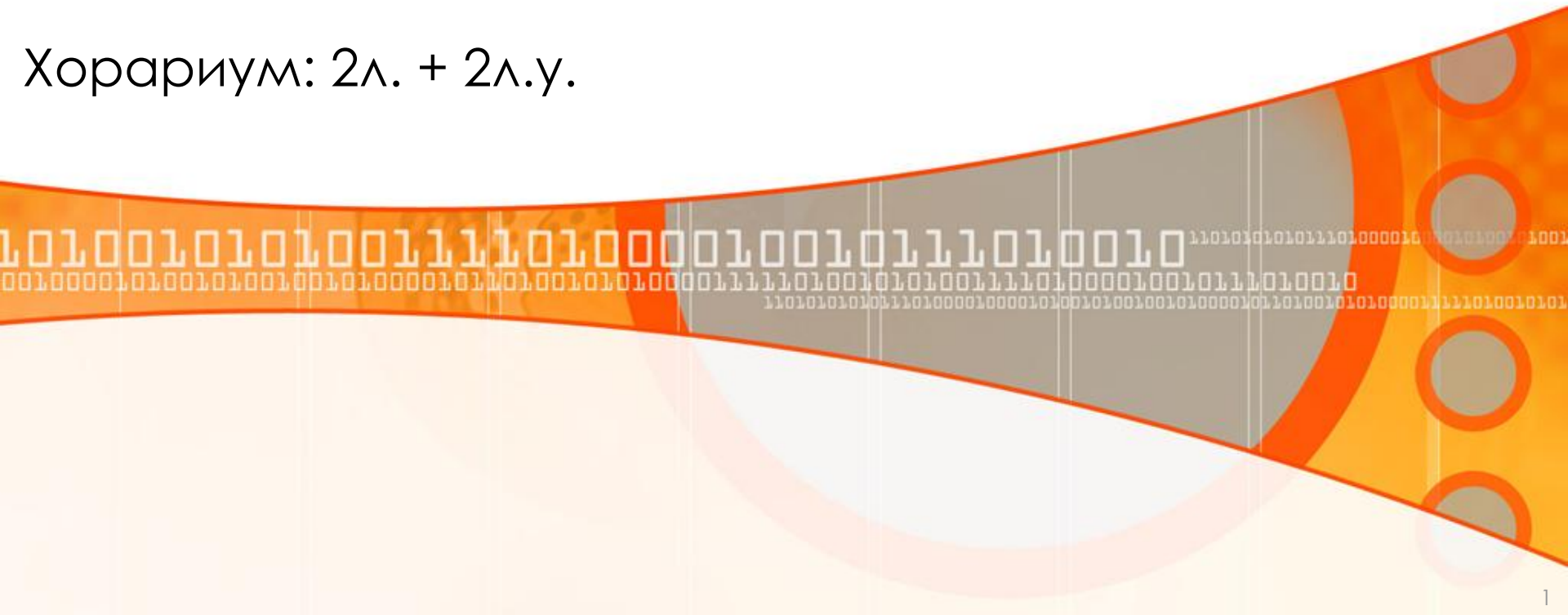


# Организация на компютъра

Хорариум: 2л. + 2л.у.



# Лекция 12:

## Методи за адресиране на операндите



10100101010011110100001001011101001011010101011101000010010111010010  
001000010100101001001010000101101001010100001111010010101001110100001001011010010  
1101010101011101000010000101001001001010000101101001010100001111010010101

# Съдържание

Методи за адресиране на операндите.

Основни методи за адресиране на операндите –  
непосредствен, пряк, косвен, относителен, индексен

Структура на машинните команди и апаратура за  
реализация на методите за адресиране на операнди

Достойнства и недостатъци на методите за адресиране

# Организация на изчислителния процес

## Методи за адресиране

Адресите могат да се изобразяват **явно** или **неявно**.

Ето защо се използват още понятията **изпълнителен адрес** и **адресен код**.

