слот памяти



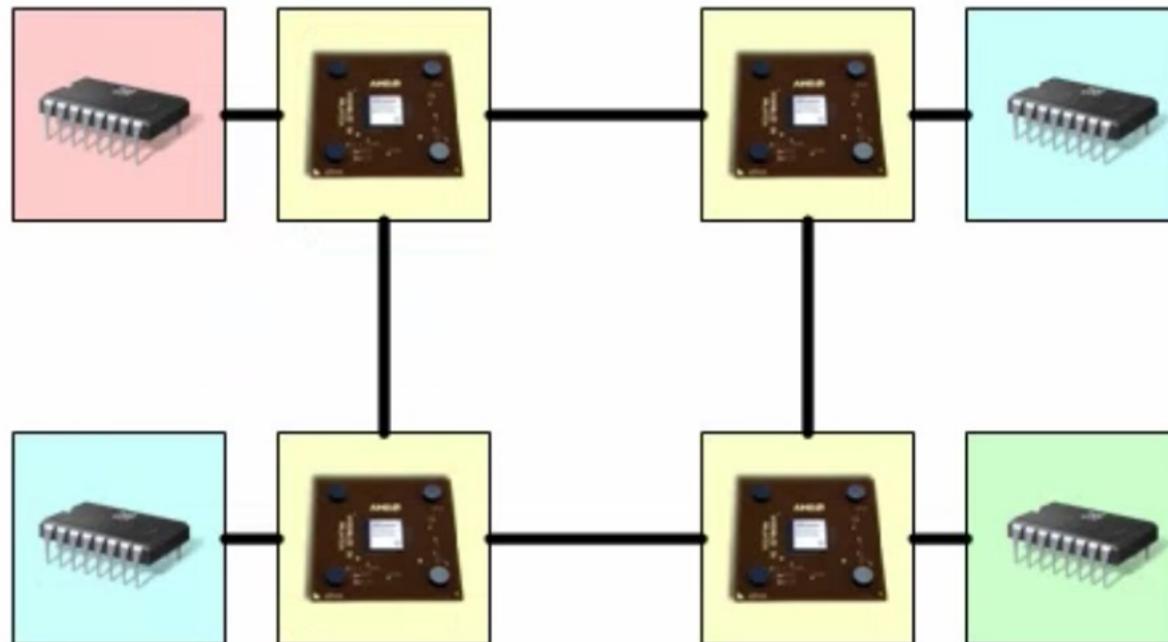
Двухканальный режим



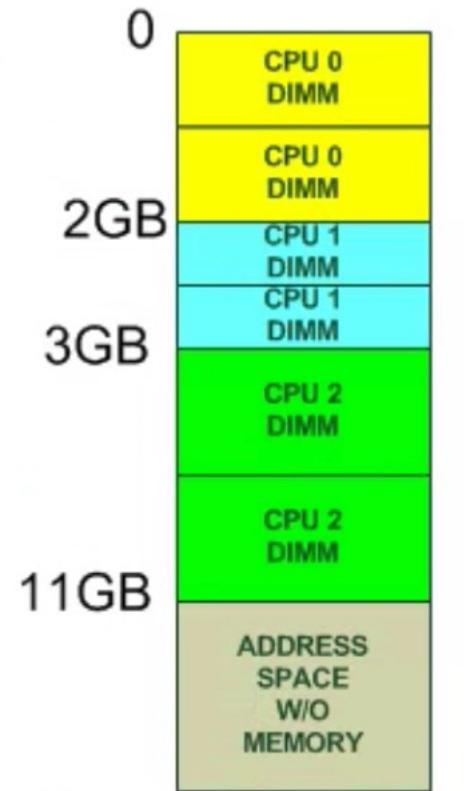
CCNUMA 4 node



ccNUMA 4-node 2 connect



Конфигурирование памяти



Доступные регионы памяти. BIOS vs EFI

```
typedef struct
{
    UINT32 Type;                                // EFI_MEMORY_TYPE
    EFI_PHYSICAL_ADDRESS PhysicalStart; // UINT64
    EFI_VIRTUAL_ADDRESS VirtualStart; // UINT64
    UINT64 NumberOfPages;
    UINT64 Attribute;
#define EFI_MEMORY_UC 0x0000000000000001
#define EFI_MEMORY_WC 0x0000000000000002
#define EFI_MEMORY_WT 0x0000000000000004
#define EFI_MEMORY_WB 0x0000000000000008
#define EFI_MEMORY_UCE 0x0000000000000010
#define EFI_MEMORY_WP 0x0000000000001000
#define EFI_MEMORY_RP 0x000000000002000
#define EFI_MEMORY_XP 0x000000000004000
#define EFI_MEMORY_RUNTIME 0x8000000000000000
} EFI_MEMORY_DESCRIPTOR;

typedef enum
{
    EfiReservedMemoryType,
    EfiLoaderCode,
    EfiLoaderData,
    EfiBootServicesCode,
    EfiBootServicesData,
    EfiRuntimeServicesCode,
    EfiRuntimeServicesData,
    EfiConventionalMemory,
    EfiUnusableMemory,
    EfiACPIReclaimMemory,
    EfiACPIMemoryNVS,
    EfiMemoryMappedIO,
    EfiMemoryMappedIOPortSpace,
    EfiPalCode,
} EFI_MEMORY_TYPE;
```

ACPI BIOS SRAT таблицы

```
+00  BYTE   Type (=1 Memory Affinity Structure)
+01  BYTE   Length (= 0x28)
+02  DWORD  Proximity Domain (ACPI meaning NUMA node id)
+06  WORD   Reserved
+08  DWORD  Base address of memory range (Low)
+0C  DWORD  Base address of memory range (High)
+10  DWORD  Length of memory range (Low)
+14  DWORD  Length of memory range (High)
+18  DWORD  Reserved
+1C  DWORD  Flags
          bit 0  valid entry
          bit 1  hot pluggable
          bit 2  non volatile
+20  QWORD  Reserved
```

```
***** ACPI SRAT table *****
00000000 : dump (0004)  SRAT signature
00000000 : 53 52 41 54 [SRAT-----]
00000004 : dword 00001158 Length of table
00000008 : byte 03 Revision
00000009 : byte 4A Checksum
0000000A : dump (0006) OEM ID
0000000A : 41 4C 41 53 4B 41 [ALASKA-----]
00000010 : dump (0008) Manufacturer mode ID
00000010 : 41 20 4D 20 49 20 00 00 [A M I ..-----]
00000018 : dword 00000001 OEM revision of table
0000001C : dword 4C544E49 OEM ID of utility that created table
00000020 : dword 20091013 Revision of utility
00000024 : dump (000C) Reserved
00000024 : 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [.....----]
----- SRAT Entry -----
00000030 : byte 00 Type
          * Local APIC Affinity
00000031 : byte 10 Length
00000032 : byte 00 Proximity domain [7:0]
00000033 : byte 00 Local APIC ID
00000034 : dword 00000001 Flags
          [00] * Enabled
00000038 : byte 00 Local SAPIC EID
00000039 : dump (0003) Proximity domain [31:8]
00000039 : 00 00 00 [.....-----]
0000003C : dword 00000000 Clock domain

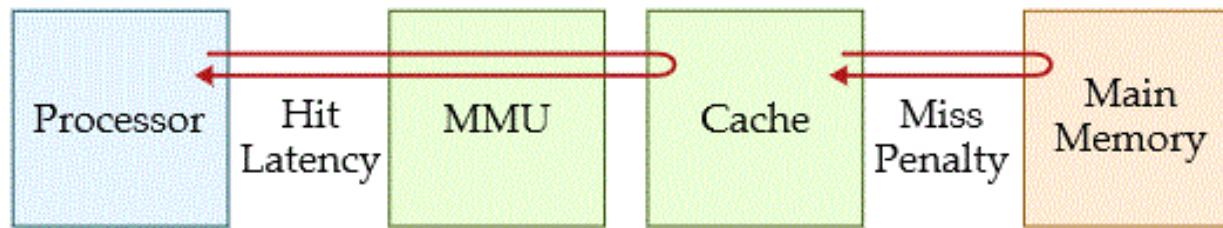
..... показано по 1му процессору с каждой Node

----- SRAT Entry -----
000000D0 : byte 00 Type
          * Local APIC Affinity
000000D1 : byte 10 Length
```

