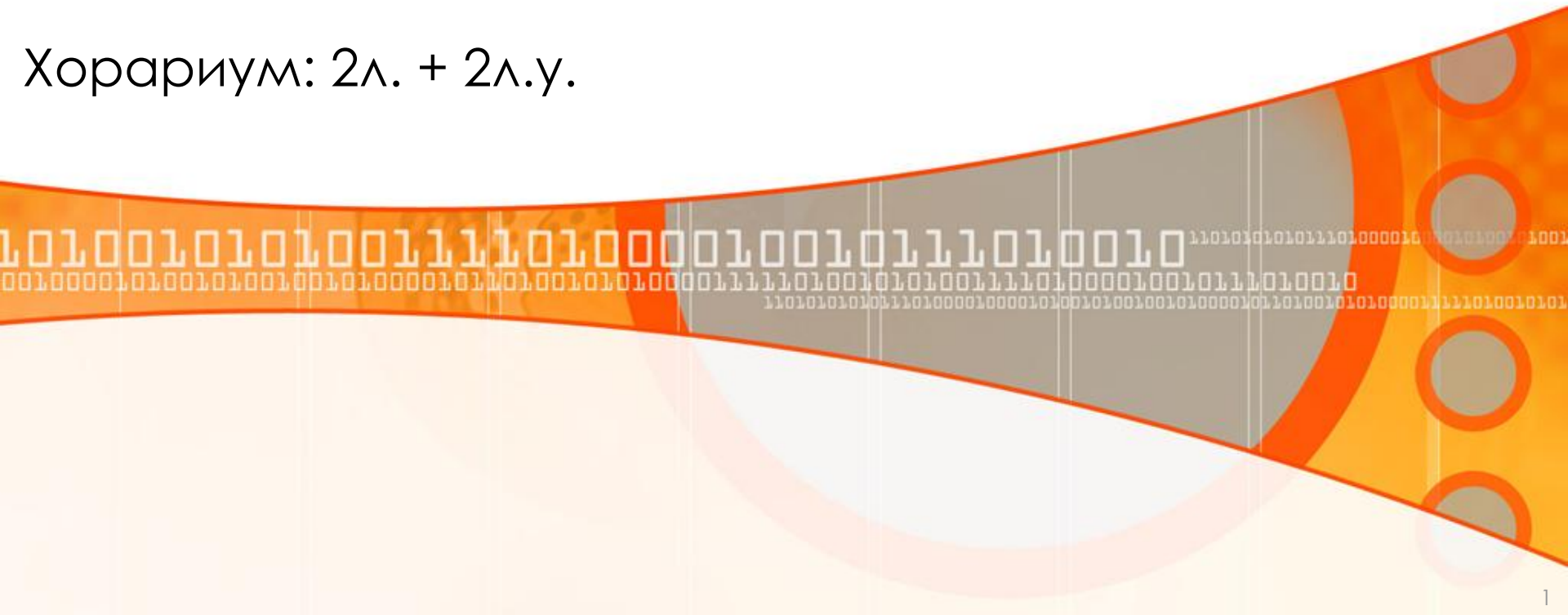


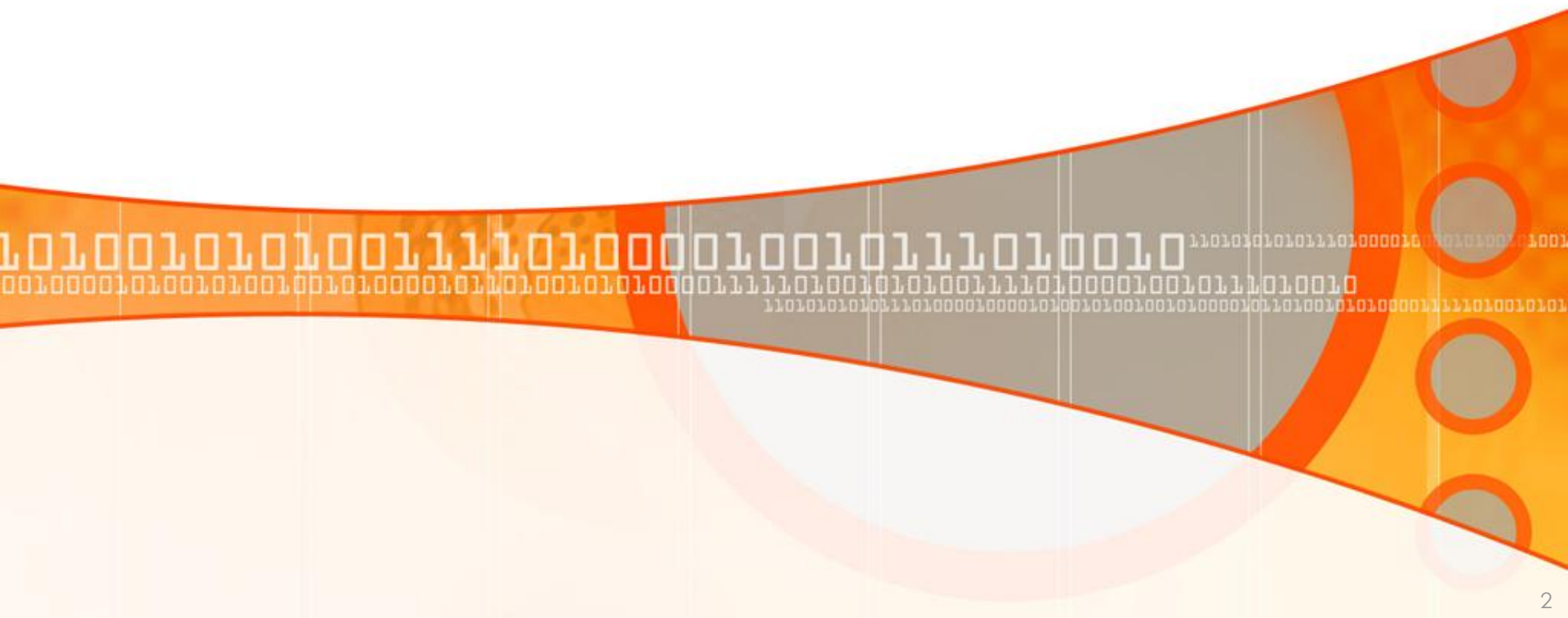
# Организация на компютъра

Хорариум: 2л. + 2л.у.



# Лекция 13:

## Система за прекъсване



# Съдържание

Прекъсване, система за прекъсване, предназначение и характеристики.

# Организация на изчислителния процес

## Система за прекъсване – общи положения

Процесорът е в състояние да реагира на външна намеса в изчислителния процес не в произволен момент, а само в момент, съответстващ на **границата** между два структурни елемента, т.е. **между две команди**, тъй като основният алгоритъм за управление на изчислителния процес е **КОМАНДНИЯТ ЦИКЪЛ**.

В този смисъл, възможността на процесора за ответна реакция е **първично заложена** в него чрез условия блок за проверка на заявките за прекъсване.