**Изпълнителен адрес** е двоичната комбинация, която се зарежда в адресния регистър на оперативната памет, т.е. това е *истинският* (действителният или още *абсолютният*) адрес на клетката от паметта, в която се намира (или от която започва) операндът.

**Адресен код** е някаква двоична информация, отнасяща се за адреса на посочваната клетка. Адресният код съдържа *указател и начални данни* – списък от елементите  $\langle Sp., reg., const \rangle$ . Указателят ( $Sp$ ) показва определен начин (един от възможните) за преработка на началните данни ( $reg., const.$ ). Началните данни се намират в регистровата памет (РП) на процесора.

# Организация на изчислителния процес

## Методи за адресиране

Отделните функционалности, според които се извършва преработката, включително и случая, когато тя фактически липсва, се наричат **методи за адресиране**.

В резултат на преработката, т.е. на метода за адресиране, се получава **изпълнителният адрес**  $d$ .

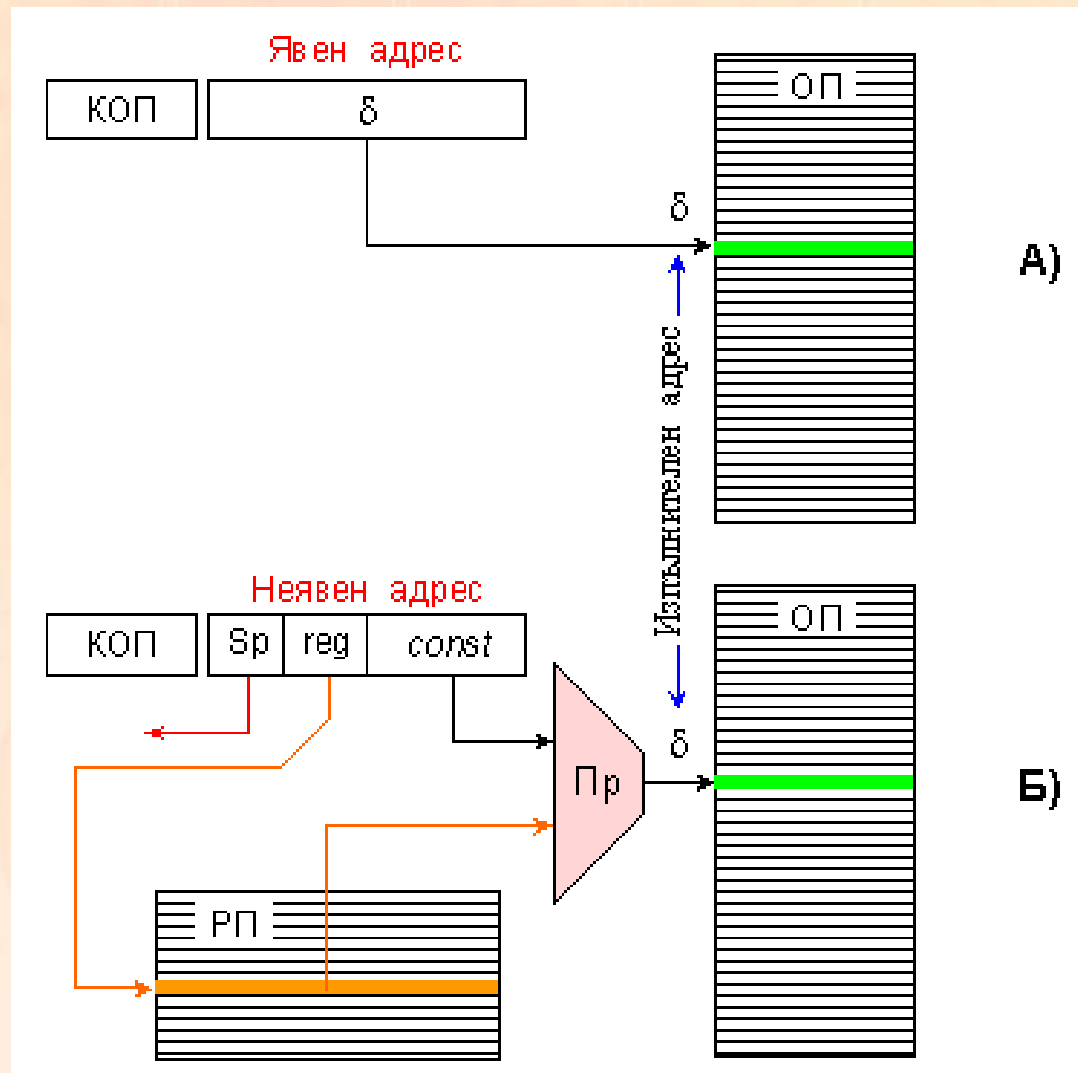
Когато адресът е неявен, адресният код не може да се зареди в адресния регистър на оперативната памет, тъй като не може да се използва непосредствено в качеството му на истински адрес, а трябва да се **преработи**, т. е. налага се да се използва **хардуерен преобразувател**.





