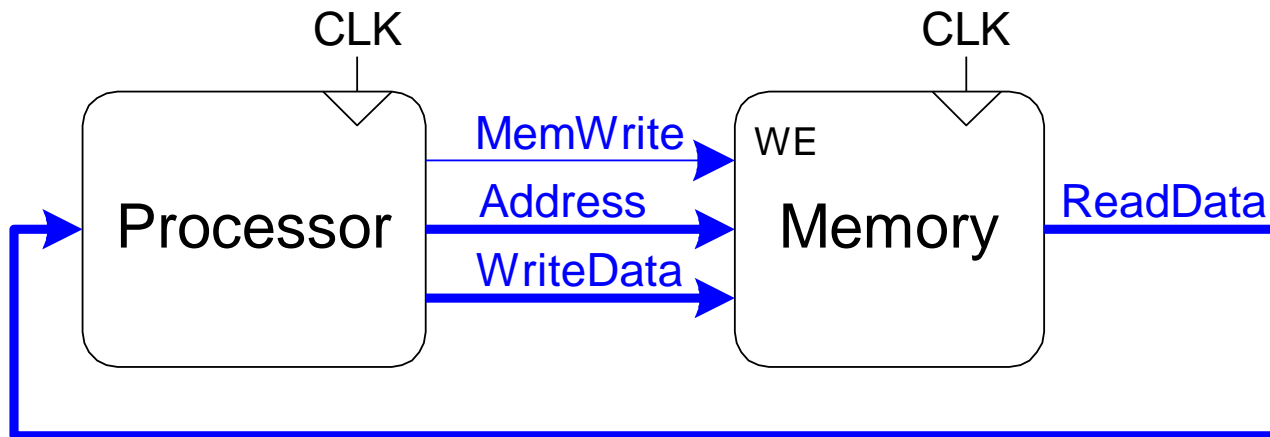


КАРХ: Тема_12: Памет и входно-изходна система

Въведение.

- Производителността на компютъра, освен от качеството на използвания процесор, зависи силно от системната памет и Входно-изходните устройства (input-output (I/O) devices), като монитор, клавиатура, принтер и др.
- Най-значимо е влиянието на *организацията на паметта*.

Интерфейс на паметта (Memory Interface)

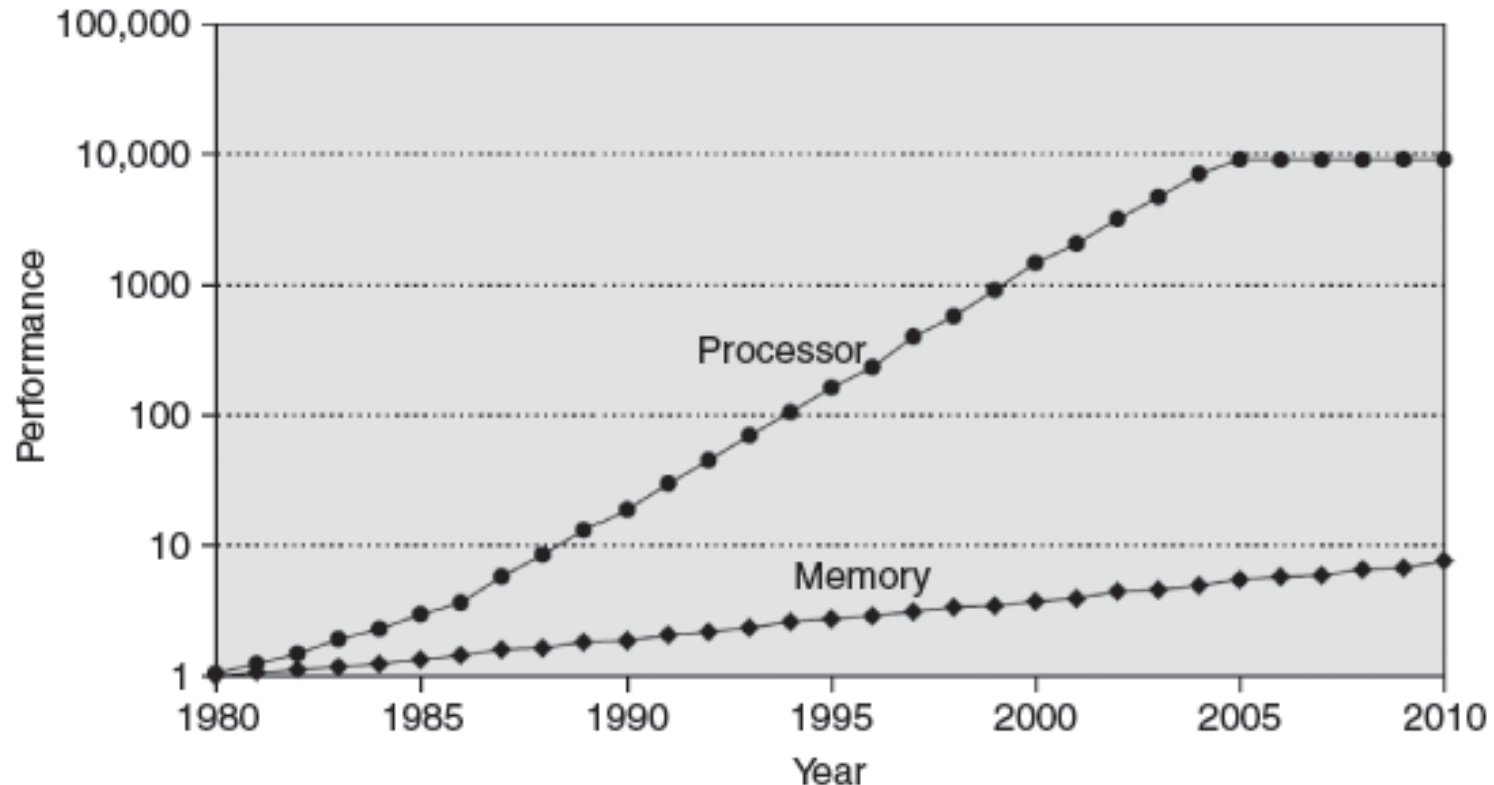


Application Software	
Operating Systems	
Architecture	
Micro-architecture	
Logic	
Digital Circuits	
Analog Circuits	
Devices	
Physics	

КАРХ: Тема_12: Памет и входно-изходна система

Процесор - памет.

- В досегашните разглеждания предполагахме, че достапът до паметта става за 1 clock cycle – това **не е вярно** още от 80-те години на миналия век!
- Развитието на паметите по производителност сериозно **изостава** спрямо това на процесорите.



КАРХ: Тема_12: Памет и входно-изходна система

Предизвикателства пред паметта.

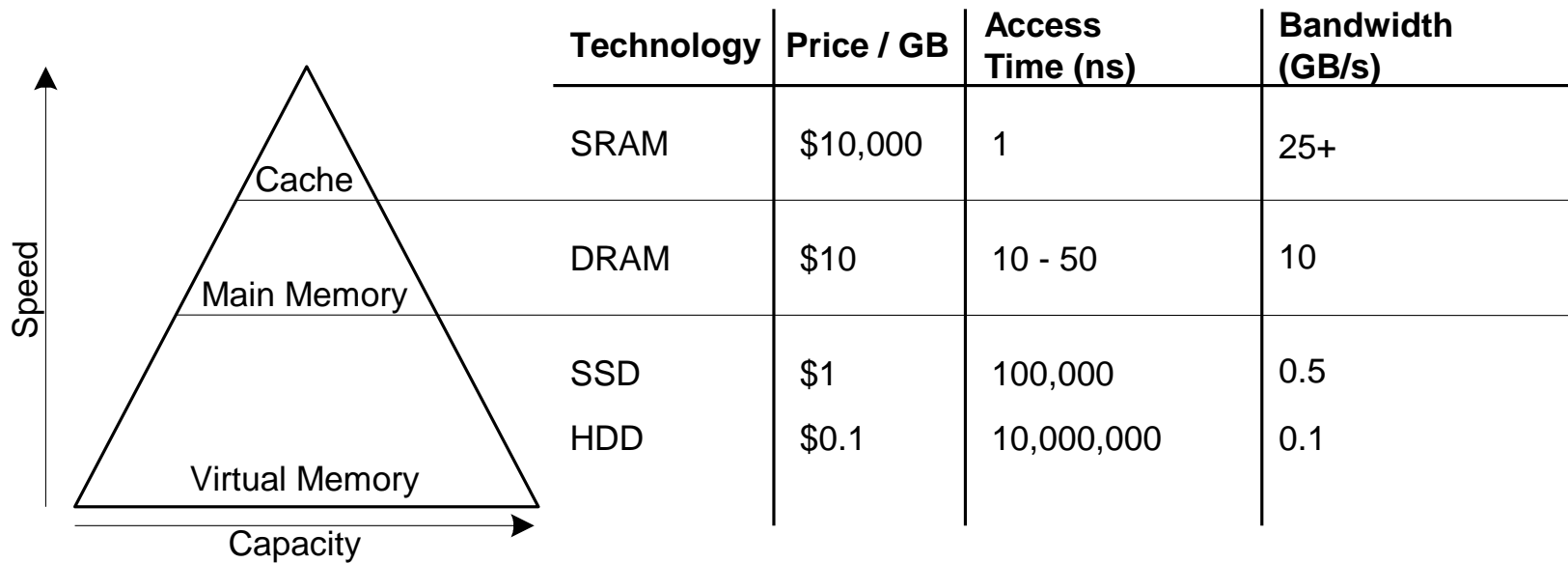
- Системната памет да е толкова бърза, колкото използвания процесор.
- Използване йерархия на използваните видове памет.
- **Идеалната памет е:**
 - *Бърза* (Fast)
 - *Евтина* (Cheap (inexpensive))
 - *Голяма* (Large (capacity))
- **Но можем да изберем само две от трите!**



КАРХ: Тема_12: Памет и входно-изходна система

Йерархия на паметта.

- Статична памет (**SRAM**)
- Динамична памет (**DRAM**)
- Твърдотелен диск (**Solid-State Disk**)
- Магнитен диск (**Hard Disk Drive**)



КАРХ: Тема_12: Памет и входно-изходна система

Локалност.

Използване на локалност за бърз достъп до паметта

- **Времева локалност (Temporal Locality):**
 - Локалност във времето;
 - Ако дадени данни са използвани веднъж, много е вероятно да се използват скоро пак;
 - **Начин на употреба:** задържане на скоро използвани данни във високите йерархични нива на паметта.
- **Пространствена локалност (Spatial Locality):**
 - Локалност в пространството;
 - Ако дадени данни са използвани скоро, много е вероятно да се използват скоро и данните около тях;
 - **Начин на употреба:** при осигуряване на достъп до дадени данни, да се вземат (пренасят) и техните съседи в по-високите йерархични нива на паметта.

КАРХ: Тема_12: Памет и входно-изходна система

Оценка на производителността на паметта.

- **Попадение (Hit):** намерени данни в дадено йерархично ниво на паметта
- **Пропуск (Miss):** ненамерени данни (трябва да се слезе на по-ниско ниво)
- **Степен на попаденията (Hit Rate)** = брой попадения (# hits) / общ брой обръщания към паметта (# memory accesses)
= 1 – Miss Rate
- **Степен на пропуските (Miss Rate)** = брой пропуски (# misses) / общ брой обръщания към паметта (# memory accesses)
= 1 – Hit Rate
- **Средно време за достъп до паметта (Average memory access time (AMAT)):** това е средното време, за което процесорът чете/записва данни от/в паметта.
- **AMAT** = $t_{\text{cache}} + MR_{\text{cache}}[t_{MM} + MR_{MM}(t_{VM})]$, където t_{cache} , t_{MM} , t_{VM} са времената на достъп съответно до кеша (cache), основната памет (main memory) и виртуалната памет (virtual memory), а MR_{cache} и MR_{MM} са съответните степени на пропуските.

КАРХ: Тема_12: Памет и входно-изходна система

Пример 1.

- Програма прави 2,000 обръщания към паметта.
- 1,250 от данните се намират в кеша.
- Останалите се намират в други нива на паметта.
- **Каква е степента на попадения (пропуски) на кеша?**

КАРХ: Тема_12: Памет и входно-изходна система

Пример 1.

- Програма прави 2,000 обръщания към паметта.
- 1,250 от данните се намират в кеша.
- Останалите се намират в други нива на паметта.
- **Каква е степента на попадения (пропуски) на кеша?**
- **Hit Rate = $1250/2000 = 0.625$**
- **Miss Rate = $750/2000 = 0.375 = 1 - \text{Hit Rate}$**

КАРХ: Тема_12: Памет и входно-изходна система

Пример 2.

- Нека паметта има две нива на йерархия: кеш (cache) и основна памет (main memory), като времената им за достъп са съответно:
- $t_{\text{cache}} = 1 \text{ cycle}$, $t_{MM} = 100 \text{ cycles}$
- **Колко е АМАТ за програмата от Пример 1?**

КАРХ: Тема_12: Памет и входно-изходна система

Пример 2.

- Нека паметта има две нива на йерархия: кеш (cache) и основна памет (main memory), като времената им за достъп са съответно:
- $t_{\text{cache}} = 1 \text{ cycle}$, $t_{MM} = 100 \text{ cycles}$
- **Колко е АМАТ за програмата от Пример 1?**

- $$\text{АМАТ} = t_{\text{cache}} + MR_{\text{cache}}(t_{MM})$$
- $$= [1 + 0.375(100)] \text{ cycles}$$
- $$= \mathbf{38.5 \text{ cycles}}$$

КАРХ: Тема_12: Памет и входно-изходна система

Закон на Амдал (Amdahl's Law).

- **Закон на Амдал :** „Усилията хвърлени за подобряване производителността на една подсистема са безсмислени, освен ако подсистемата влияе на голям процент от общата производителност на цялата система.“ (1965)
- Ph.D по теоретична физика (1952).
- Работил за IBM.
- Съосновател на 3 компании (вкл. на Amdahl Corporation в 1970).

Gene Amdahl, 1922 -



КАРХ: Тема_12: Памет и входно-изходна система

Кеш памет (Cache).

- Най-високото ниво в йерархията на паметта.
- Бърза памет (типично ~ 1 cycle време на достъп).
- Теоретически предоставя повечето данни на процесора.
- Обикновени съдържа данни, които наскоро са били достъпни.
- Възникват следните въпроси:
 - Какви данни се държат в кеш паметта?
 - Как се намират данните в кеша?
 - Как се заместват данните в кеш паметта?
- (Фокусът е върху четенето на данни, но и записът следва същите принципи).