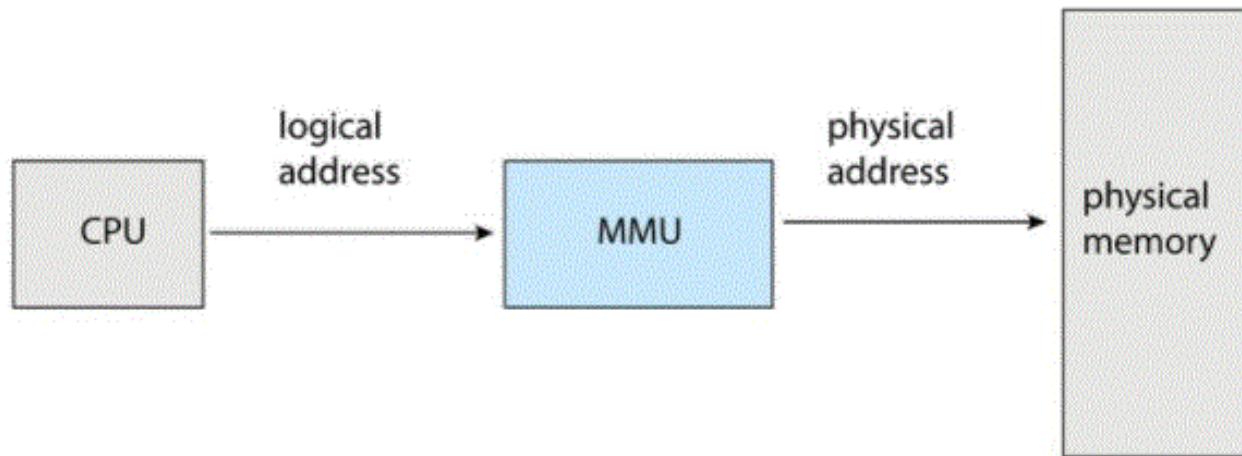
ACPI SLIT

```
$ numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 24 25 26 27 28 29 30 31 32 33 34 35
node 0 size: 31796 MB
node 0 free: 30999 MB
node 1 cpus: 12 13 14 15 16 17 18 19 20 21 22 23 36 37 38 39 40 41 42 43 44 45 46
node 1 size: 32248 MB
node 1 free: 31946 MB
node distances:
node    0    1
 0:  10   21
 1:   21   10
```

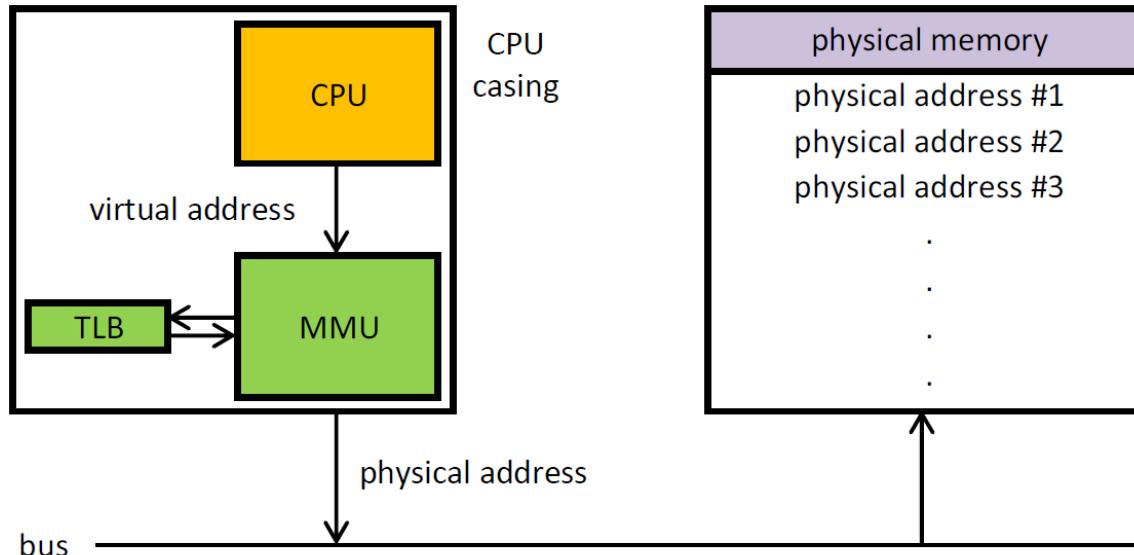
Адресное пространство и виртуальная память.