# Организация на изчислителния процес

## Методи за адресиране



# Организация на изчислителния процес

## Методи за адресиране на операндите

### Безадресни команди

Това са команди, които **не съдържат** в структурата си адресна част. Те се представят само от кода на операцията (КОП).

а) команди **с подразбиращ се операнд**.

Това са команди, които изпълняват определена операция, в която операндът(-ите) се подразбира(-ат) от кода на операцията. Обикновено тези операции се наричат *унитарни*. Такива операции са например +1 (*increment*), -1 (*decrement*), нулиране и др., в които единият от операндите се явява съдържание на точно определен логически възел в структурата на процесора. Резултатът от операцията се явява новото му съдържание.



# Организация на изчислителния процес

## Методи за адресиране на операндите

### Безадресни команди

б) команди **с подразбиращ се адрес.**

Това са команди, които адресират (намират) операндите си в регистровата памет на процесора (например в стека).

Тези команди са много бързи.

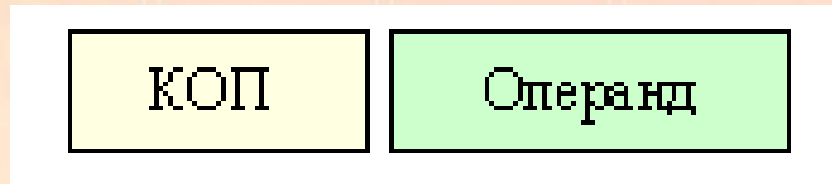
Те се изпълняват върху стандартните формати на данните, тъй като регистрите с общо предназначение обикновено имат дължината на разрядната мрежа.





## Непосредствено адресиране

Според този метод се приема, че командата не търси операнда, тъй като той се съдържа в самата нея, т.е. командата “носи” със себе си своя операнд.



Такъв операнд, който за нуждите на операцията се доставя от самата команда, се нарича **непосредствен**.

Непосредственият операнд се поставя (записва) в адресната част на машинната команда от програмиста при съставяне на програмата.

# Организация на изчислителния процес

## Методи за адресиране на операндите

### Пряко адресиране

Пряко адресиран е всеки операнд, чийто *абсолютен* (действителен) адрес се съдържа в адресната част на машинната команда.