КАРХ: Тема_12: Памет и входно-изходна система

Кеш памет (Cache).

- Какви данни се държат в кеш паметта?
- В идеалния случай, кеша „предугажда“ нужните данни и ги зарежда в себе си.
- Но проблемът е в това, че не може да се предскаже бъдещето (данните, които ще се използват).
- Използване на миналото за предсказване на бъдещето – чрез времевата и пространствената локалност (temporal and spatial locality):
 - **Времева локалност (Temporal locality):** копиране на новоизползваните данни в кеша
 - **Пространствена локалност (Spatial locality):** копиране и на техните съседи в кеша

КАРХ: Тема_12: Памет и входно-изходна система

Кеш терминология (Cache Terminology).

- **Капацитет (Capacity (C)):**
 - Броят на байтовете в кеша
- **Размер на блок (Block size (b)):**
 - Брой на байтовете, които могат да се запишат едновременно (наведнъж) в кеша
- **Брой на блоковете (Number of blocks ($B = C/b$)):**
 - Броят на блоковете в кеша: $B = C/b$
- **Степен на асоциативност (Degree of associativity (N)):**
 - Брой на блоковете в ред (сет)
- **Брой на редовете (сетовете) (Number of sets ($S = B/N$)):**
 - Всеки адрес от паметта се нанася (maps) в точно един ред (сет) на кеша

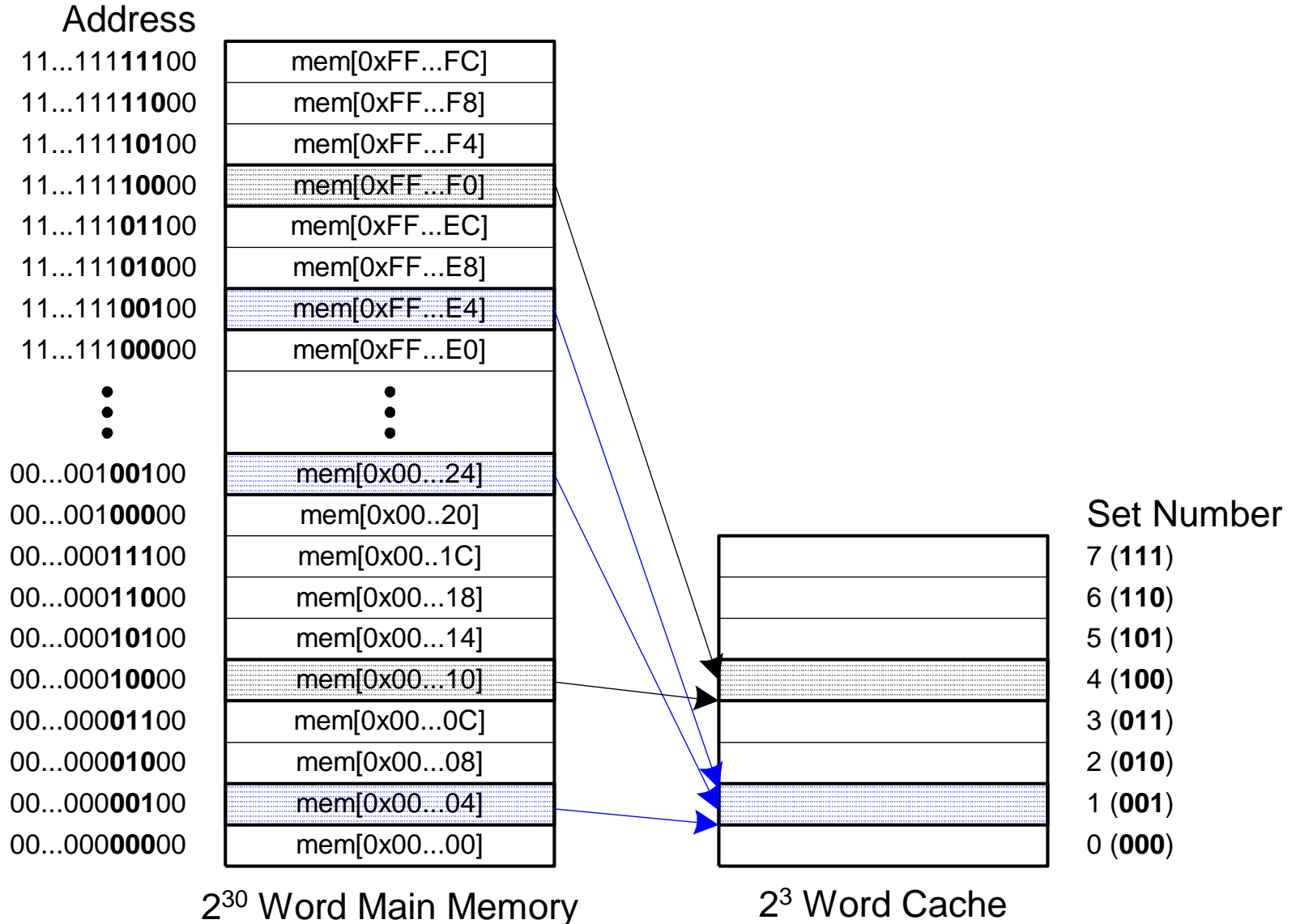
КАРХ: Тема_12: Памет и входно-изходна система

Организация на данните в кеша.

- Кешът е организиран в S реда (сета)
- Всеки адрес от паметта се нанася (maps) в точно един ред (сет) на кеша
- Кешът се категоризира според броя на блоковете в един ред (сет):
 - **Direct mapped:** 1 блок на ред (block per set)
 - **N -way set associative:** N блока на ред (blocks per set)
 - **Fully associative:** всички блокове на кеша в един ред (all cache blocks in 1 set)
- Ще разгледаме всяка от тези организации за кеш памет със следните параметри:
 - Капацитет ($C = 8$ думи)
 - Размер на блока ($b = 1$ дума)
 - Следователно, брой на блоковете ($B = 8$)

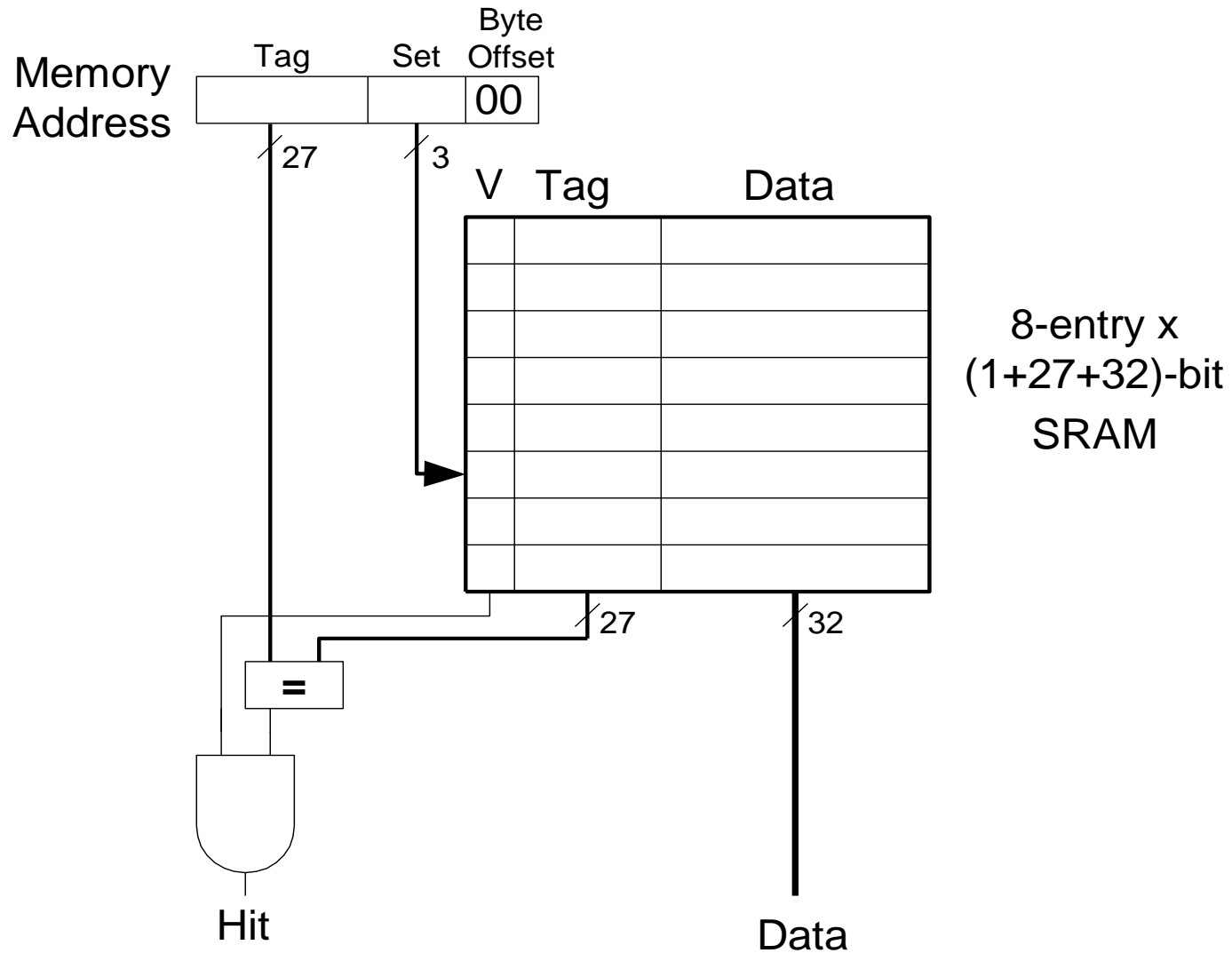
КАРХ: Тема_12: Памет и входно-изходна система

Direct Mapped Cache.



КАРХ: Тема_12: Памет и входно-изходна система

Хардуерна реализация Direct Mapped Cache.

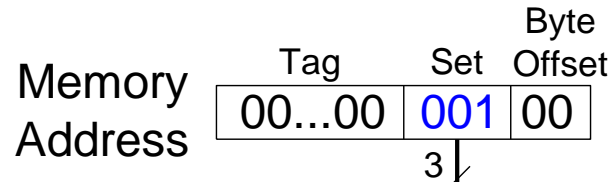


КАРХ: Тема_12: Памет и входно-изходна система

Ефективност – пример (Direct Mapped Cache Performance).

MIPS assembly code

```
    addi $t0, $0, 5
loop: beq  $t0, $0, done
      lw   $t1, 0x4($0)
      lw   $t2, 0xC($0)
      lw   $t3, 0x8($0)
      addi $t0, $t0, -1
      j    loop
done:
```



	V	Tag	Data	
	0			Set 7 (111)
	0			Set 6 (110)
	0			Set 5 (101)
	0			Set 4 (100)
	1	00...00	mem[0x00...0C]	Set 3 (011)
	1	00...00	mem[0x00...08]	Set 2 (010)
	1	00...00	mem[0x00...04]	Set 1 (001)
	0			Set 0 (000)

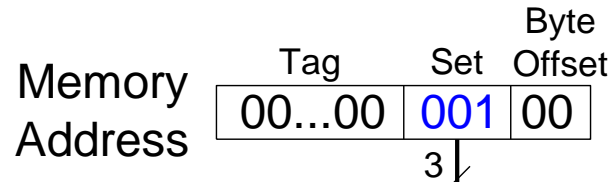
Miss Rate = ?

КАРХ: Тема_12: Памет и входно-изходна система

Ефективност – пример (Direct Mapped Cache Performance).

MIPS assembly code

```
    addi $t0, $0, 5
loop: beq  $t0, $0, done
      lw   $t1, 0x4($0)
      lw   $t2, 0xC($0)
      lw   $t3, 0x8($0)
      addi $t0, $t0, -1
      j    loop
done:
```



V	Tag	Data	
0			Set 7 (111)
0			Set 6 (110)
0			Set 5 (101)
0			Set 4 (100)
1	00...00	mem[0x00...0C]	Set 3 (011)
1	00...00	mem[0x00...08]	Set 2 (010)
1	00...00	mem[0x00...04]	Set 1 (001)
0			Set 0 (000)

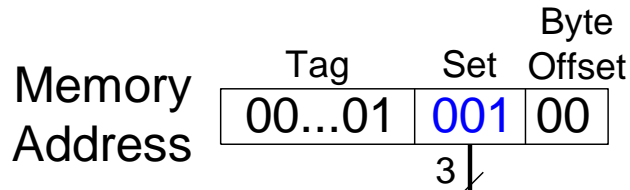
Miss Rate = 3/15 = 20%

Temporal Locality

**Принудителни пропуски
(Compulsory Misses)**

КАРХ: Тема_12: Памет и входно-изходна система

Ефективност – пример с конфликт (Direct Mapped Cache Performance).



MIPS assembly code

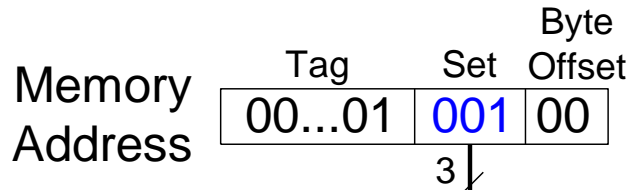
```
    addi $t0, $0, 5
loop: beq  $t0, $0, done
      lw   $t1, 0x4($0)
      lw   $t2, 0x24($0)
      addi $t0, $t0, -1
      j    loop
done:
```

V	Tag	Data	
0			Set 7 (111)
0			Set 6 (110)
0			Set 5 (101)
0			Set 4 (100)
0			Set 3 (011)
0			Set 2 (010)
1	00...00	mem[0x00...04] mem[0x00...24]	Set 1 (001)
0			Set 0 (000)

Miss Rate = ?

КАРХ: Тема_12: Памет и входно-изходна система

Ефективност – пример с конфликт (Direct Mapped Cache Performance).