Memory management unit



MMU свойства и задачи

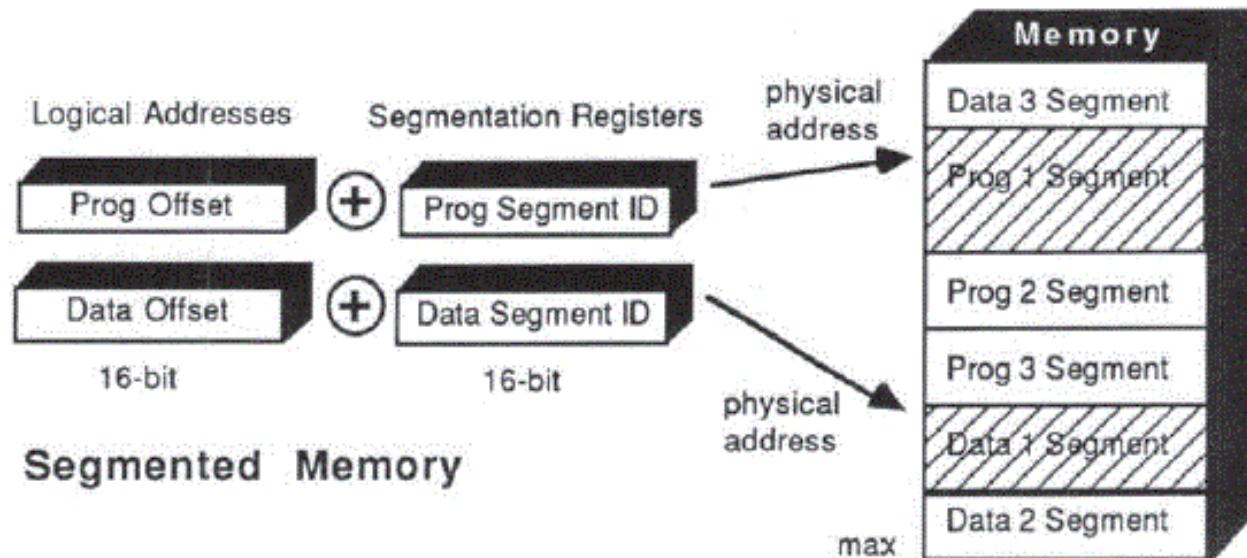


CPU: Central Processing Unit

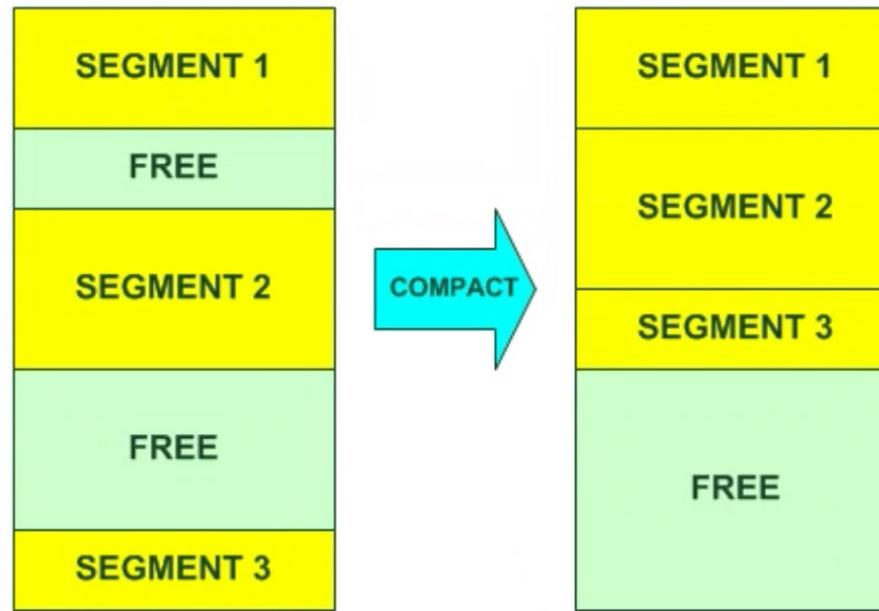
MMU: Memory Management Unit

TLB: Translation lookaside buffer

Виртуальная сегментная память

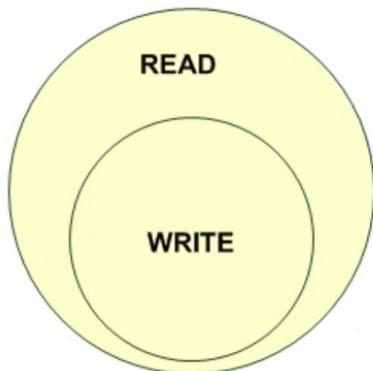


Зачем нужна сегментизация АП

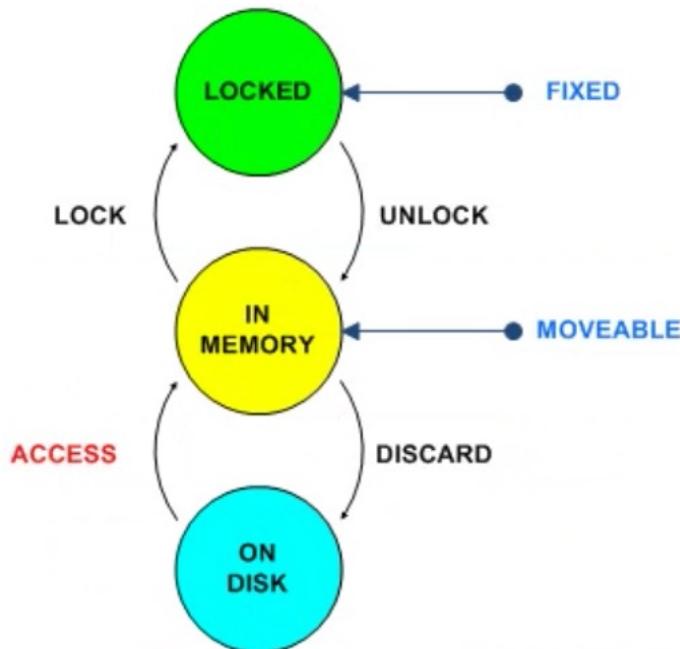
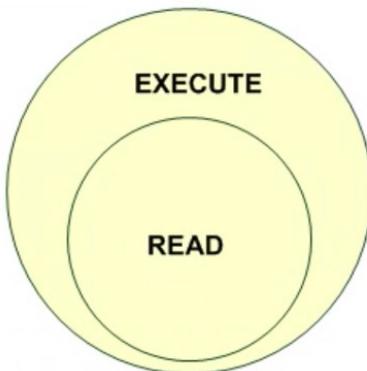


Свойства сегмента

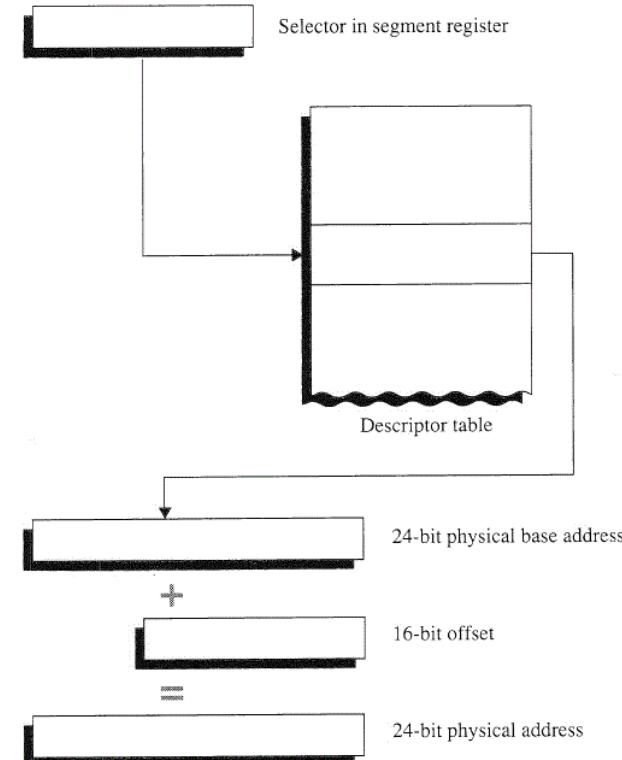
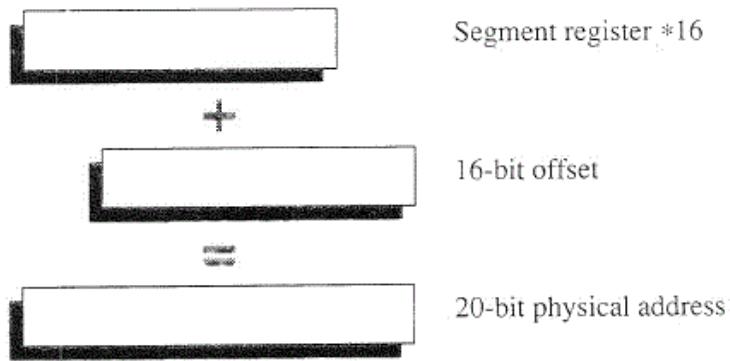
DATA
SEGMENT