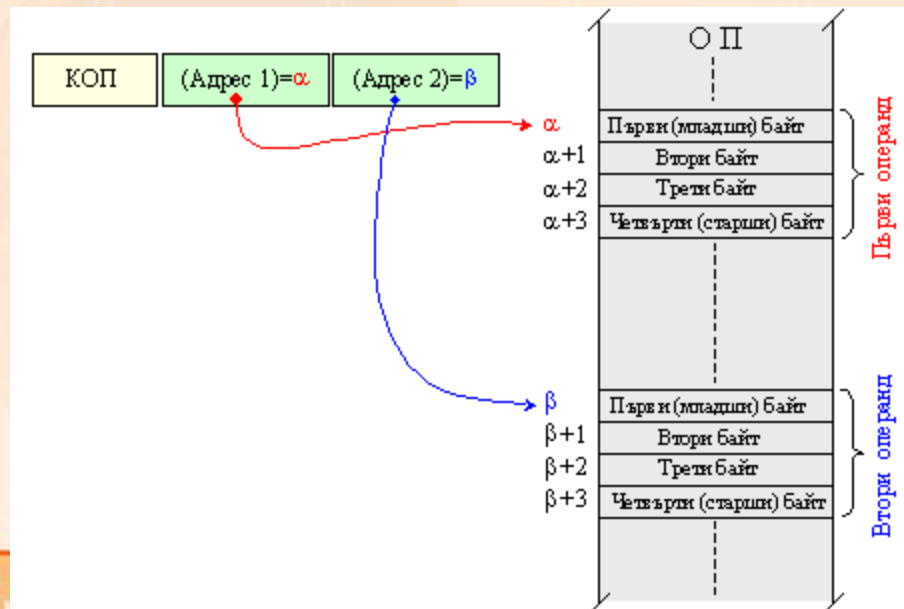
# Организация на изчислителния процес

## Методи за адресиране на операндите

### Пряко адресиране

При съставяне на програмата програмистът следва да запише в адресните полета на тази команда началните адреси на операндите – за първи операнд адрес **a**, а за втори операнд адрес **b**.

Дължината на операнда е без значение – началният му адрес е **единствен**.



# Организация на изчислителния процес

## Методи за адресиране на операндите

### Пряко адресиране

Командите с пряко адресиране са едни от най-бързо изпълняващите се.

В същото време прякото адресиране притежава един съществен **недостатък** - фиксираният в командата (в програмата) адрес на операнда е един и същ при всяко стартиране на програмата, което означава, че по място в адресното пространство на оперативната памет той е **непреместваем**.



# Организация на изчислителния процес

## Методи за адресиране на операндите

### Косвено адресиране

Според този метод в адресната част на командата не се намира нито самият операнд, нито неговият изпълнителен адрес, а **адресен указател**, който посочва къде се намира адресът на операнда.

Съдържащият се в командата адрес се нарича **косвен адрес**, той не е изпълнителен.

Адресът-препратка е *пряк* (действителен) адрес. Това прави този адрес непреместваем, но неговото съдържание може да бъде променяно преди всяко ново обръщение. Последното означава, че операндът придобива свойството да бъде свободен и преместваем. За да се получи достъп до операнда, указателят се използва за отваряне на клетка, в която се съдържа изпълнителният адрес.