MIPS assembly code

```
addi $t0, $0, 5
loop: beq  $t0, $0, done
      lw   $t1, 0x4($0)
      lw   $t2, 0x24($0)
      addi $t0, $t0, -1
      j    loop
done:
```

V	Tag	Data	
0			Set 7 (111)
0			Set 6 (110)
0			Set 5 (101)
0			Set 4 (100)
0			Set 3 (011)
0			Set 2 (010)
1	00...00	mem[0x00...04] mem[0x00...24]	Set 1 (001)
0			Set 0 (000)

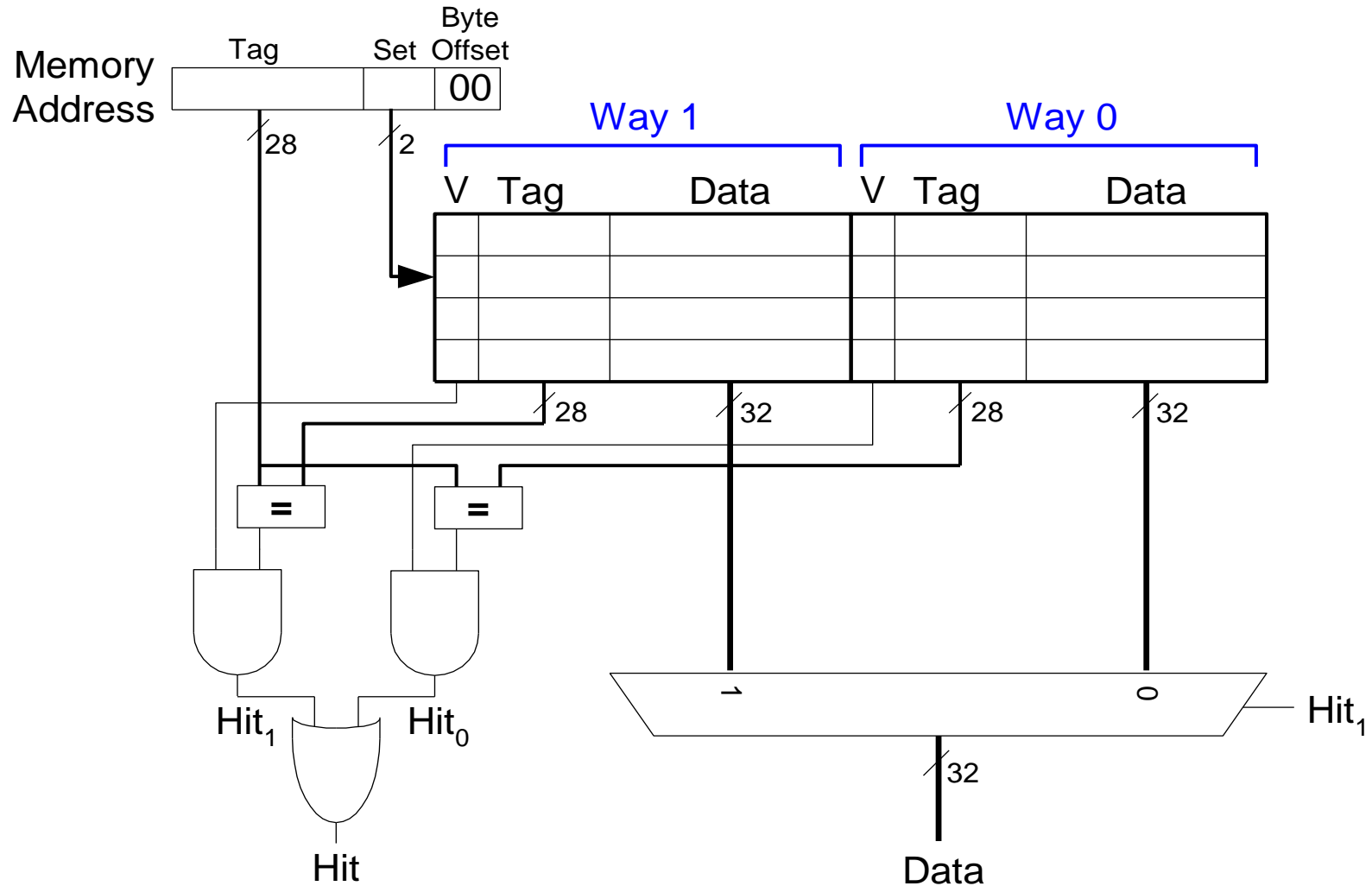
Miss Rate = 10/10 = 100%

Conflict Misses

КАРХ: Тема_12: Памет и входно-изходна система

Хардуерна реализация N-Way Set Associative Cache.

За $N = 2$



КАРХ: Тема_12: Память и входно-исходная система

Эффективность (N-Way Set Associative Cache).

MIPS assembly code

```
        addi $t0, $0, 5
loop:   beq  $t0, $0, done
        lw   $t1, 0x4($0)
        lw   $t2, 0x24($0)
        addi $t0, $t0, -1
        j    loop
done:
```

Miss Rate = ?

Way 1			Way 0			
V	Tag	Data	V	Tag	Data	
0			0			Set 3
0			0			Set 2
0			0			Set 1
0			0			Set 0

КАРХ: Тема_12: Памет и входно-изходна система

Ефективност (N-Way Set Associative Cache).

MIPS assembly code

```
        addi $t0, $0, 5
loop:   beq  $t0, $0, done
        lw   $t1, 0x4($0)
        lw   $t2, 0x24($0)
        addi $t0, $t0, -1
        j    loop
done:
```

Miss Rate = 2/10 = 20%

**Асоциативността намалява
конфликтните пропуски**

Way 1			Way 0			
V	Tag	Data	V	Tag	Data	
0			0			Set 3
0			0			Set 2
1	00...10	mem[0x00...24]	1	00...00	mem[0x00...04]	Set 1
0			0			Set 0

КАРХ: Тема_12: Памет и входно-изходна система

Fully Associative Cache.

V	Tag	Data	V	Tag	Data	V	Tag	Data	V	Tag	Data	V	Tag	Data	V	Tag	Data	V	Tag	Data

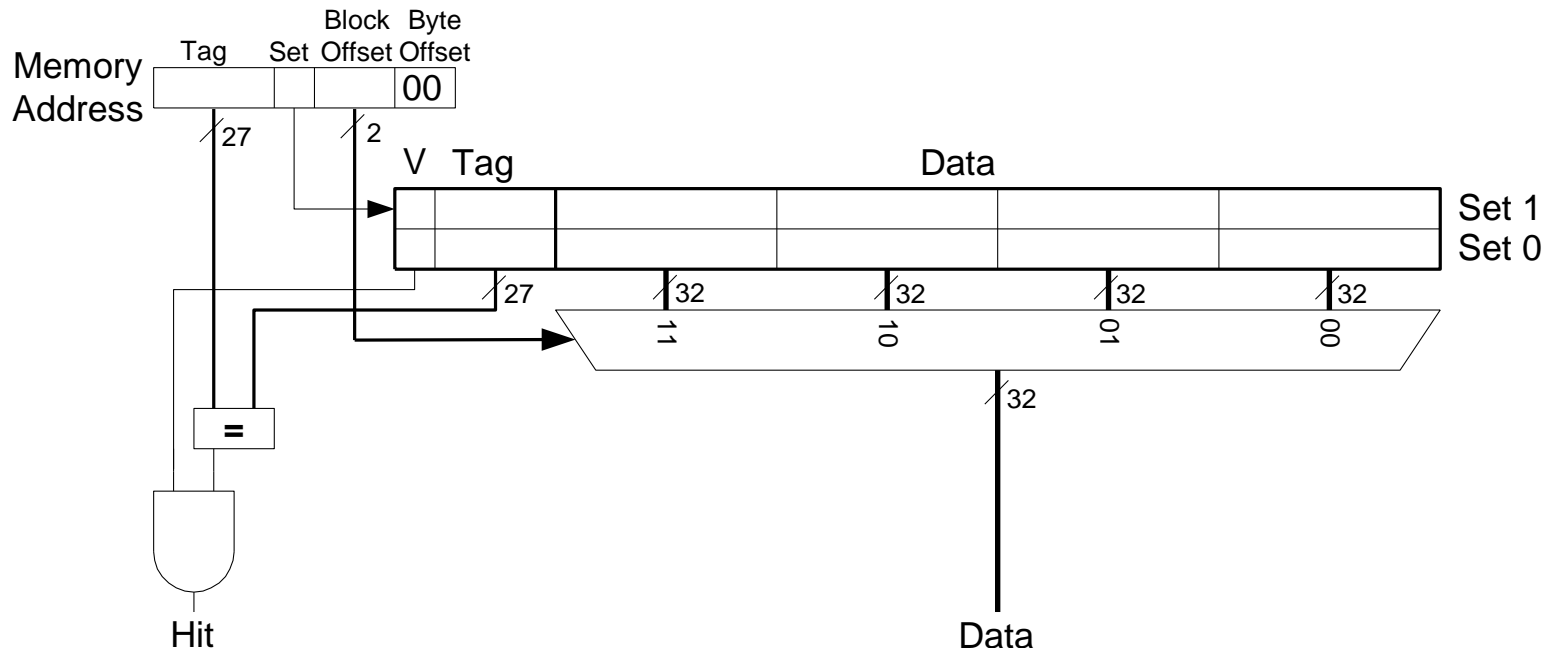
Намалява конфликтните прпуски

Скъпа за изграждане

КАРХ: Тема_12: Памет и входно-изходна система

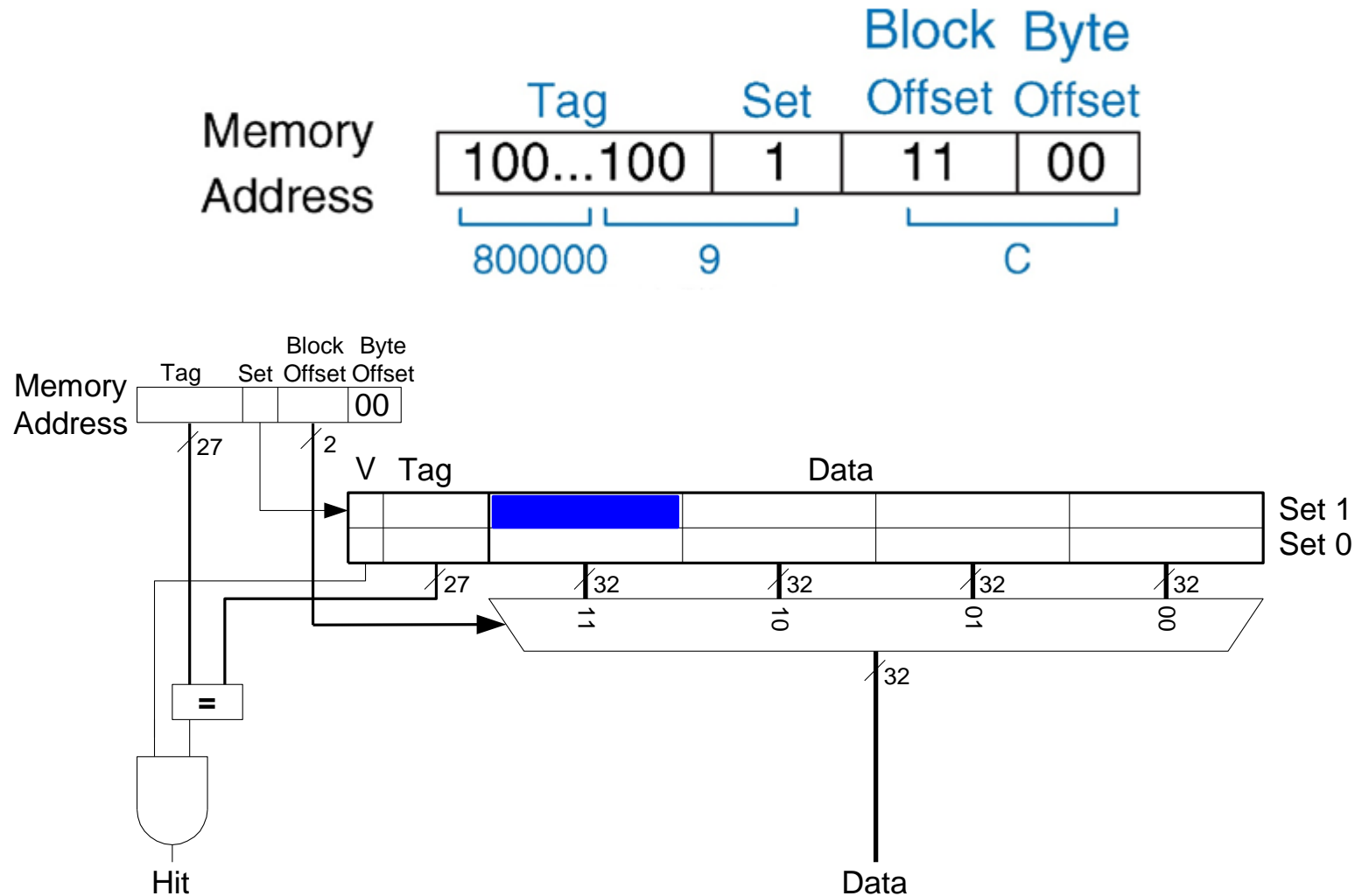
Влияние на пространствената локалност (Spatial Locality).

- Увеличаваме размера на блока:
 - Block size, $b = 4$ думи
 - $C = 8$ думи
 - Direct mapped (1 block per set)
 - Брой на блоковете, $B = 2$ ($C/b = 8/4 = 2$)



КАРХ: Тема_12: Памет и входно-изходна система

Кеш с по-голям размер на блока (Cache with Larger Block Size).



КАРХ: Тема_12: Памет и входно-изходна система

Ефективност (Кеш с по-голям размер на блока).

```
addi $t0, $0, 5
```

```
loop: beq  $t0, $0, done
```

```
      lw   $t1, 0x4($0)
```

```
      lw   $t2, 0xC($0)
```

```
      lw   $t3, 0x8($0)
```

```
      addi $t0, $t0, -1
```

```
      j    loop
```

```
done:
```

Miss Rate = ?