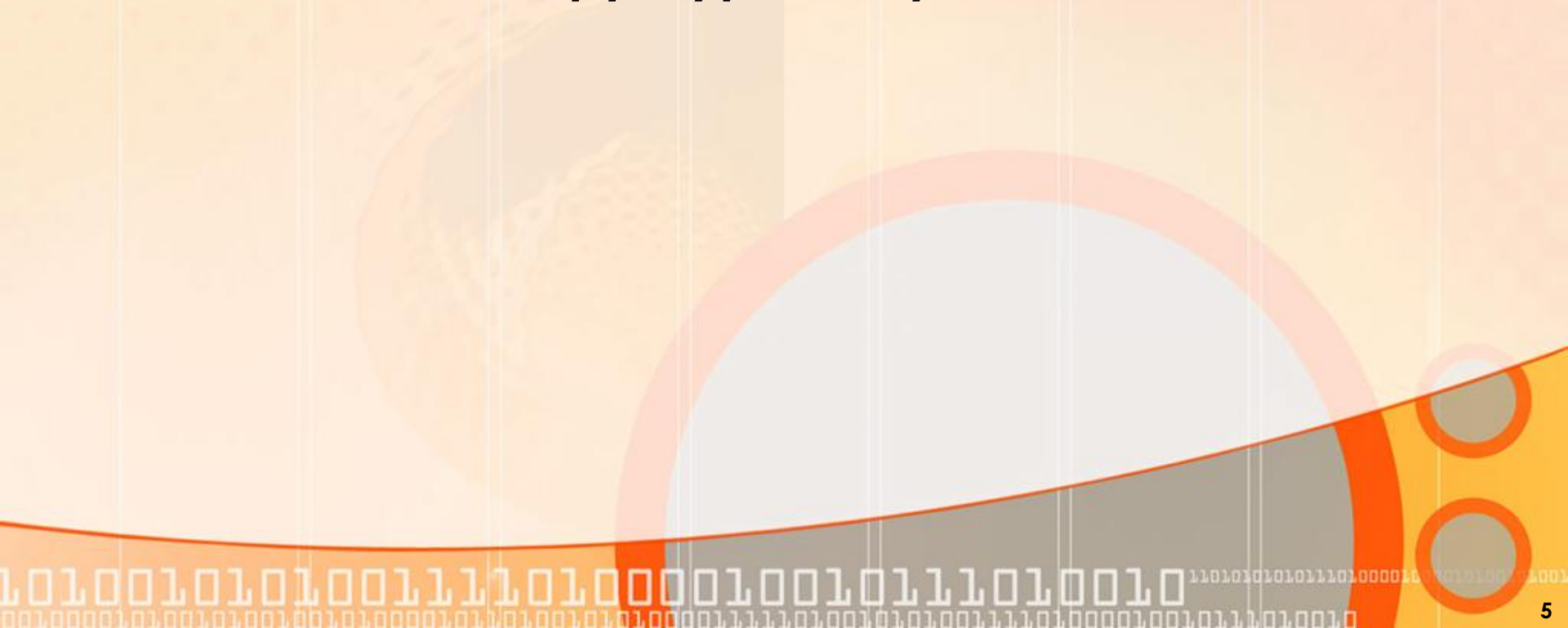


# Организация на изчислителния процес

## Система за прекъсване – общи положения

Успешното управление на реакцията на процесора в отговор на външни събития е свързано с понятието **състояние на процесора**.

Под състояние на процесора в даден момент най-общо се разбира **съвкупността от съдържанието на всички възли с памет в структурата му**.



# Организация на изчислителния процес

## Система за прекъсване – общи положения

Процесорът реагира на прекъсването чрез превключването му към друга програма, т.е. изчислителният процес, съответстващ на текущата програма, се прекъсва.

Прекъсването на изпълнението на текущата програма следва да се осъществи по такъв начин, че да бъде **възможно възстановяване** на изпълнението ѝ след отминаване на причинилите го събития и свързаните с тях действия, при това **от точката на прекъсването ѝ нататък**.