CODE
SEGMENT

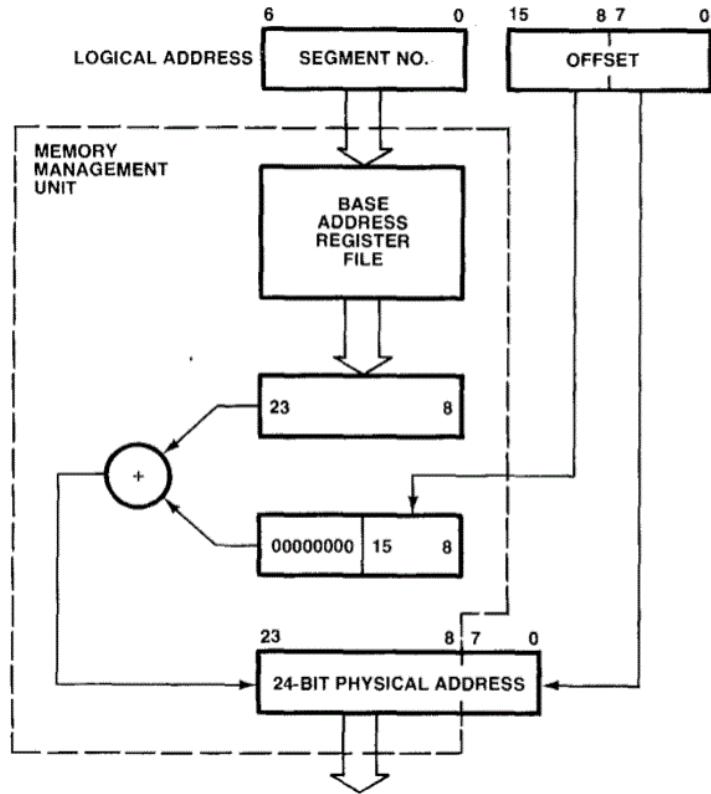


Сегментная трансляция

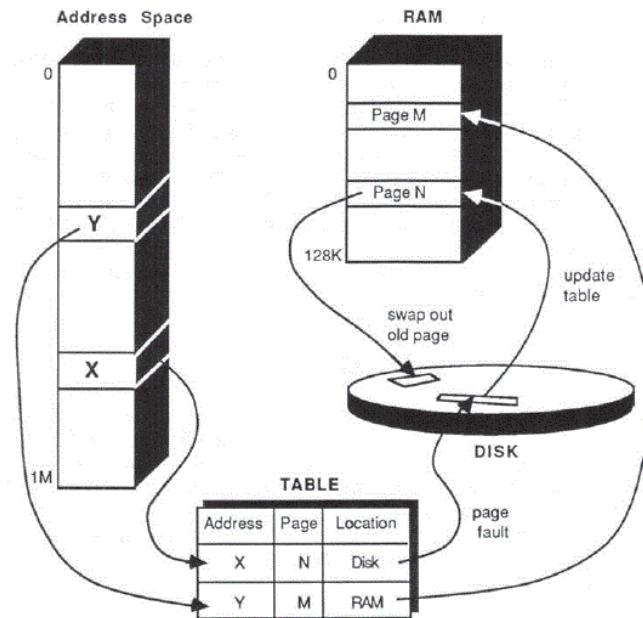
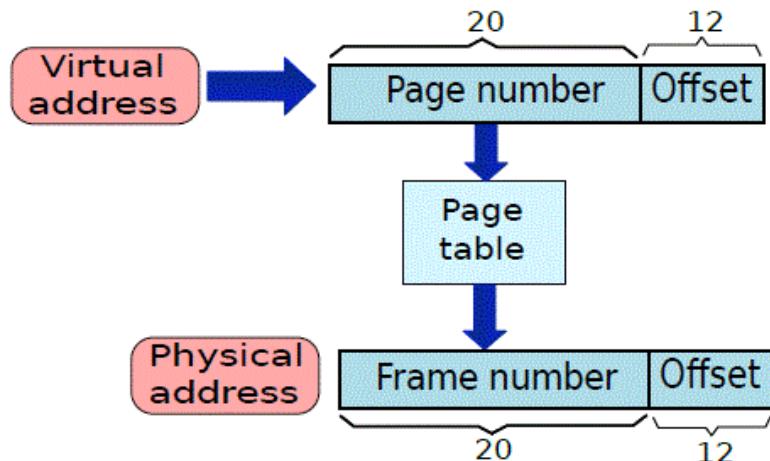




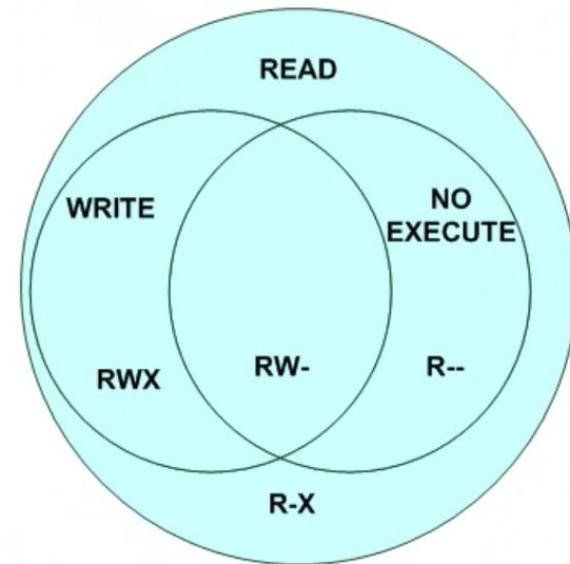
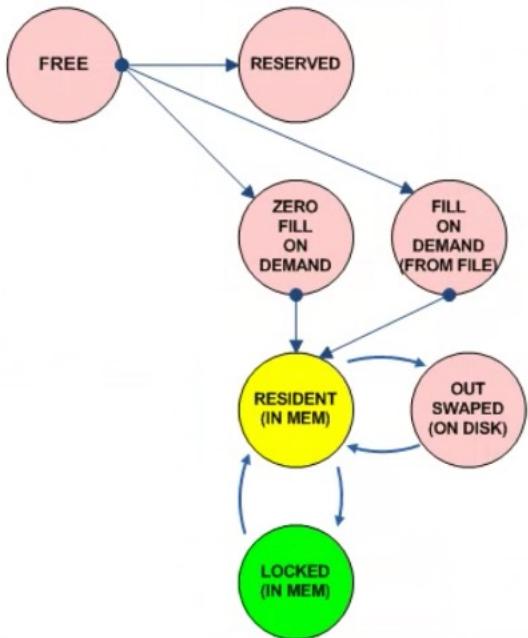
Примеры сегментной трансляции z8000



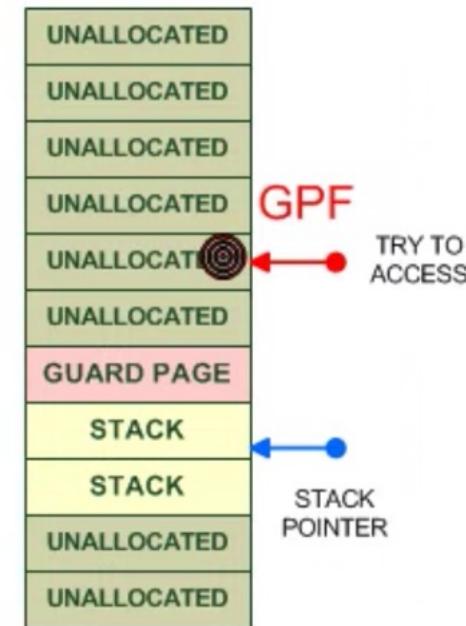
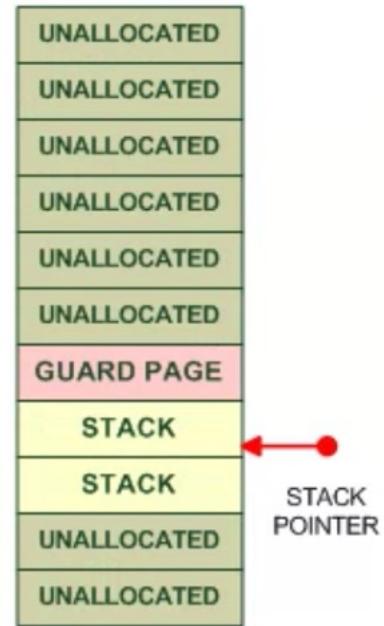
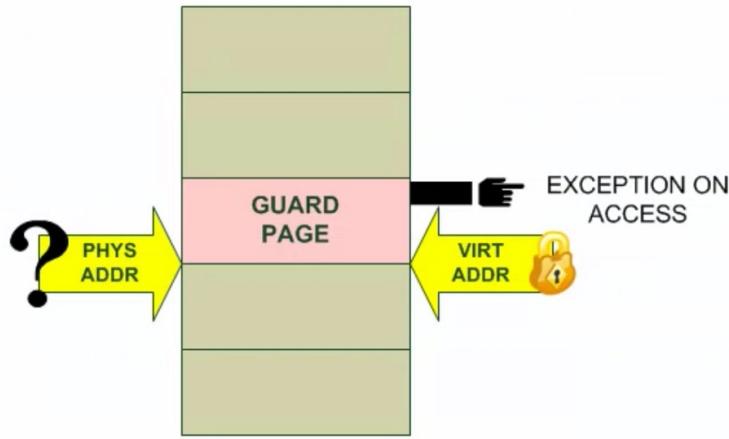
Виртуальная страницчная память