КАРХ: Тема_12: Памет и входно-изходна система

Ефективност (Кеш с по-голям размер на блока).

```
addi $t0, $0, 5
```

```
loop: beq  $t0, $0, done
```

```
      lw   $t1, 0x4($0)
```

```
      lw   $t2, 0xC($0)
```

```
      lw   $t3, 0x8($0)
```

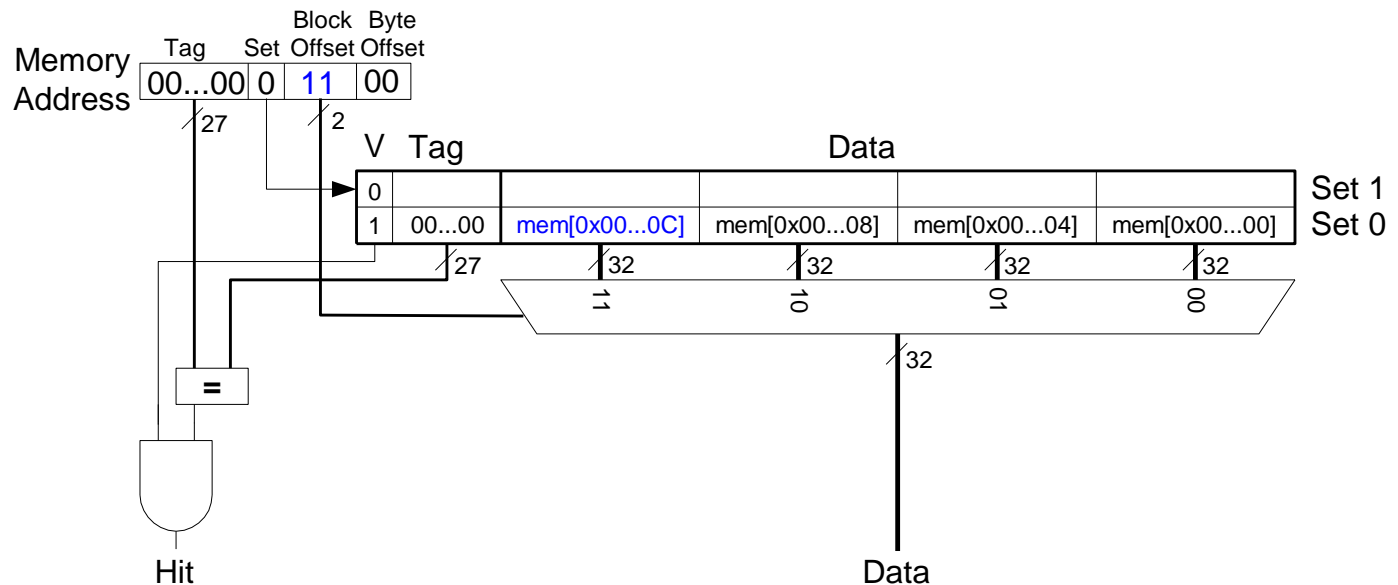
```
      addi $t0, $t0, -1
```

```
      j    loop
```

```
done:
```

Miss Rate = 1/15 = 6.67%

**По-големите блокове
намаляват принудителните
пропуски чрез пространствена
локалност (spatial locality)**



КАРХ: Тема_12: Памет и входно-изходна система

Организация на кеш паметта – обобщение.

- Капацитет (Capacity): C
- Размер на блока (Block size): b
- Брой на блоковете в кеша: $B = C/b$
- Брой на блоковете в 1 ред (сет): N
- Брой на редовете (сетове): $S = B/N$

<i>Organization</i>	Number of Ways (N)	Number of Sets ($S = B/N$)
Direct Mapped	1	B
N-Way Set Associative	$1 < N < B$	B / N
Fully Associative	B	1

КАРХ: Тема_12: Памет и входно-изходна система

Пропуски от недостатъчен капацитет на кеш паметта (Capacity Misses).

- Кешът е твърде малък за да побере всички необходими данни наведнъж.
- Когато кешът е пълен: програмата прибавя данни X като изхвърля данни Y.
- *Capacity miss* - възниква когато пак потребват данни Y.
- Как да се изберат данни Y така, че да е минимален шанса да потребват отново?
- **Least recently used (LRU) replacement:** най-отдавна използваният блок в реда (сета) се изчиства, за да поеме нови данни.

КАРХ: Тема_12: Памет и входно-изходна система

Видове пропуски.

- **Принудителни (Compulsory):** при първи достъп до данни
- **Капацитетни (Capacity):** кешът е твърде малък да побере всички необходими данни
- **Конфликтни (Conflict):** работни данни се нанасят в една и съща локация на кеша

Miss penalty: необходимото време да се получи блок данни от памет, която е на по-ниско ниво в йерархията.

КАРХ: Тема_12: Памет и входно-изходна система

Заместване на данни в кеша (LRU Replacement).

- # MIPS assembly

- lw \$t0, 0x04(\$0)
- lw \$t1, 0x24(\$0)
- lw \$t2, 0x54(\$0)

Way 1				Way 0				
V	U	Tag	Data	V	Tag	Data		
0	0			0				Set 3 (11)
0	0			0				Set 2 (10)
0	0			0				Set 1 (01)
0	0			0				Set 0 (00)

КАРХ: Тема_12: Памет и входно-изходна система

Заместване на данни в кеша (LRU Replacement).

- # MIPS assembly

- lw \$t0, 0x04(\$0)
- lw \$t1, 0x24(\$0)
- lw \$t2, 0x54(\$0)

Way 1				Way 0				
V	U	Tag	Data	V	Tag	Data		
0	0			0				Set 3 (11)
0	0			0				Set 2 (10)
1	0	00...010	mem[0x00...24]	1	00...000	mem[0x00...04]		Set 1 (01)
0	0			0				Set 0 (00)

(a)

Way 1				Way 0				
V	U	Tag	Data	V	Tag	Data		
0	0			0				Set 3 (11)
0	0			0				Set 2 (10)
1	1	00...010	mem[0x00...24]	1	00...101	mem[0x00...54]		Set 1 (01)
0	0			0				Set 0 (00)

(b)

КАРХ: Тема_12: Памет и входно-изходна система

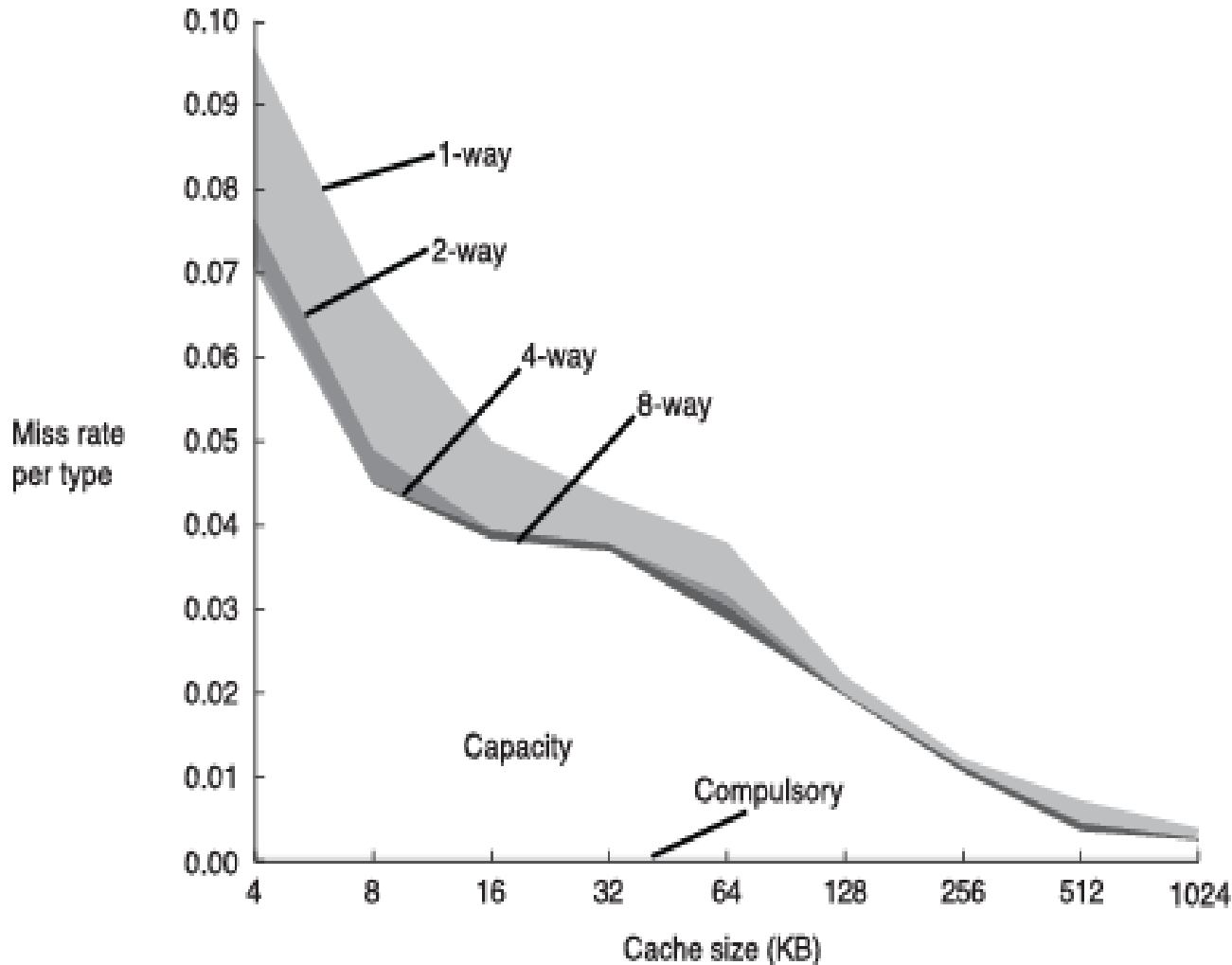
Кеш памет – обобщение.

- **Какви данни се съдържат в кеша?**
 - Наскоро използвани данни (temporal locality)
 - Близки по място до тях (spatial locality)
- **Как се намират данните?**
 - Редът (сетът) се определя от адреса на данните
 - Думата в блока също се определя от адреса
 - В асоциативния кеш, данните могат да бъдат в един от няколко пътя
- **Кои данни се заместват?**
 - Тези, от най-отдавна използвания път в реда (сета)

КАРХ: Тема_12: Памет и входно-изходна система

Степен на пропуските – тенденции.

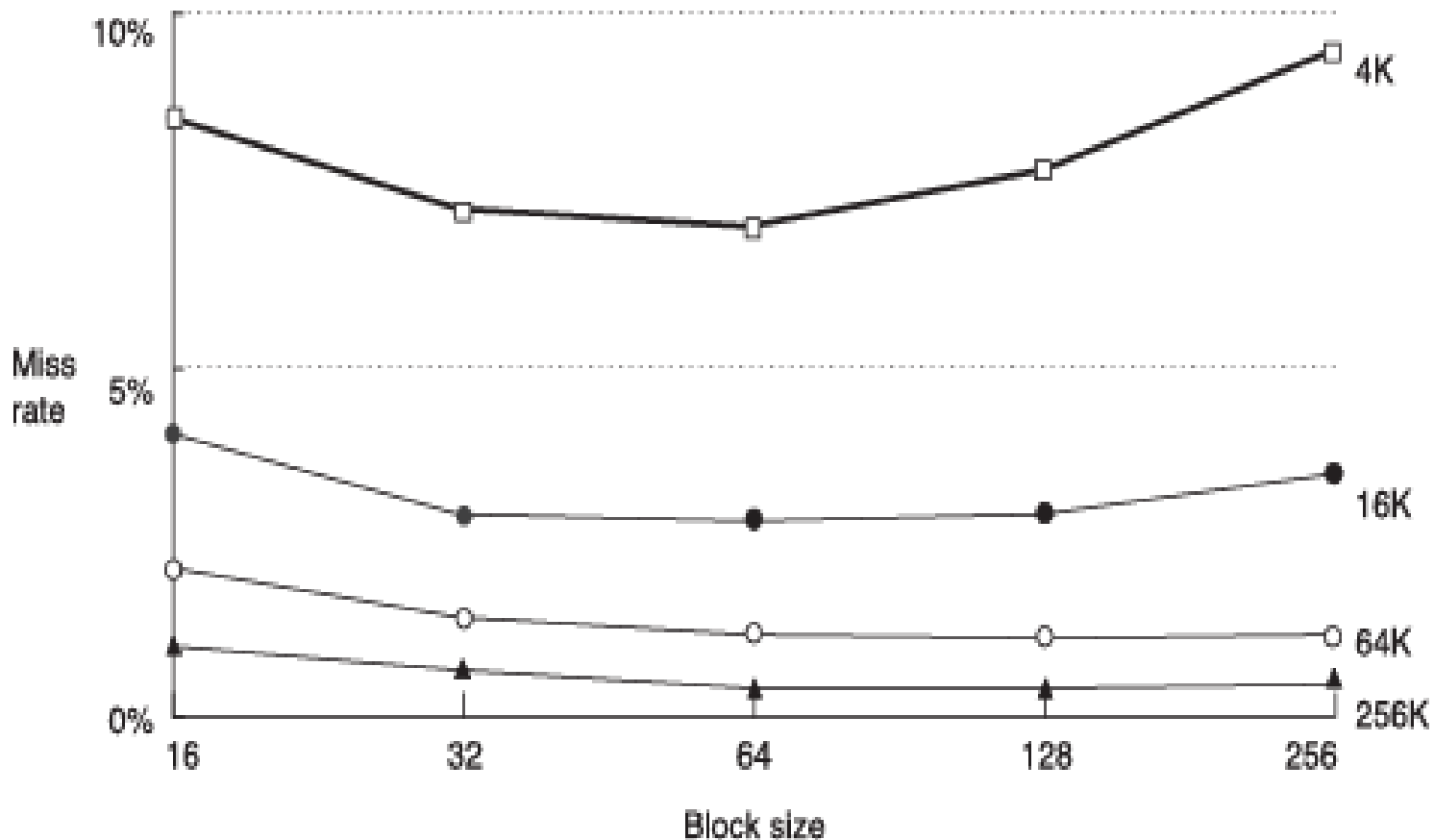
- По-големият кеш намалява пропуските от капацитет
- По-голямата асоциативност намалява конфликтните пропуски



КАРХ: Тема_12: Памет и входно-изходна система

Степен на пропускаите – тенденции.