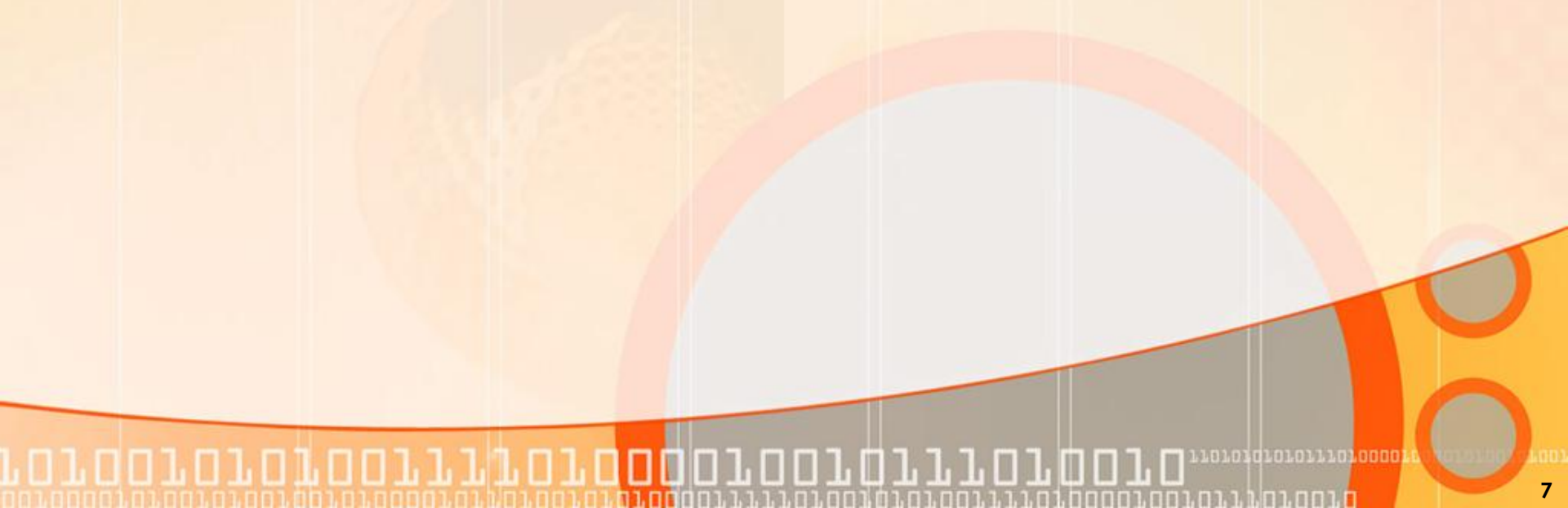


# Организация на изчислителния процес

## Система за прекъсване – общи положения

При прекъсване състоянието на процесора се представя от съдържанието само на онези възли, загубата на което е фатално за възобновяване на прекъснатия изчислителен процес от точката, в която го е заварило външното събитие.

Тази най-необходима информация обикновено се нарича **дума за състоянието** или **вектор на състоянието**.



# Организация на изчислителния процес

## Система за прекъсване – общи положения

Типичната за думата на състоянието информация се формира от:

- ✓ **Съдържанието на програмния брояч**, който съхранява текущия адрес. Текущият адрес е функция на командния цикъл и в настоящия случай представлява маркер за точката на прекъсване;
- ✓ **Съдържанието на адресните регистри**, съхраняващи базови, косвени и индексни адреси;
- ✓ Съдържанието **на регистрите с общо предназначение**, съдържащи междинните резултати;
- ✓ Съдържанието на **регистъра на признаците**;
- ✓ Други **възли със служебна информация**, отнасяща се до текущия режим на работа на процесора.