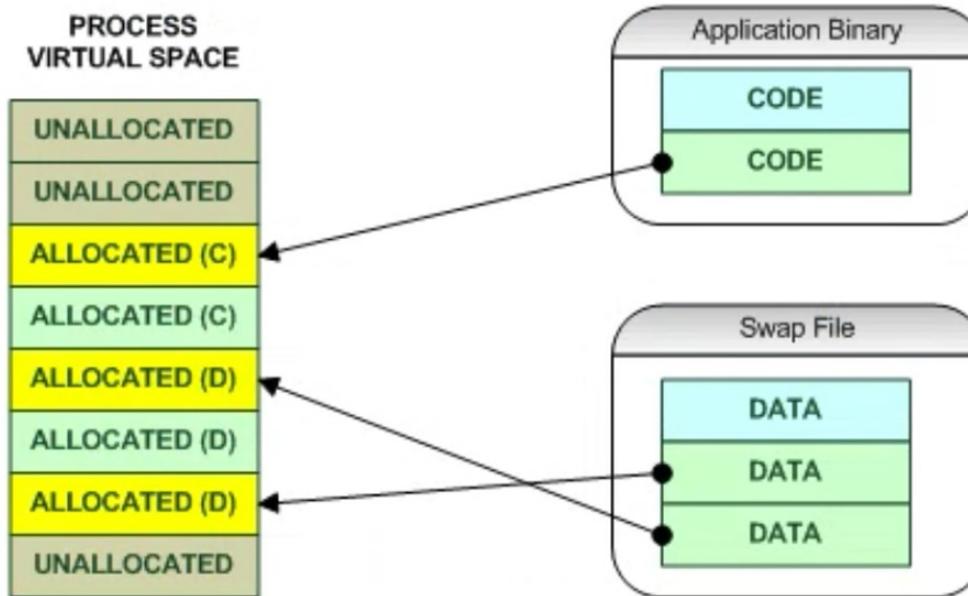
Свойства страниц



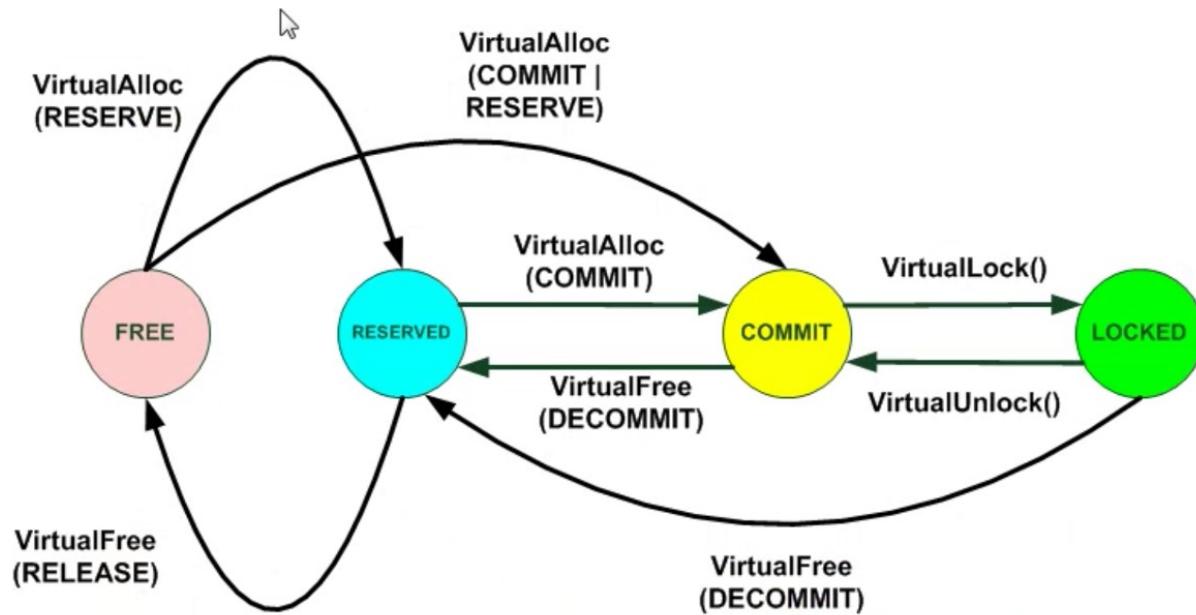
Stack guard и guard page



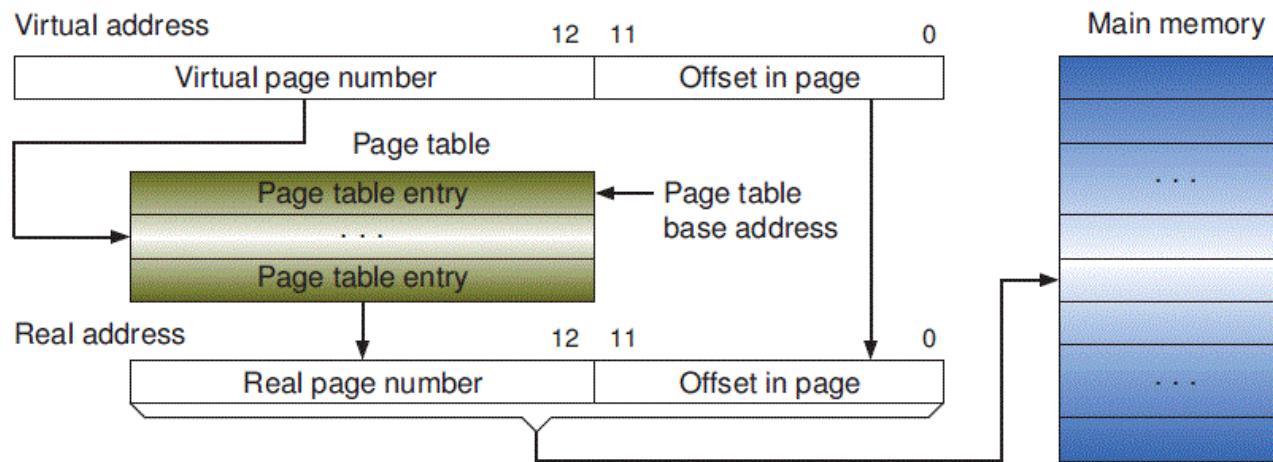
Резервирование памяти



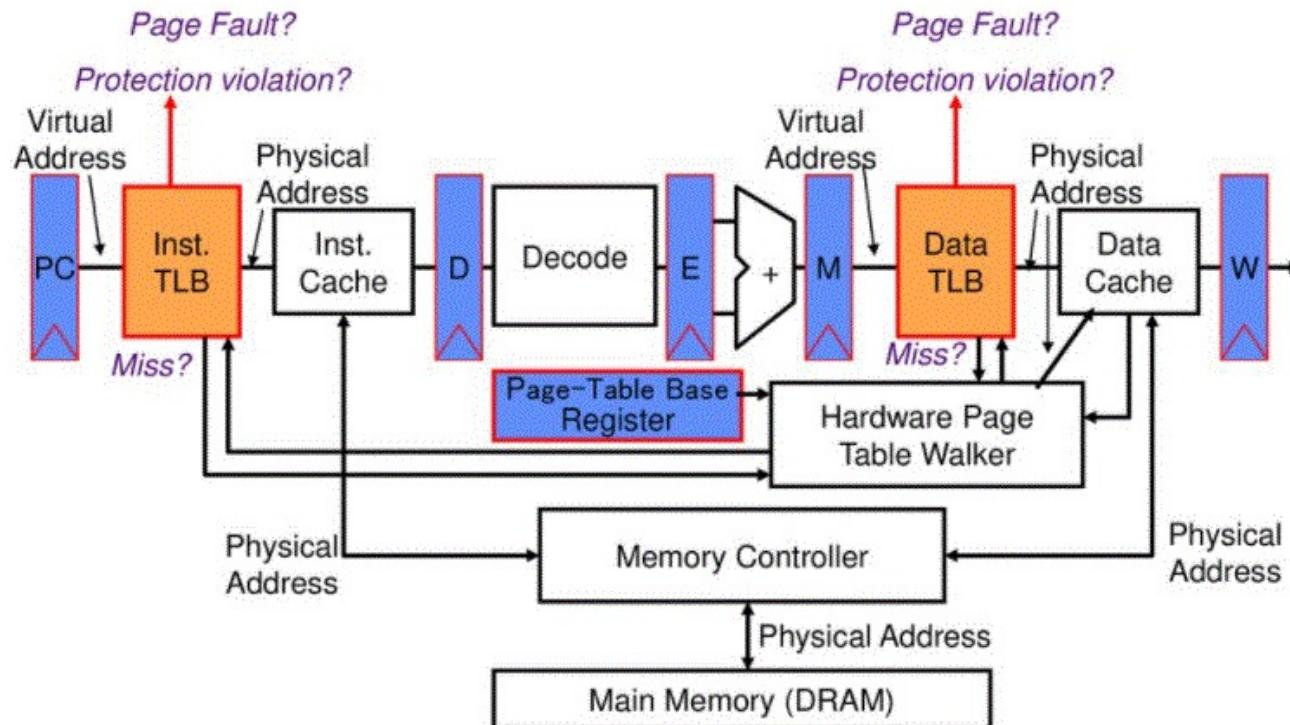
Механики обращения к страницной памяти



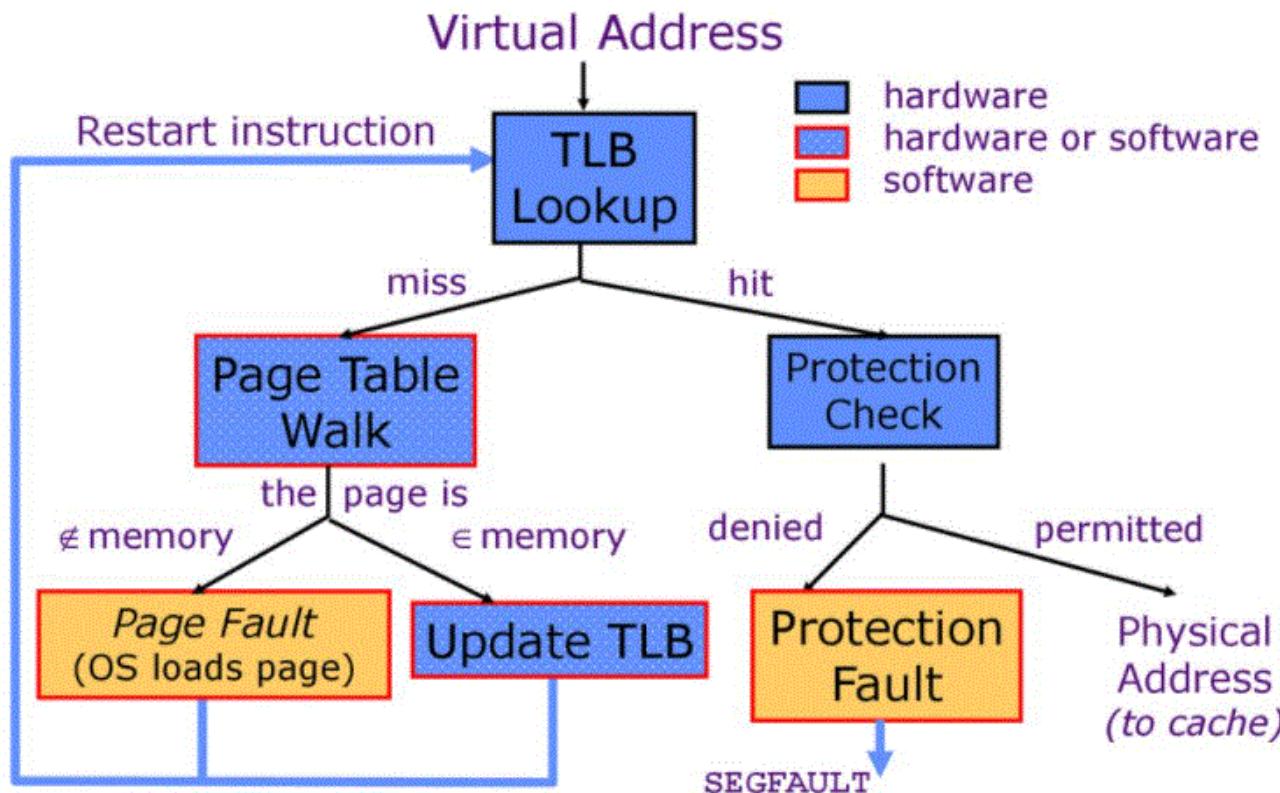
Страницчная трансляция



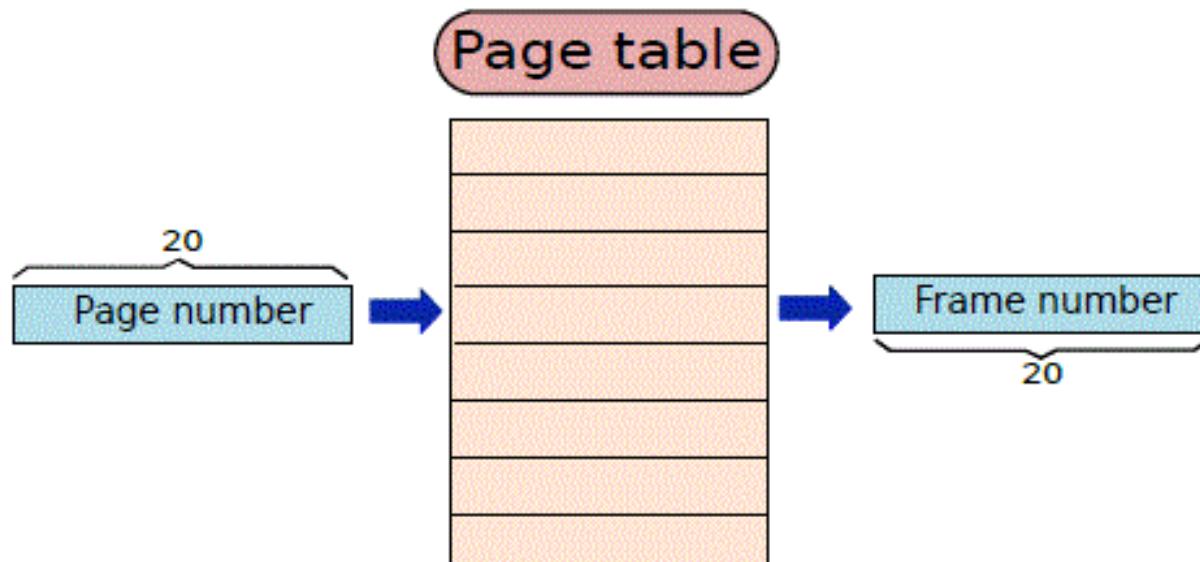
Аппаратная поддержка трансляции



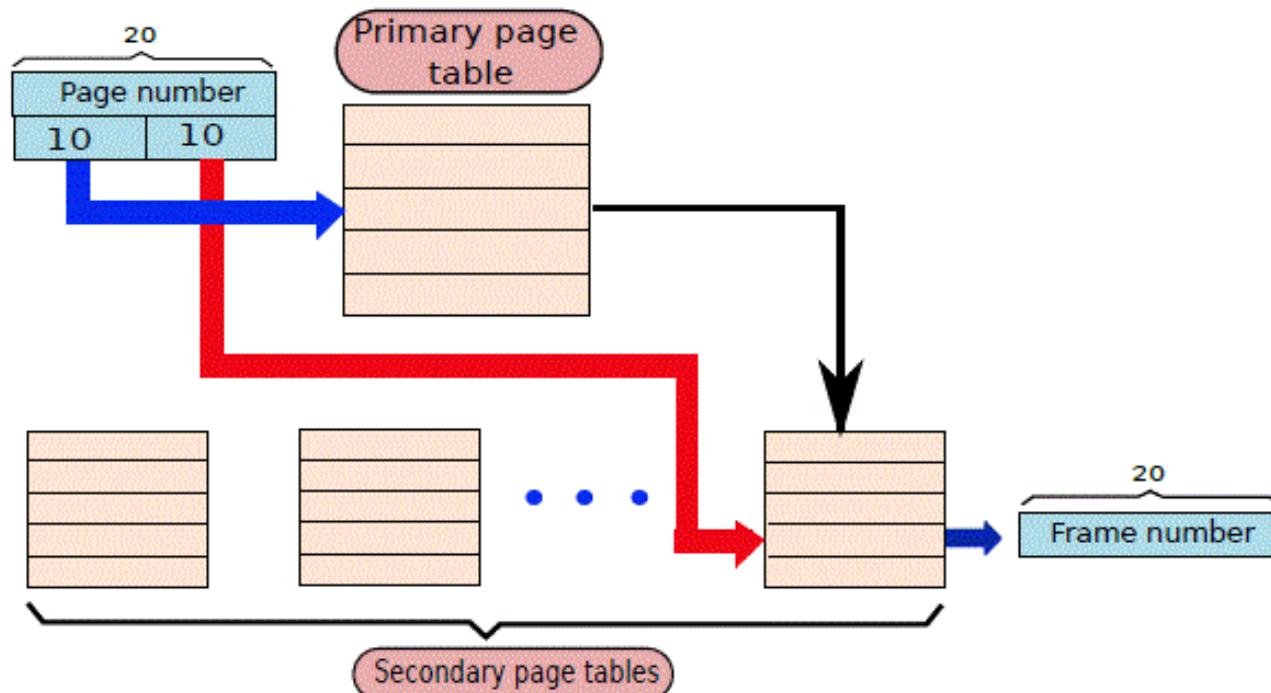
Программная поддержка трансляции



Уровни таблиц трансляции