- По-големите блокове намаляват принудителните пропускаки
- По-големите блокове увеличават конфликтните пропускаки



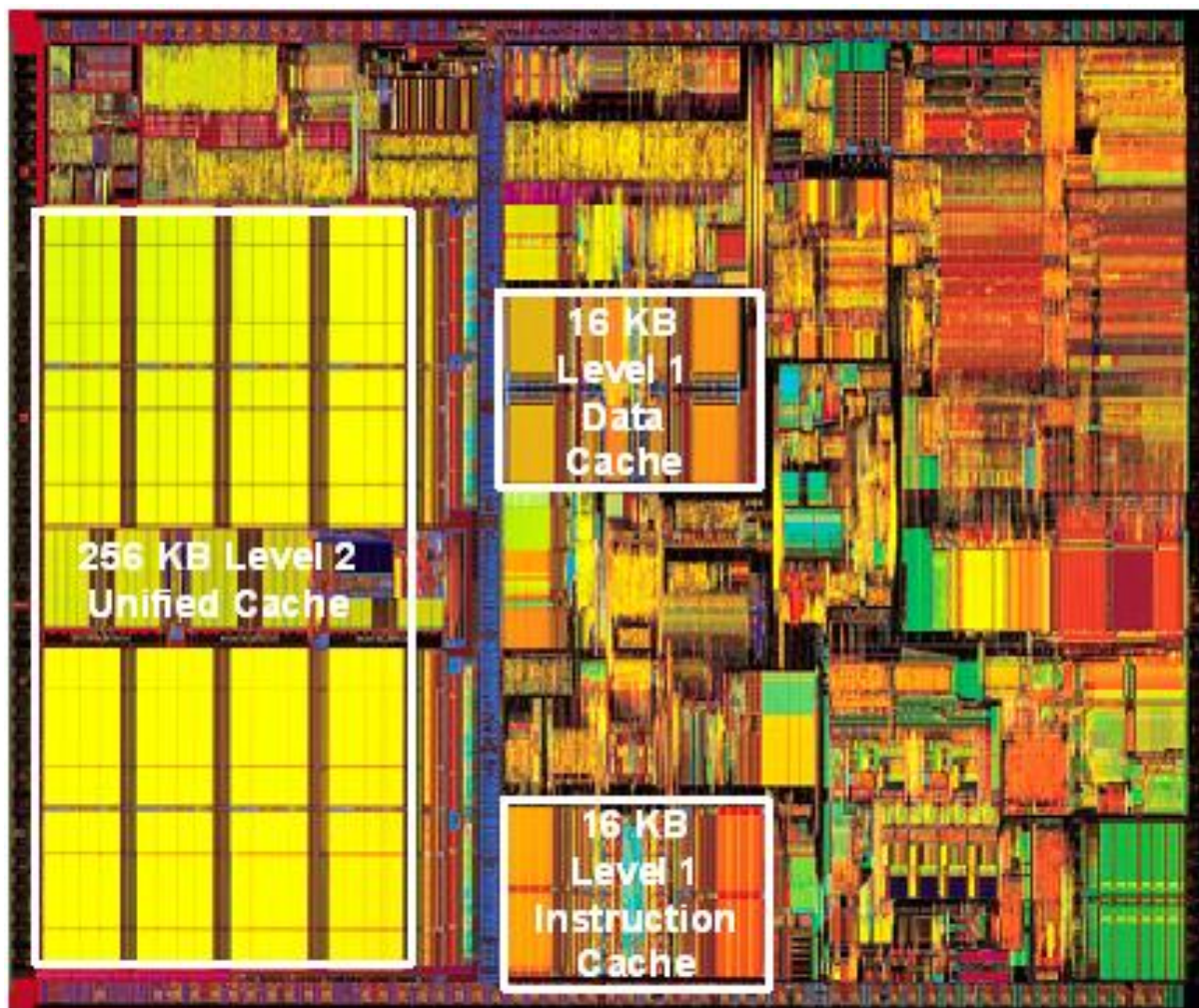
КАРХ: Тема_12: Памет и входно-изходна система

Кеш памет на много нива (Multilevel Caches).

- По-големите кешове имат по-ниска степен на пропуски, но по-голямо време на достъп.
- За това кеш-паметта се прави йерархична – на няколко нива.
- Level 1: малка и бърза (напр. 16 KB, 1 cycle).
- Level 2: по-голяма и по-бавна (напр. 256 KB, 2-6 cycles).
- Повечето модерни РС-та имат L1, L2, и L3 cache.

КАРХ: Тема_12: Памет и входно-изходна система

Пример – Intel Pentium III процесор.



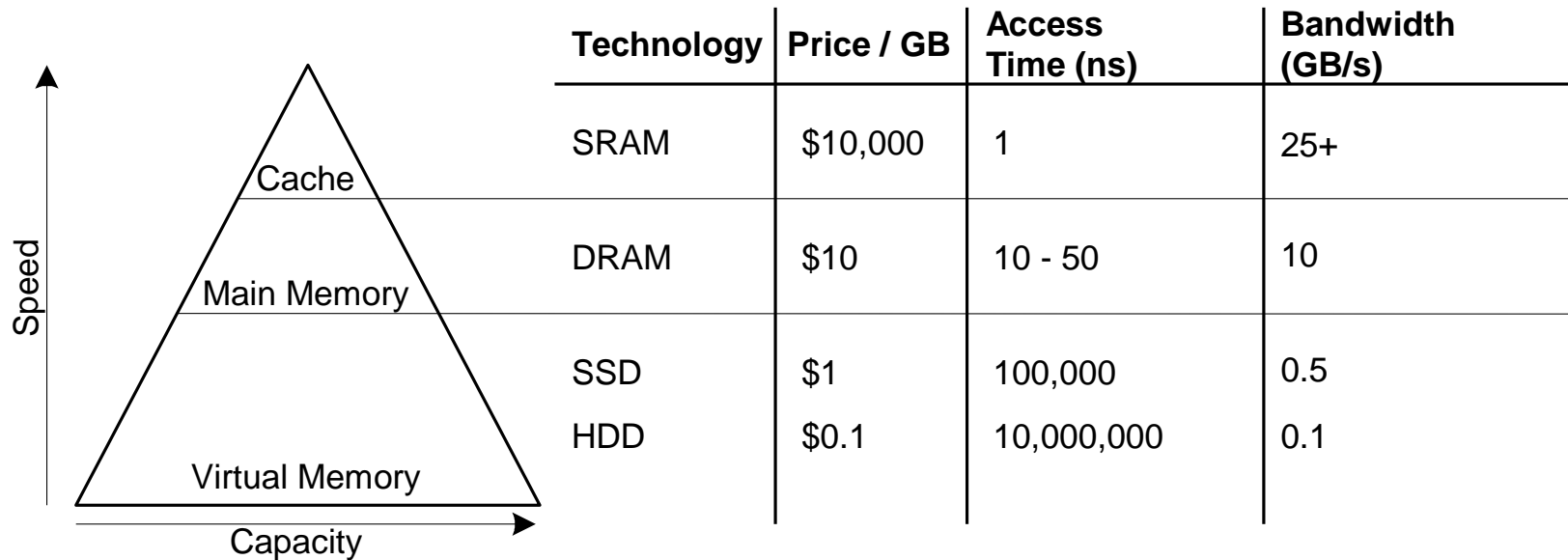
КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет (Virtual Memory).

- Създава илюзията за по-голяма памет.
- Основната памет (DRAM) действа като кеш за твърдия диск (hard disk).

КАРХ: Тема_12: Памет и входно-изходна система

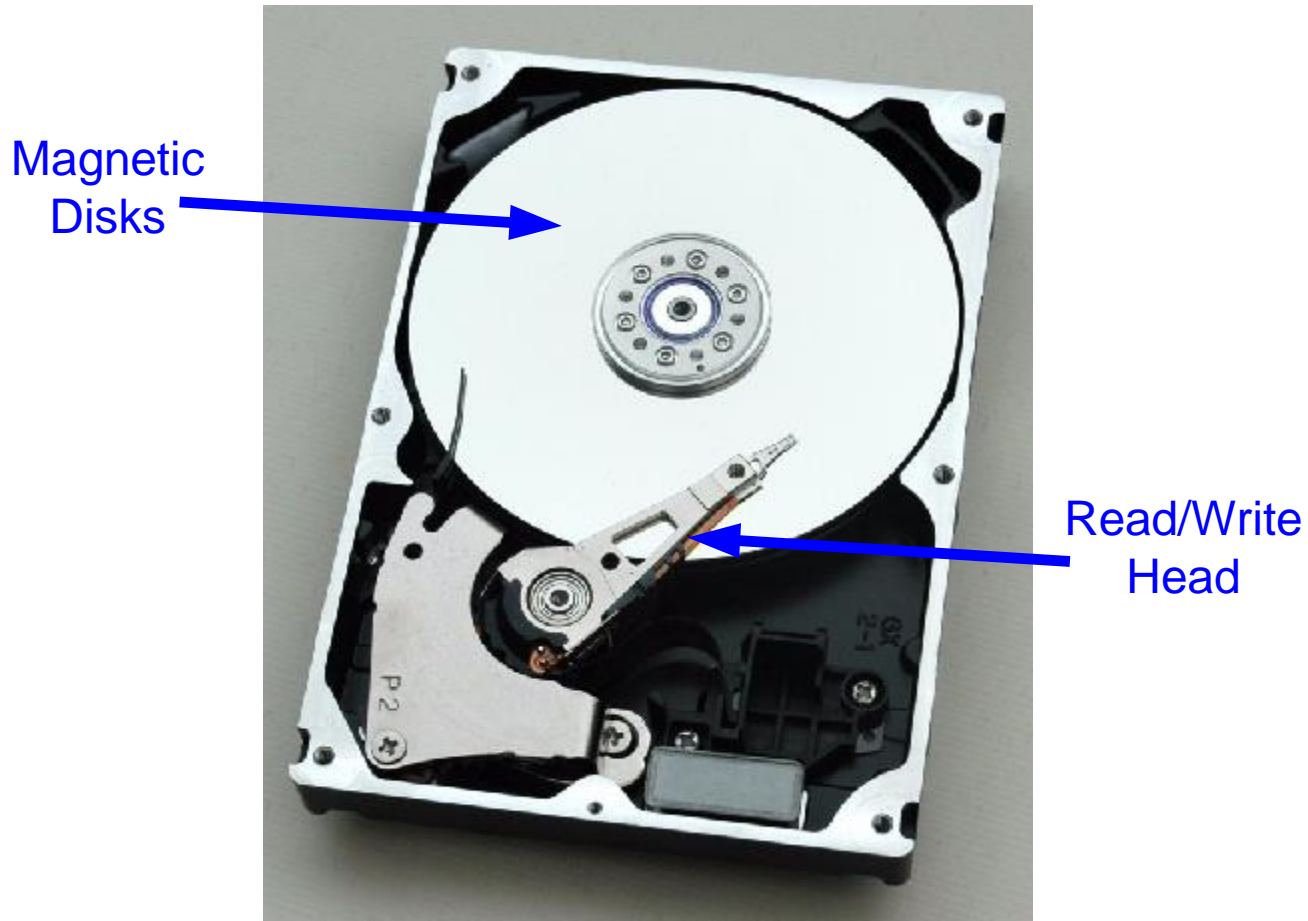
Йерархия на паметта.



- **Physical Memory:** DRAM – Основна памет (Main Memory)
- **Virtual Memory:** Твърд/Твърдотелен диск (Hard drive)
 - Бавна, Голяма, Евтина (Slow, Large, Cheap)

КАРХ: Тема_12: Памет и входно-изходна система

Твърд диск (Hard Disk).



Отнема милисекунди откриването на точното положение на данните върху диска

КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет (Virtual Memory).

- **Виртуални адреси**
 - Програмите използват виртуални адреси
 - Цялото виртуално адресно пространство се съхранява на твърдия диск
 - Част (Subset) от данните с виртуални адреси се намират в DRAM
 - Процесорът трансформира виртуалните адреси (virtual addresses) във *физически адреси (physical addresses)* (DRAM addresses)
 - Данните, които не се намират в DRAM се извличат от твърдия диск
- **Защита на паметта (Memory Protection)**
 - Всяка програма има собствено адресно транслиране (virtual to physical mapping)
 - Две програми могат да използват едни и същи виртуални адреси за различни данни
 - Програмите не усещат наличието на другите изпълнявани в момента програми (не си влияят)
 - Дадена програма (или вирус) не може да увреди памет ползвана от друга програма

КАРХ: Тема_12: Памет и входно-изходна система

Аналогия Кеш – Виртуална памет.

Cache	Virtual Memory
Block	Page
Block Size	Page Size
Block Offset	Page Offset
Miss	Page Fault
Tag	Virtual Page Number

Физическата памет действа като кеш за виртуалната памет.

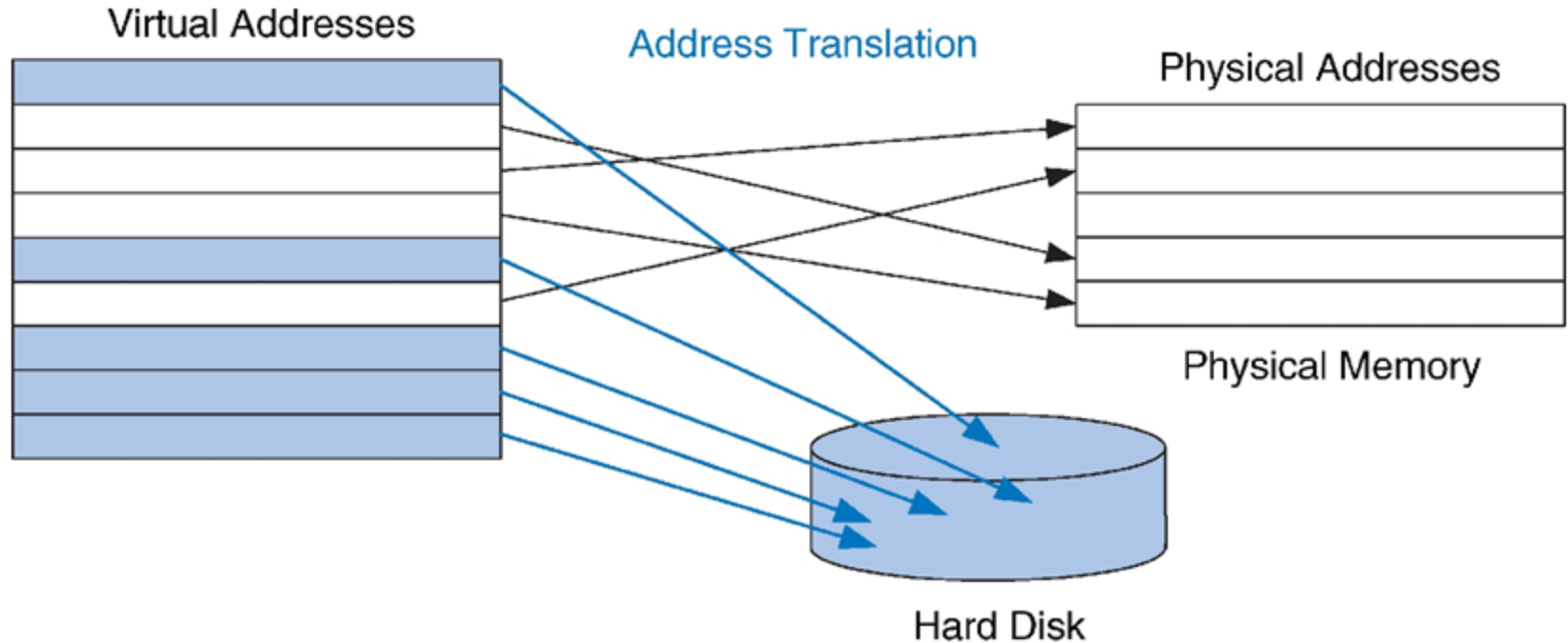
КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет – дефиниции.