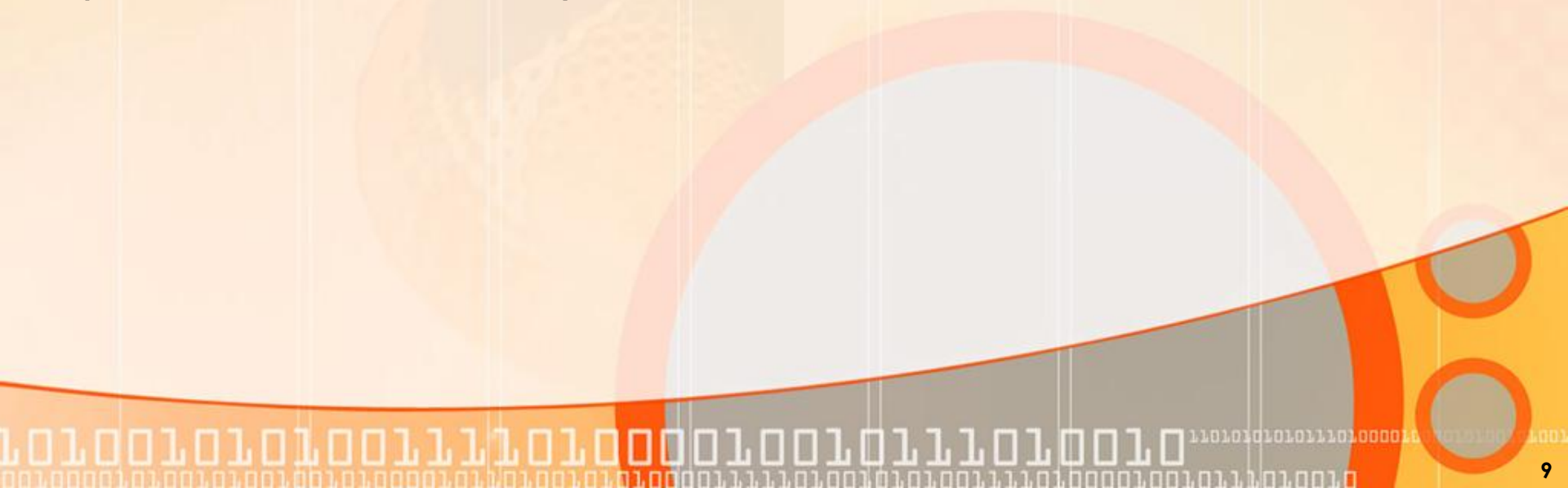
# Организация на изчислителния процес

## Система за прекъсване – общи положения

Принципно най-важното в разглеждания проблем е **случайният** характер на събитията, които могат да прекъснат текущите изчисления.

Този случаен характер ги прави **непредсказуеми** както по време, така и по вид.

По тази причина реакцията на процесора, т.е. на потребителя, **не може** да бъде заложена (запрограмирана) предварително в програмите.

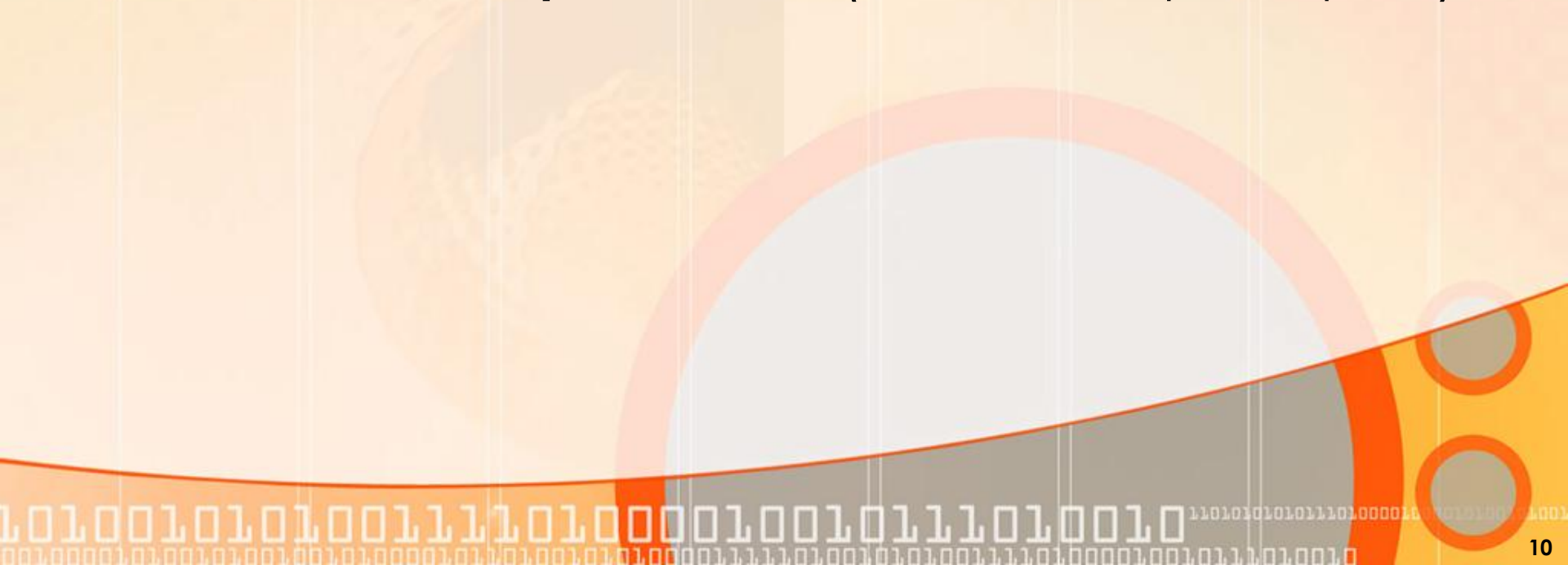


# Организация на изчислителния процес

## Система за прекъсване – общи положения

Възможността за реакция на процесора е предвидена в командния цикъл, а настъпващите събития, изискващи вниманието на процесора, са длъжни за целта да подават сигнали, наричани **заявки за прекъсване**.