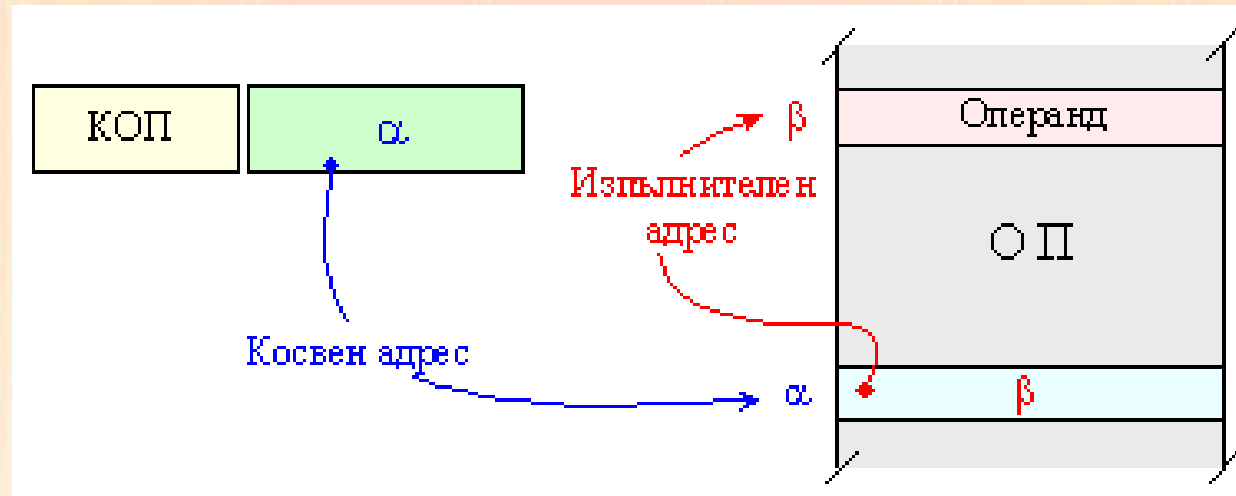




# Организация на изчислителния процес

## Методи за адресиране на операндите

### Косвено адресиране



# Организация на изчислителния процес

## Методи за адресиране на операндите

### Относително адресиране

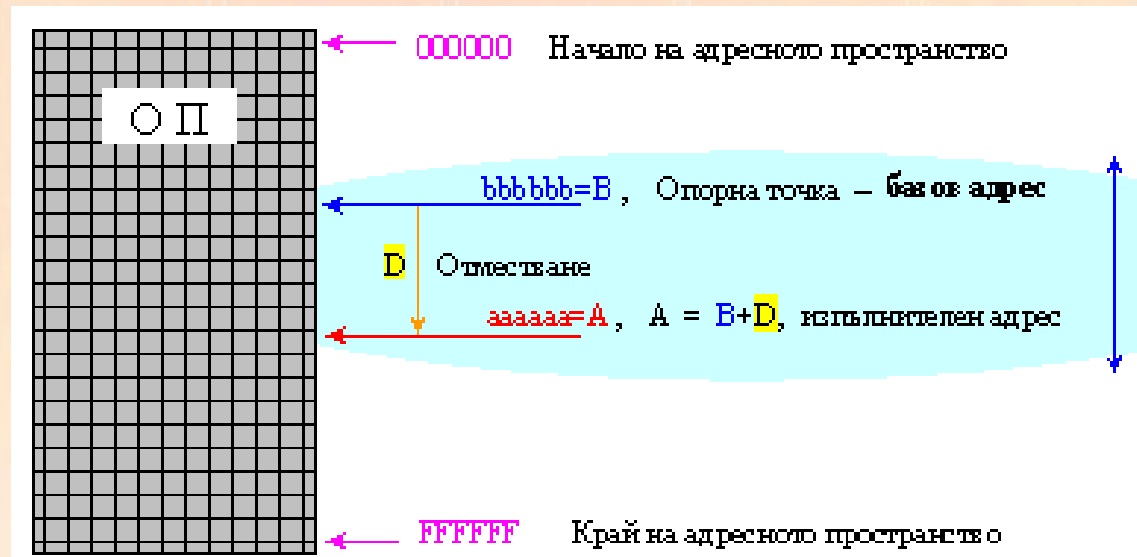
Този метод за адресиране изисква **допълнителна апаратура** за своята реализация и е наложен от изискванията за ефективно програмиране.

**Изпълнителният адрес**  $A_i$  на операндите при този метод се формира въз основа на закона  **$A_i = B + D$** ,  
където  $B$  се нарича база (базов адрес), а  $D$  (displacement) - отместване (или относителен адрес).  
Отместването се измерва относно базовия адрес в брой адресируеми единици.

# Организация на изчислителния процес

## Методи за адресиране на операндите

### Относително адресиране



Преместването на базовия адрес нагоре или надолу в адресното пространство не променя конструкцията на изпълнителния адрес.