- **Размер на страница (Page size):** количество памет (данни), трансферирано наведнъж от твърдия диск в DRAM.
- **Транслация на адрес (Address translation):** определяне на физическия адрес от виртуалния адрес
- **Page table:** lookup table използвана за транслиране на виртуалните адреси във физически адреси

КАРХ: Тема_12: Памет и входно-изходна система

Виртуални и физически адреси.



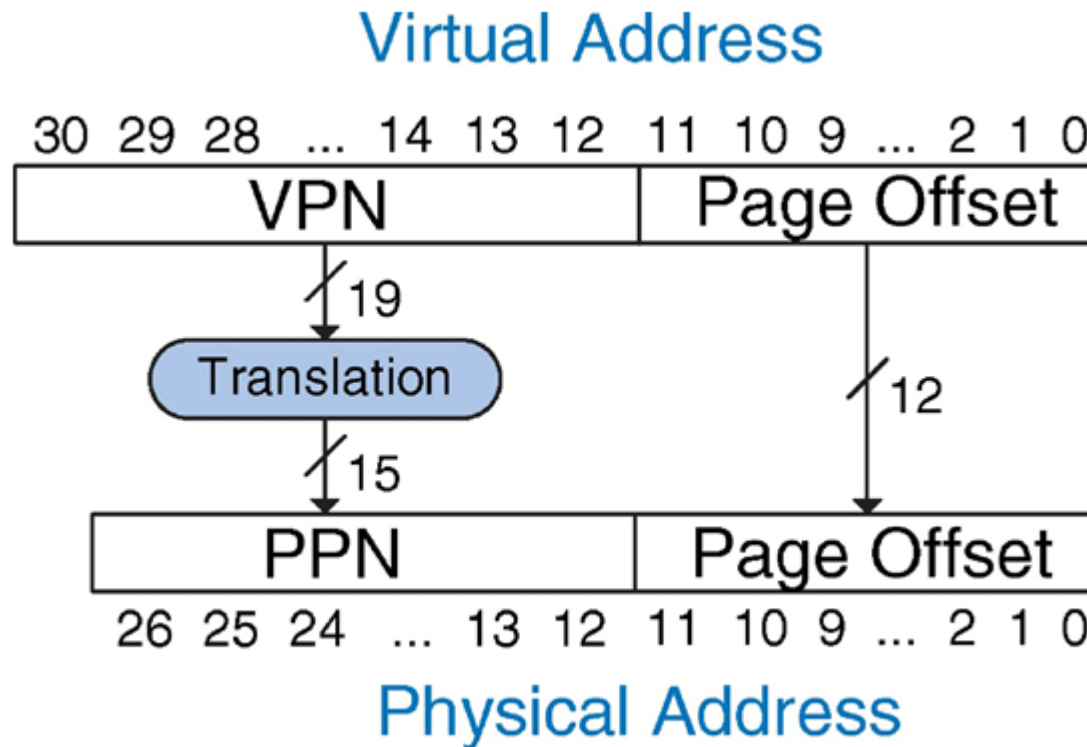
- Повечето опити за достъп до данни попадат във физическата памет.
- Въпреки това програмите използват голям обем виртуална памет.

КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет – пример.

- **Система с:**

- Размер на виртуалната памет (Virtual memory size): $2\text{ GB} = 2^{31}\text{ bytes}$
- Размер на физическата памет (Physical memory size): $128\text{ MB} = 2^{27}\text{ bytes}$
- Размер на страницата (Page size): $4\text{ KB} = 2^{12}\text{ bytes}$



КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет – пример.

- **Система с:**

- Размер на виртуалната памет (Virtual memory size): $2\text{ GB} = 2^{31}\text{ bytes}$
- Размер на физическата памет (Physical memory size): $128\text{ MB} = 2^{27}\text{ bytes}$
- Размер на страницата (Page size): $4\text{ KB} = 2^{12}\text{ bytes}$

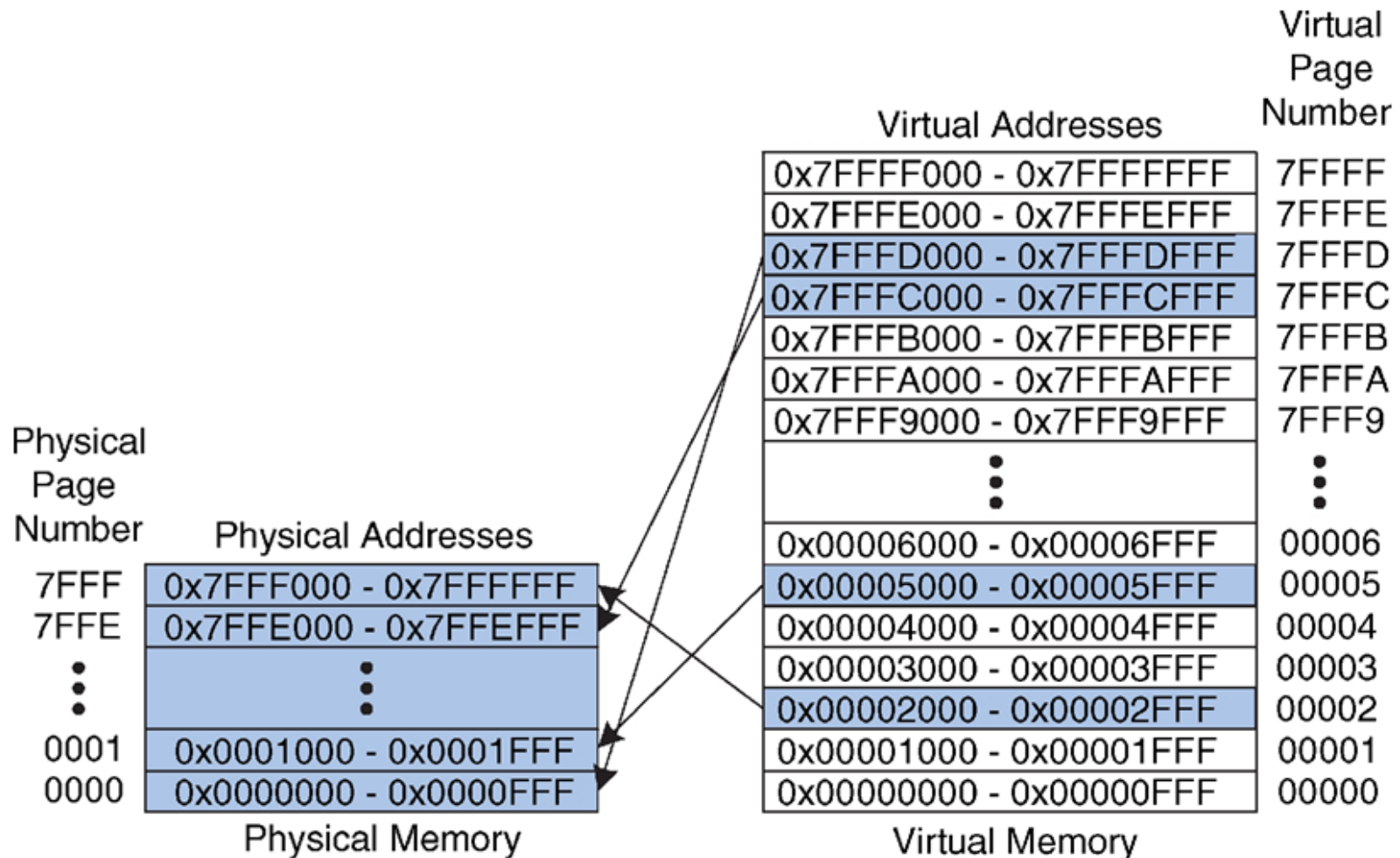
- **Организация:**

- Виртуален адрес (Virtual address): **31** bits
- Физически адрес (Physical address): **27** bits
- Page offset: **12** bits
- Брой виртуални страници (# Virtual pages) = $2^{31}/2^{12} = 2^{19}$ (VPN = 19 bits)
- Брой физически страници (# Physical pages) = $2^{27}/2^{12} = 2^{15}$ (PPN = 15 bits)

КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет – пример.

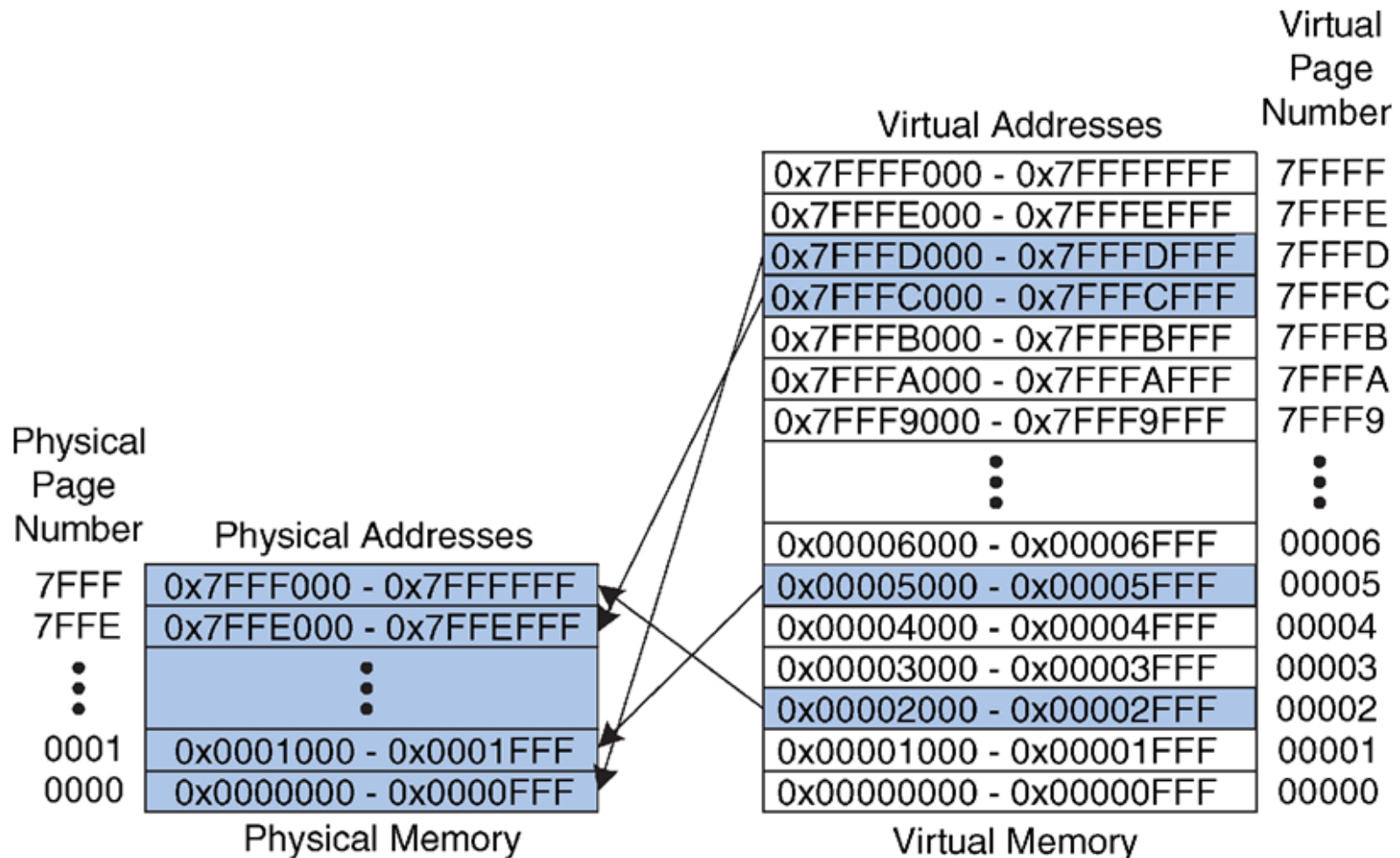
- 19-bit virtual page numbers
- 15-bit physical page numbers



КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет – пример.

Какъв физически адрес отговаря (съответства) на виртуалния адрес **0x247C**?

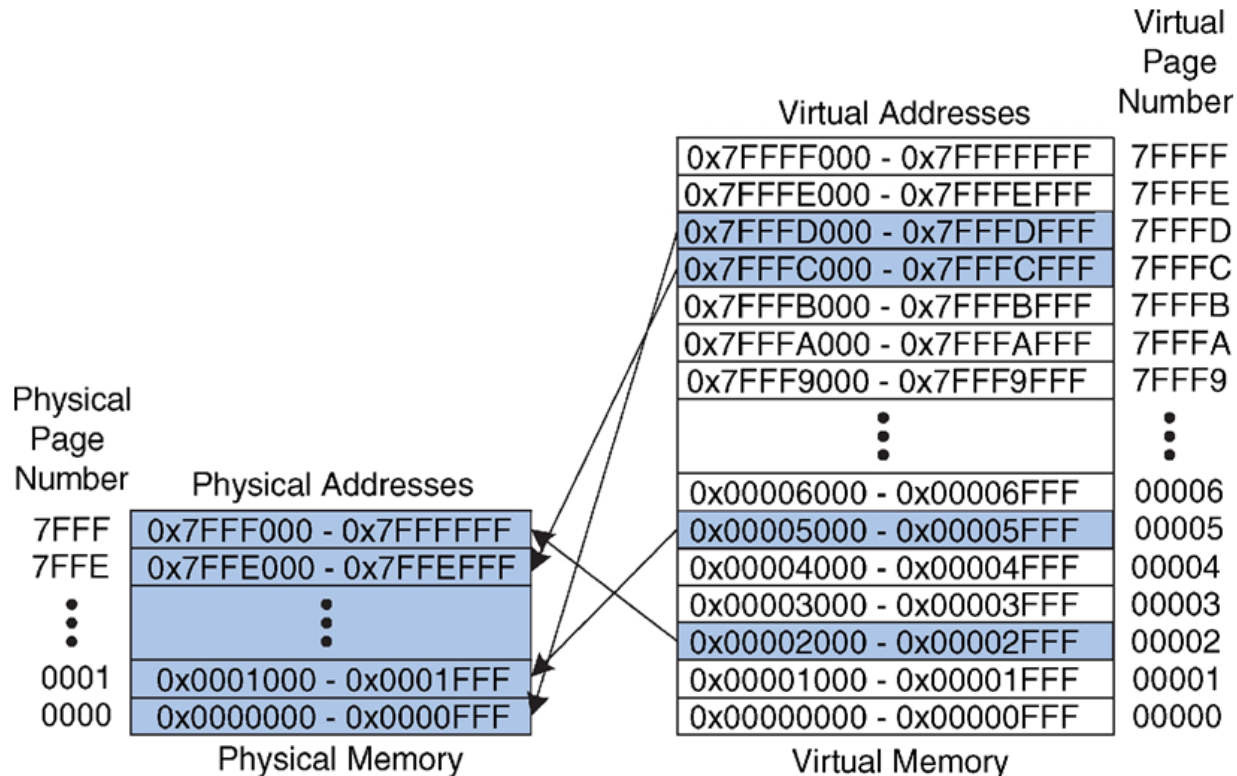


КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет – пример.