Заявките за прекъсване трябва да се подават към централното управляващо устройство на специален вход – **заявка за прекъсване** (IRQ – Interrupt Request).



# Организация на изчислителния процес

## Система за прекъсване – общи положения

Възможността за прекъсване на работата на една програма, изпълнението на определени действия, свързани с настъпилото събитие и връщане на управлението обратно на изходната програма, без при това програмистът да е ангажиран явно в управлението на този процес, е важно **архитектурно свойство** на съвременните изчислителни машини.

Това свойство се постига чрез една съвкупност от специални за целта апаратни и програмни средства, която се нарича **система за прекъсване**.

# Организация на изчислителния процес

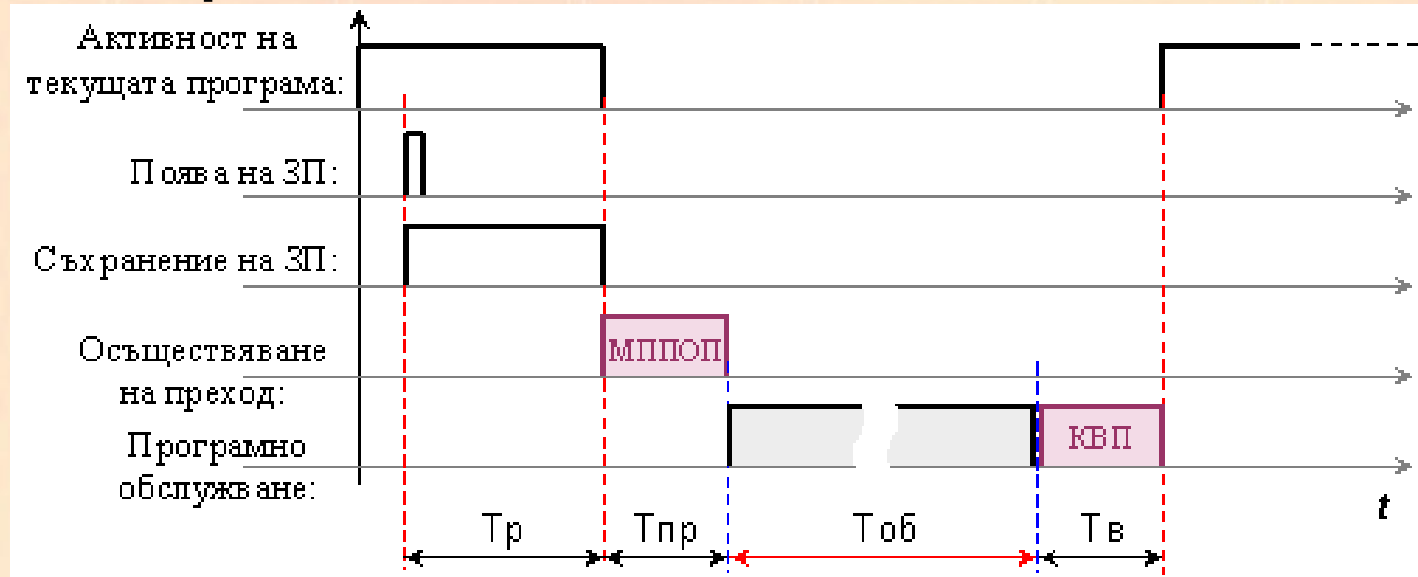
## Система за прекъсване – общи положения

Основните **функции** на системата за прекъсване са:

- ✓ Регистриране на заявките за прекъсване и избор на заявка за прекъсване;
- ✓ Запомняне на състоянието на прекъснатата програма и автоматичен преход към програма, обслужваща източника на заявката за прекъсване;
- ✓ Възстановяване на състоянието на прекъснатата програма, ако обслужващата програма вземе решение за това.