Уровни таблиц трансляции



Примеры размера страниц ARM



Fault	31	24 23	20 19	14	12 11 10 9 8	5 4 3 2 1 0	
				IGN			0 0
Page table				Coarse page table base address	P	Domain	SBZ 0 1
Section			Section base address	S B Z 0 n G S AP X	TEX AP P	Domain X N C B 1 0	

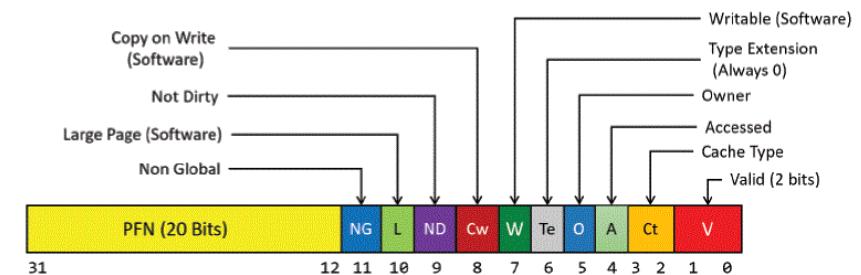
(a) Alternative first-level descriptor formats

Fault	31	16 15 14	12 11 10 9 8 7 6 5 4 3 2 1 0	
		IGN		0 0
Small page		Small page base address	n G S AP X	TEX AP C B 1 X N
Large page		Large page base address	X N TEX n G S AP X	SBZ AP C B 0 1