Какъв физически адрес отговаря (съответства) на виртуалния адрес **0x247C**?

- VPN = **0x2**
- VPN 0x2 maps to PPN **0x7FFF**
- 12-bit page offset: **0x47C**
- Physical address = **0x7FFF47C**



КАРХ: Тема_12: Памет и входно-изходна система

Как се извършва трансляцията на страници.

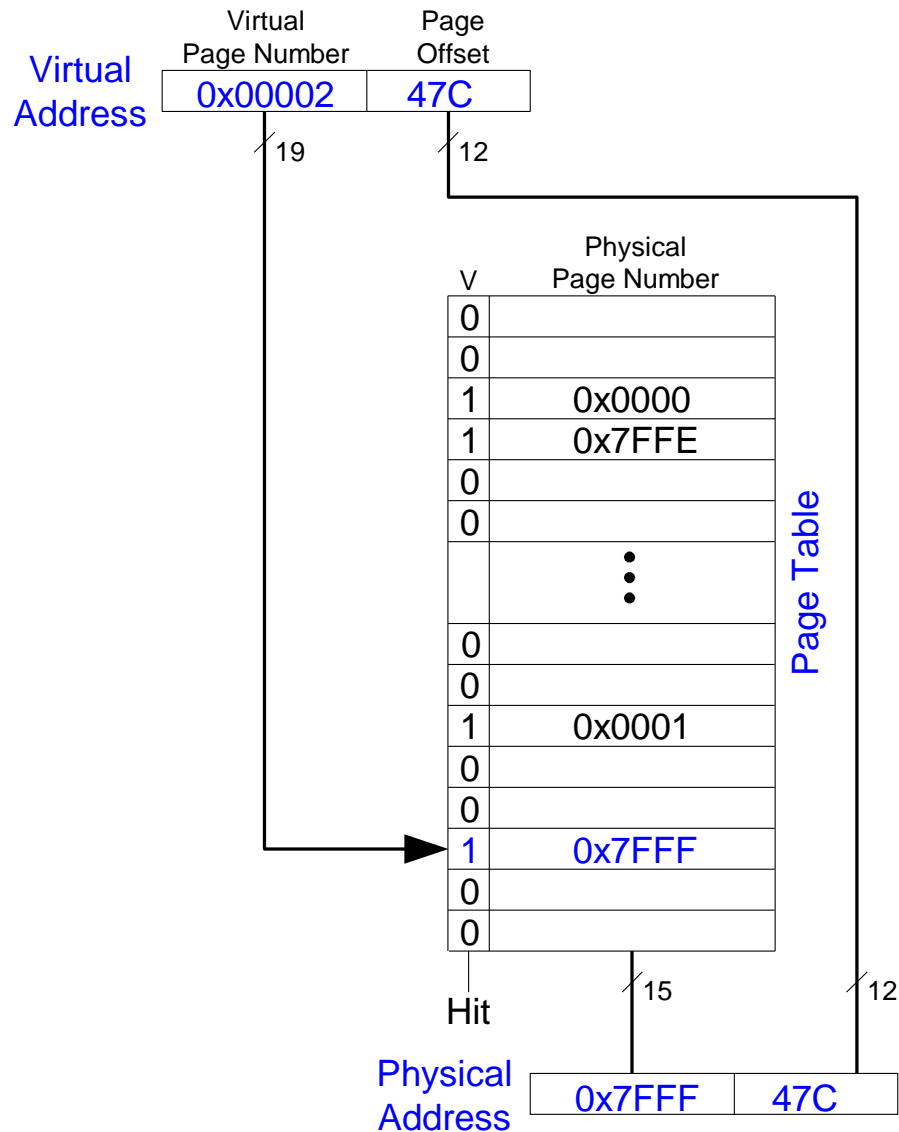
- **Page table**

- Задава вход към всяка виртуална страница
- Входящи полета (Entry fields):
 - **Бит за валидност (Valid bit):** 1, ако страницата е във физическата памет
 - **Номер на физическа страница (Physical page number):** показва къде се намира страницата

КАРХ: Тема_12: Памет и входно-изходна система

Page Table – пример.

VPN is index
into page table



КАРХ: Тема_12: Памет и входно-изходна система

Page Table – пример 1.

Какъв физически адрес отговаря
на виртуалния адрес **0x5F20**?

V	Physical Page Number
0	
0	
1	0x0000
1	0x7FFE
0	
0	
	⋮
0	
0	
1	0x0001
0	
0	
1	0x7FFF
0	
0	

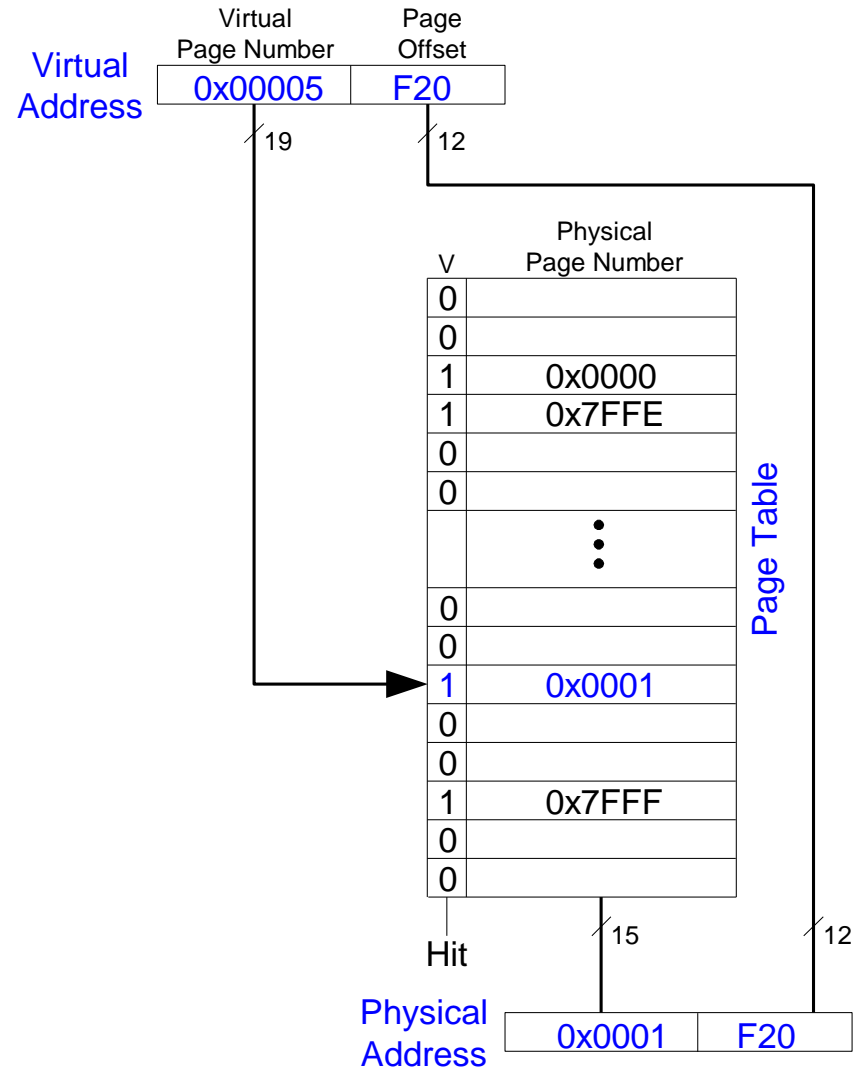
Page Table

КАРХ: Тема_12: Памет и входно-изходна система

Page Table – пример 1.

Какъв физически адрес отговаря на виртуалния адрес **0x5F20**?

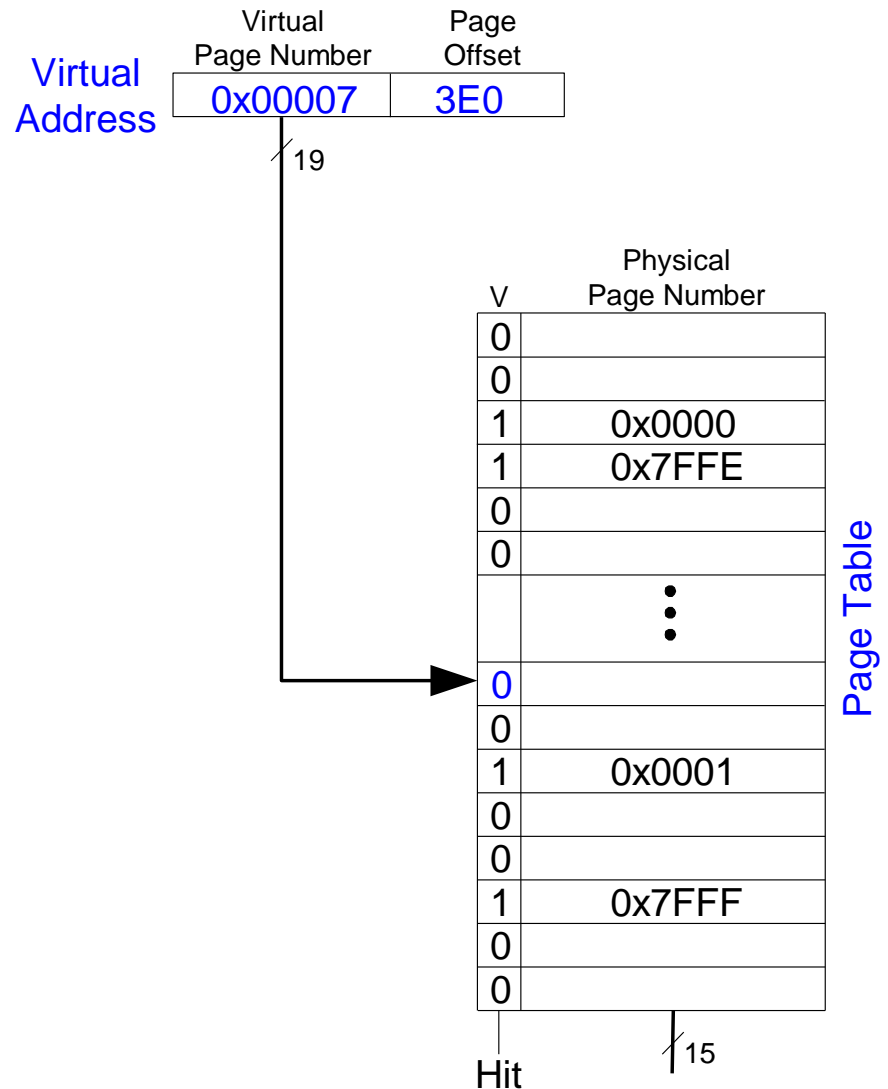
- VPN = **5**
- Entry 5 in page table
VPN 5 => physical page **1**
- Physical address: **0x1F20**



КАРХ: Тема_12: Памет и входно-изходна система

Page Table – пример 2.

Какъв физически адрес отговаря
на виртуалния адрес **0x73E0**?

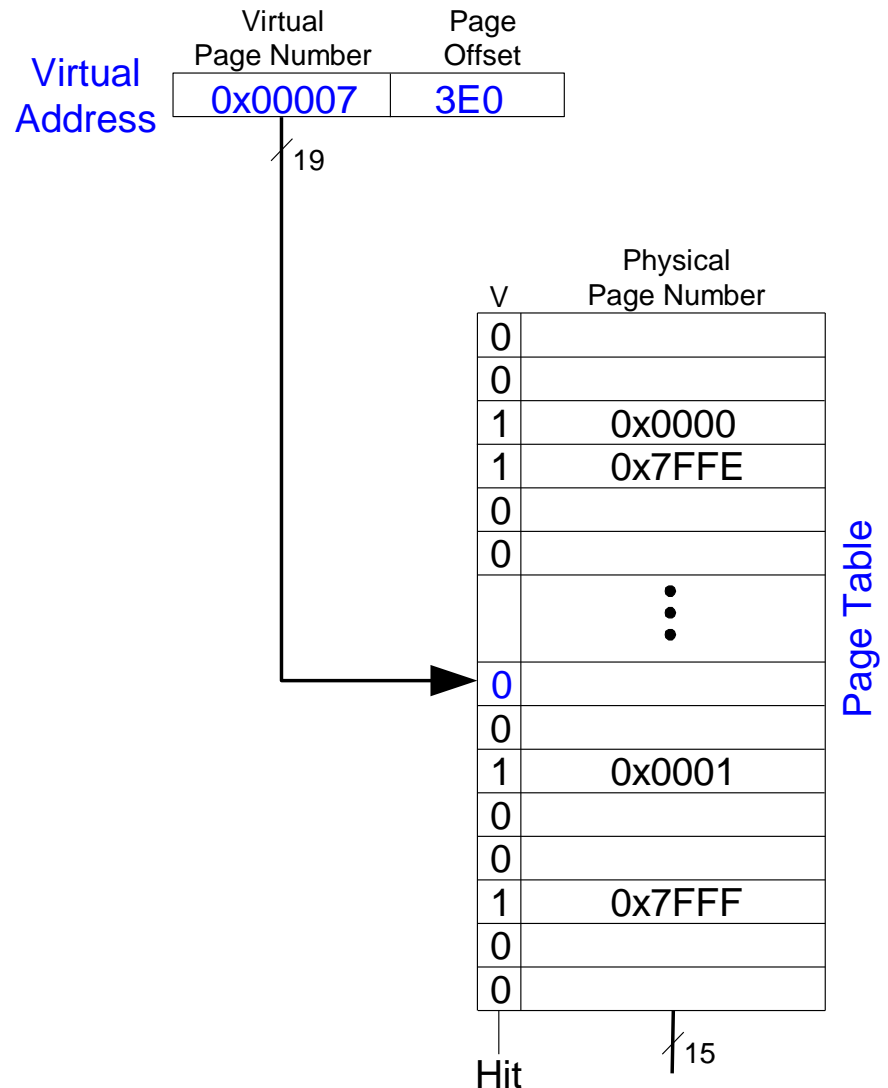


КАРХ: Тема_12: Памет и входно-изходна система

Page Table – пример 2.