# Организация на изчислителния процес

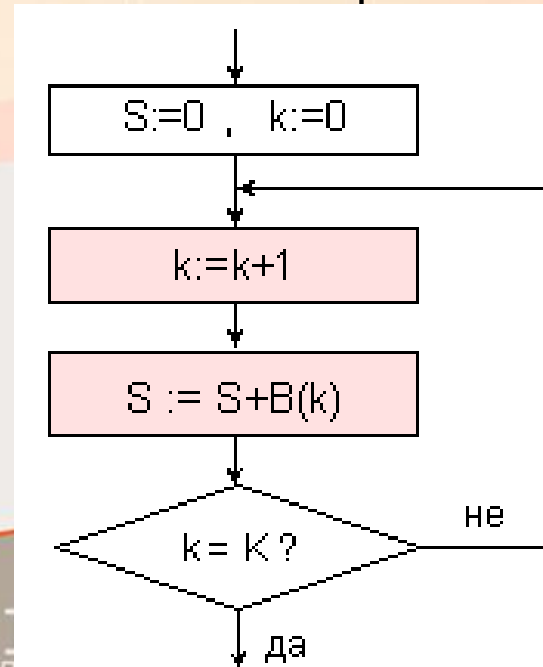
## Методи за адресиране на операндите

### Индексно адресиране

Това е единственият метод за адресиране, който е разработен по причина на алгоритмични (софтуерни) проблеми.

Предназначен е за адресиране на **еднотипни данни (масиви)**, които са разположени в последователни клетки от паметта. Подреждането им се изразява чрез техния **индекс**.

Изчисляването на изпълнителния адрес на текущия операнд на машинната команда трябва да се извършва *вън* от програмата, т.е. **с апаратни средства**.



# Организация на изчислителния процес

## Методи за адресиране на операндите

### Индексно адресиране

Техническата реализация на преадресирането се изразява във въвеждане в структурата на процесора на регистри със специално предназначение - **индексни регистри**, които да формират и да съхраняват индекса на текущия елемент, с чиято помощ се изчислява изпълнителният адрес.

В съвременните процесори индексните регистри са част от регистровата памет и са тяхна архитектурна характеристика.

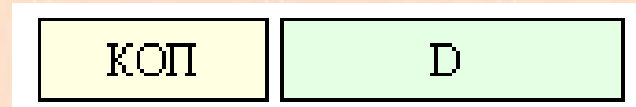
В тези регистри трябва да е възможно изпълнението на **унитарни операции**. Тези операции правят индексните регистри *програмно достъпни*. За целта в системата от машинни команди се разработва специална група команди, наричани *команди за работа с индексните регистри*.



# Организация на изчислителния процес

## Методи за адресиране на операндите

### Индексно адресиране



Структура на машинната команда при индексно адресиране

**Изпълнителният адрес** на операнда се формира по закона:  $A_i = (X) + D$ ,  $D$  – отместване

При изпълнение на програмата преадресирането се извършва в индексния регистър с отделна команда, която реализира закона  $X := (X) + k$ , където  $k$  е дължината на алгоритмичната стъпка в масива.

## Методи за адресиране на операндите

## Колко често се използват отделните методи за адресиране в реалните програми

	косвена	Базово индексна	регистрава	непосредствена	базова
Текст-редактор <i>TeX</i>	1%	0%	24%	43%	32%
CAD-система <i>Spice</i>	6%	16%	3%	17%	55%
C-компилятор <i>GCC</i>	1%	6%	11%	39%	40%

Оценки за три програмни продукта, масово използвани в компютри с архитектура VAX.

# Литература

- [1]. <http://tyanev.com/> - On-line книги – ОРГАНИЗАЦИЯ НА КОМПЮТЪРА – книга [1]
- [2]. <http://tyanev.com/> - On-line книги – ОРГАНИЗАЦИЯ НА КОМПЮТЪРА – упражнения книга [2];
- [3]. Димитър Тянев, ОРГАНИЗАЦИЯ НА КОМПЮТЪРА, том първи (ISBN 978-954-20-0412-7), Варна 2008г.
- [4]. Димитър Тянев, ОРГАНИЗАЦИЯ НА КОМПЮТЪРА - упражнения, ISBN 978-954-20-0258-0, Варна 2007г.