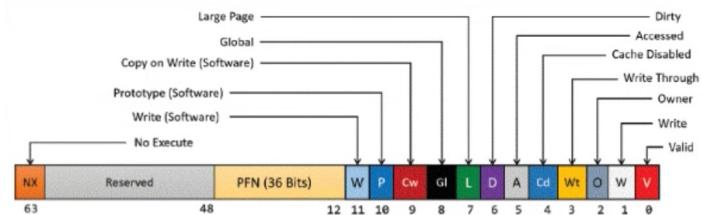
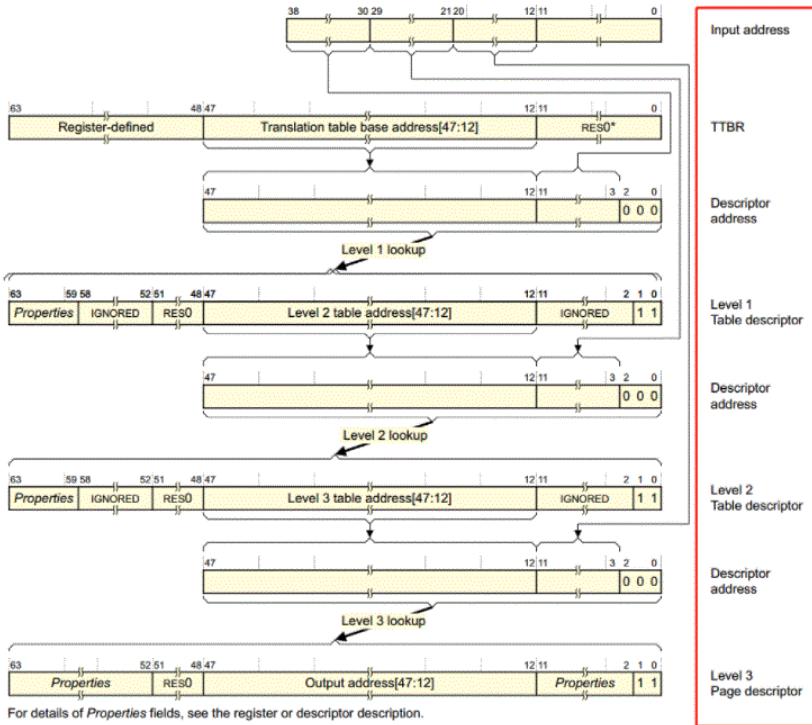
(b) Alternative second-level descriptor formats

Supersection	31	24 23	20 19	0
		Level 1 table index	Supersection index	
Section	31	20 19		0
		Level 1 table index	Section index	
Small page	31	20 19	12 11	0
		Level 1 table index	Level 2 table index	Page index
Large page	31	20 19	16 15	0
		Level 1 table index	Level 2 table index	Page index

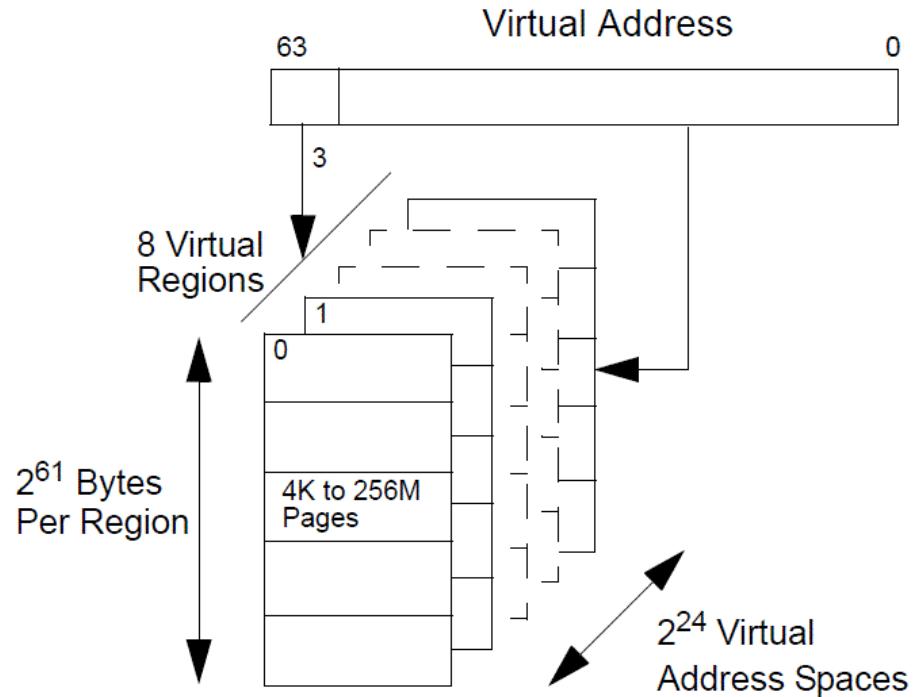
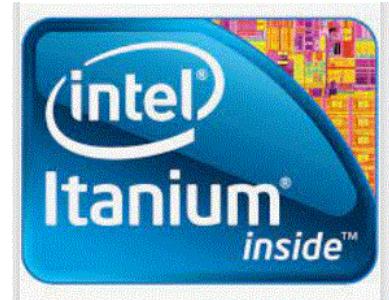
(c) Virtual memory address formats