Какъв физически адрес отговаря на виртуалния адрес **0x73E0**?

- VPN = 7
- Вход 7 е невалиден
- Virtual page must be *paged* into physical memory from disk



КАРХ: Тема_12: Памет и входно-изходна система

Page Table – предизвикателства.

- **Page table е голяма**
 - Обикновено се намира във физическата памет
- Четене/запис изискват 2 достъпа до основната памет:
 - Един път за трансляция (page table read)
 - Един път за достъп до данните (след трансляцията)
- Това намалява ефективността наполовина.
 - *Освен ако се потърси друго решение...*

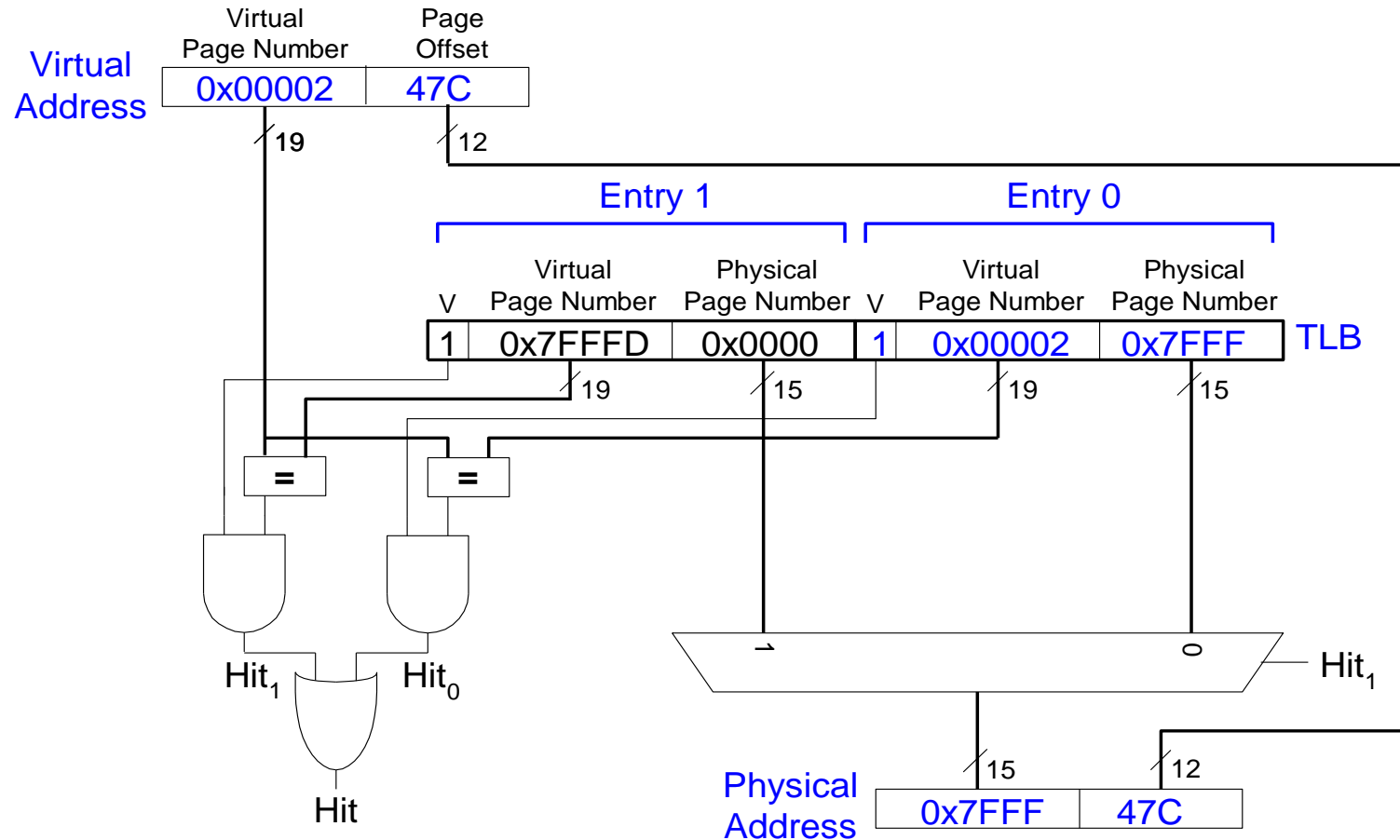
КАРХ: Тема_12: Памет и входно-изходна система

Translation Lookaside Buffer (TLB).

- Това е малък кеш за най-скорошните трансляции.
- Намалява броя на достъпите до паметта за повечето четения/записи от 2 до 1.
- TLB
 - Малко време на достъп: accessed in < 1 cycle
 - Малък обем: типично 16 - 512 входа
 - Fully associative
 - $> 99\%$ hit rates типично

КАРХ: Тема_12: Память и входно-исходная система

Translation Lookaside Buffer (TLB) – пример.



КАРХ: Тема_12: Памет и входно-изходна система

Виртуална памет – обобщение.

- Виртуалната памет увеличава обема на паметта (**capacity**).
- Част от виртуалните страници се намират във физическата памет
- **Page table** свързва виртуалните страници с физическите (address translation)
- **TLB** ускорява процеса на трансляция
- Различни page tables за различните програми, което гарантира защита на паметта (**memory protection**).

КАРХ: Тема_12: Памет и входно-изходна система

Адресиране на входно-изходни устройства.

- Процесорът се обръща към входно-изходните устройства (клавиатура, монитор, принтер, ...) по същия начин както към паметта.
- На всяко входно-изходно устройство се присвоява един или повече адреси
- Когато се направи обръщение към такъв адрес, данни се четат/записват от такова устройство (както от паметта).
- Заделя се специално адресно пространство за входно-изходните устройства.

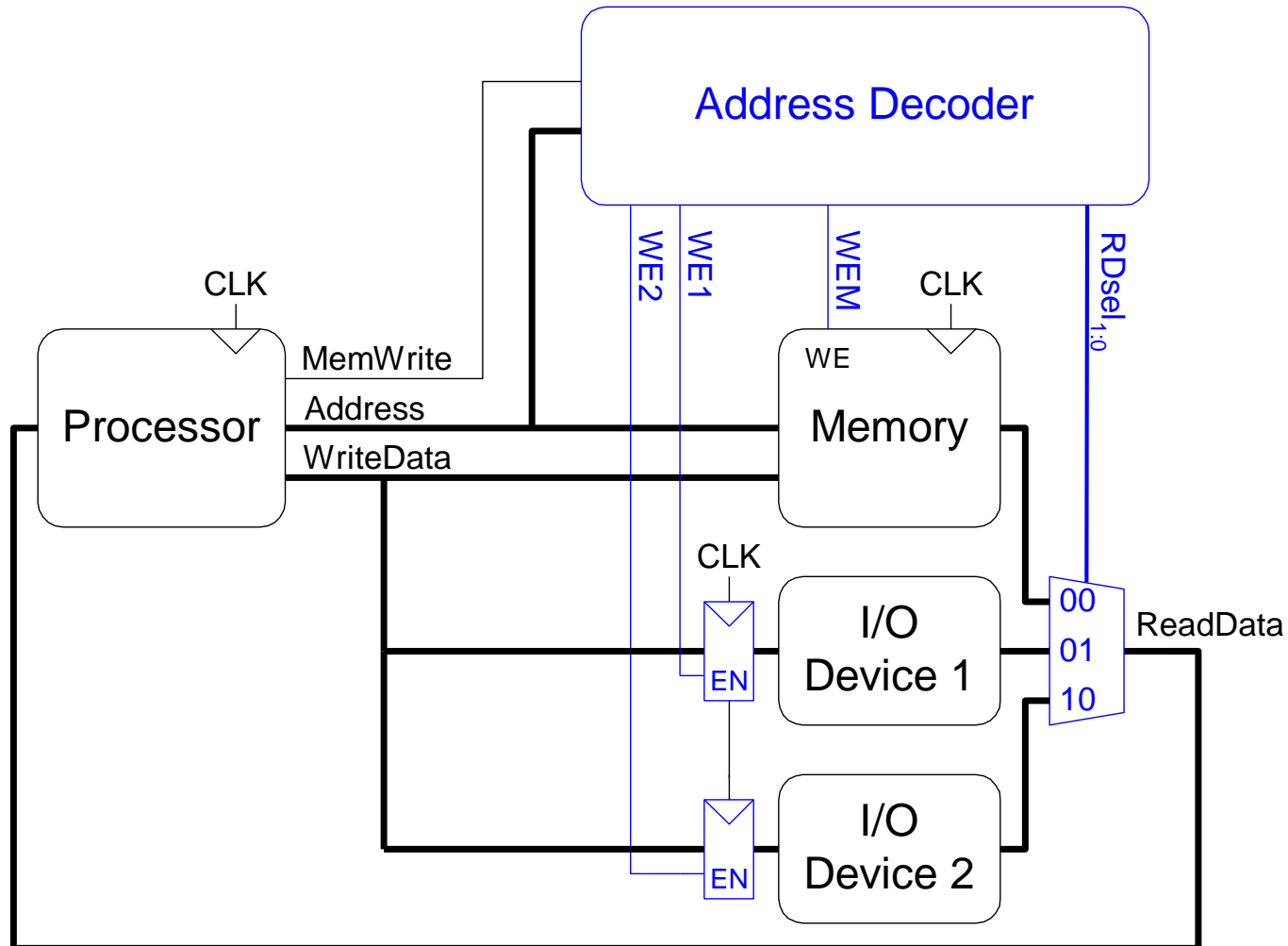
КАРХ: Тема_12: Памет и входно-изходна система

Хардуер за адресиране на входно-изходни устройства.

- **Адресен декодер (Address Decoder):**
 - Според адреса определя устройството, което се свързва с процесора.
- **Входно-изходни регистри (I/O Registers):**
 - Съдържат стойностите записвани във входно-изходните устройства.
- **ReadData Multiplexer:**
 - Избира паметта или входно-изходните устройства като източник на данни за процесора.

КАРХ: Тема_12: Памет и входно-изходна система

Хардуер за адресиране на входно-изходни устройства.



КАРХ: Тема_12: Памет и входно-изходна система

Хардуер за адресиране на входно-изходни устройства – пример.

- Нека I/O Device 1 има адрес 0xFFFFFFFF4
 - Запис на числото 42 в I/O Device 1
 - Четене на стойност от I/O Device 1 и поставянето ѝ в регистъра \$t3

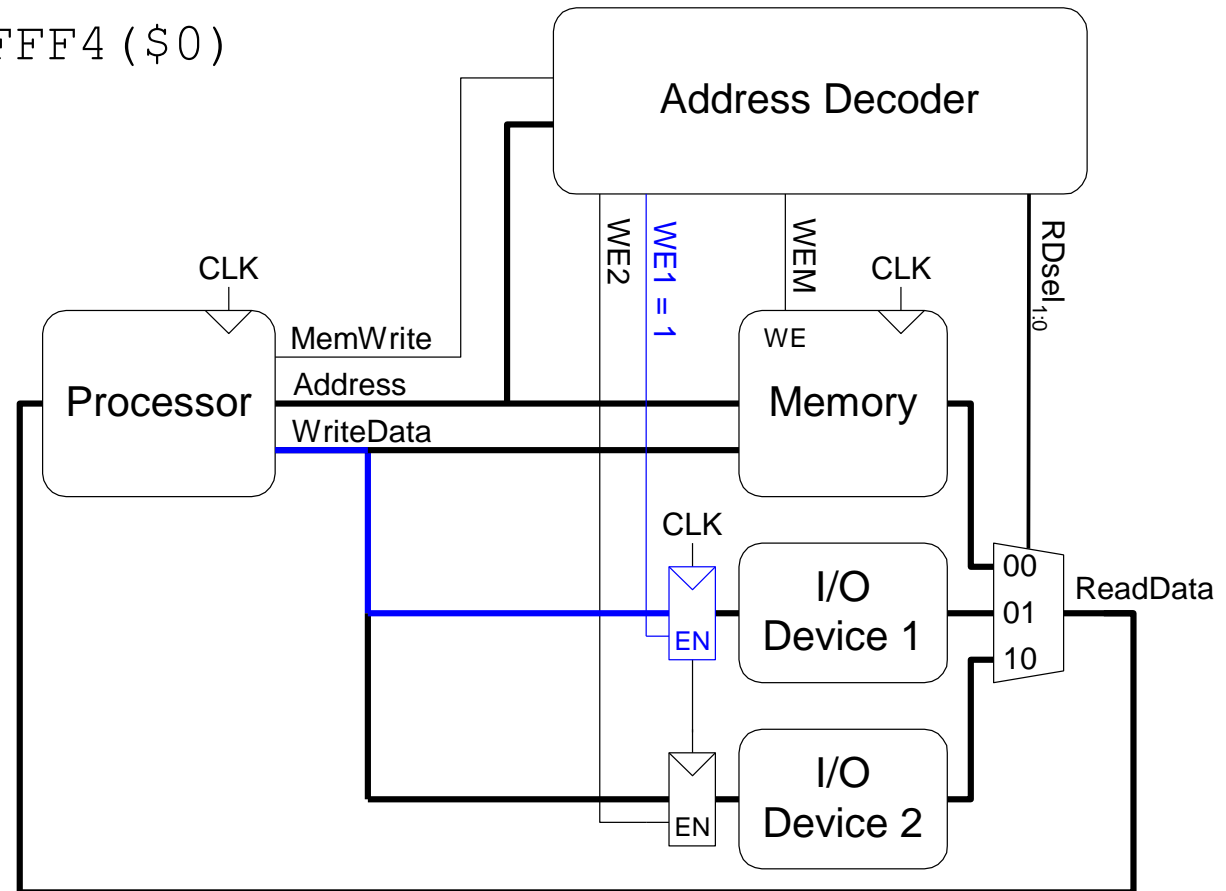
КАРХ: Тема_12: Памет и входно-изходна система

Хардуер за адресиране на входно-изходни устройства – пример.

- Запис на числото 42 в I/O Device 1 (0xFFFFFFFF4)**

```
addi $t0, $0, 42
```

```
sw $t0, 0xFFF4($0)
```



КАРХ: Тема_12: Памет и входно-изходна система

Хардуер за адресиране на входно-изходни устройства – пример.

- Четене на стойност от I/O Device 1 и поставянето ѝ в регистъра \$t3

```
lw $t3, 0xFFF4($0)
```