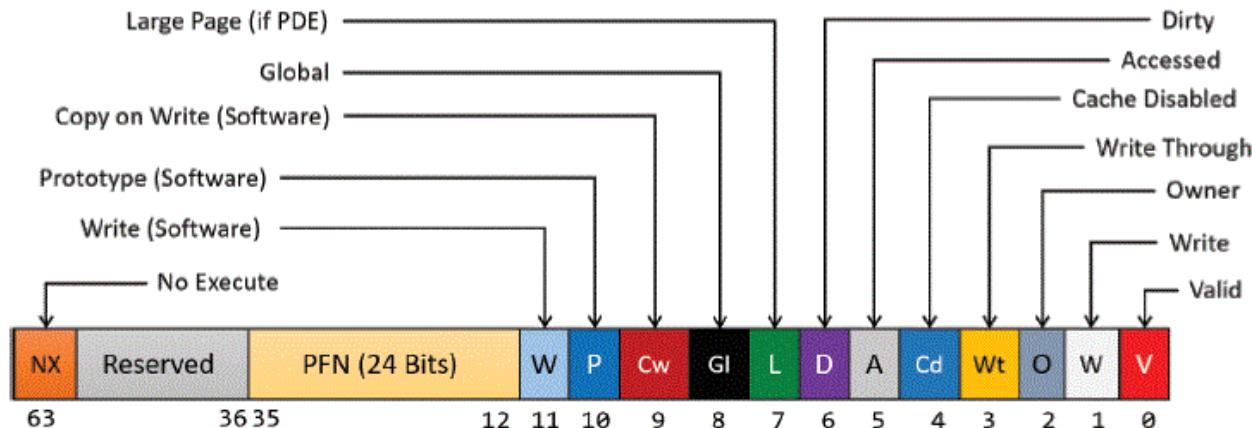
Примеры размера страниц ARM64



Примеры размера страниц IA-64



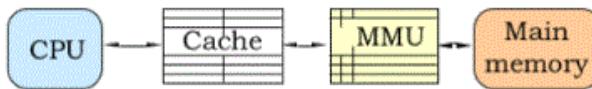
Примеры размера страниц x64



Virtual addressed cache vs Physically addressed cache

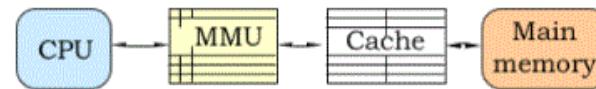
Virtually-Addressed Cache

Tags from virtual addresses



Physically-Addressed Cache

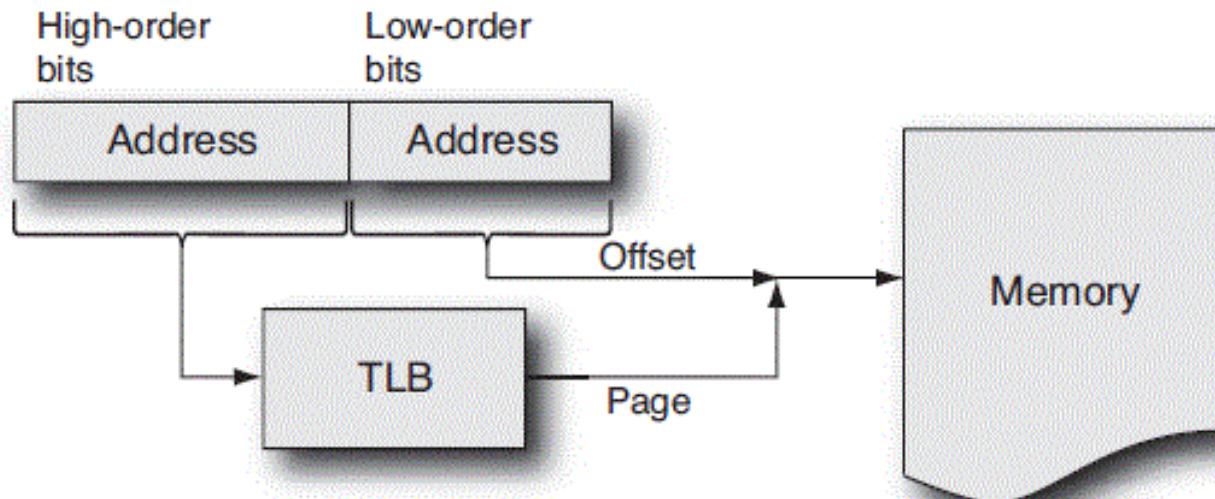
Tags from physical addresses



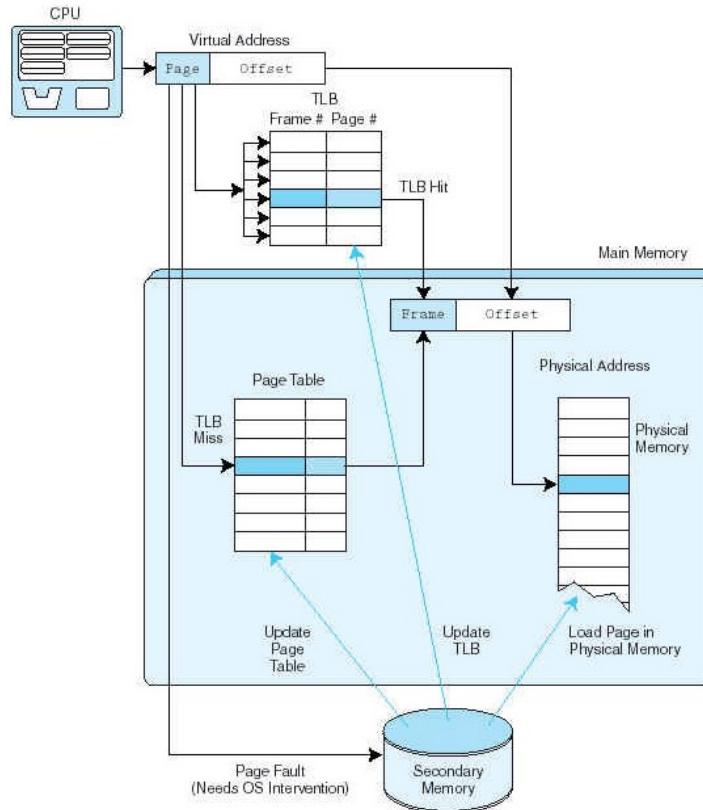
- FAST: No MMU time on HIT
- Problem: Must flush cache after context switch

- Avoids stale cache data after context switch
- SLOW: MMU time on HIT

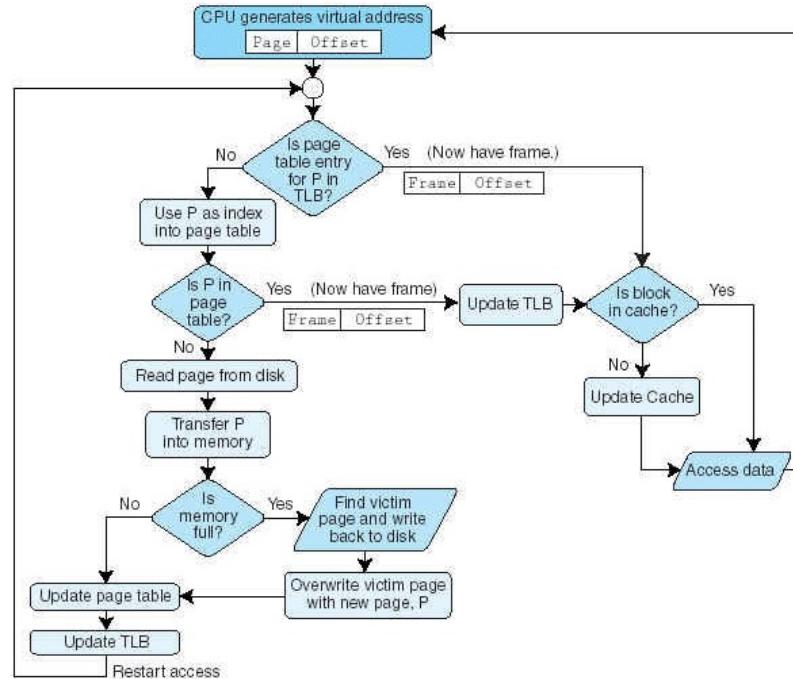
TLB - translation lookaside buffer



TLB - translation lookaside buffer



TLB - translation lookaside buffer



Основные memory API OS