# Организация на изчислителния процес

## Система за прекъсване – общи положения



МППОП (микропрограмна процедура за осъществяване на прекъсването)

$T_r$  - време за реакция на системата

$T_{пр}$  – време за осъществяване на преход

$T_{об}$  – време за обслужване

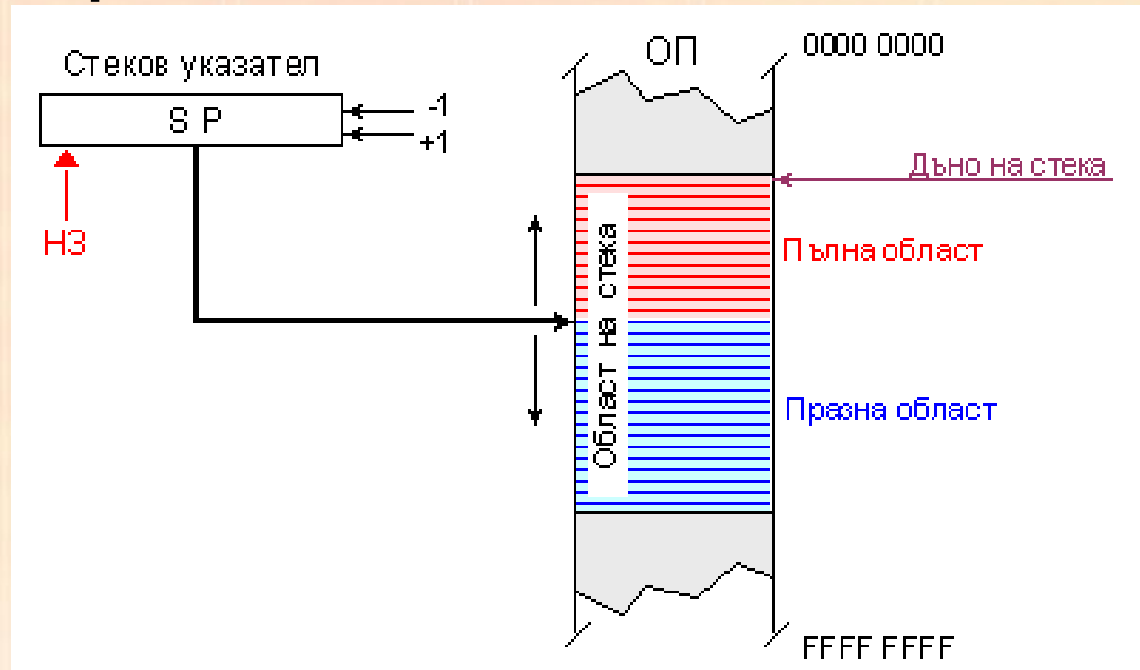
$T_v$  – време за възстановяване (връщане от прекъсване)

КВП – команда за връщане от прекъсване



# Организация на изчислителния процес

## Система за прекъсване – общи положения



### Програмен стек

**Програмният стек** е основно средство за реализация на прекъсването.

Стекът се реализира като LIFO-памет, чийто адресен регистър се нарича стеков указател (*SP – Stack Pointer*), а запомнящият масив на това запомнящо устройство се реализира като се “арендува” част (област) от основната RAM-памет на системата.

# Организация на изчислителния процес

## Система за прекъсване – общи положения

Определяне на мястото (началния адрес) и обема на стека е задача на операционната система.

Началният адрес на областта, избрана за стека, се нарича **дно** на стека.

Зареждането (H3) на стековия указател с начален адрес също е задача на операционната система. Това тя прави при зареждане на ново задание.

Програмният стек обикновено е стек *винаги готов* за запис, т. е. във всеки момент съдържанието на SP сочи първата свободна за запис клетка.



# Организация на изчислителния процес

## Система за прекъсване – общи положения

**Обслужващата програма** минава през следните етапи:

1. Съхранение (при нужда) на допълнителна информация за прекъсваната програма и определяне на новото състояние на процесора чрез зареждане на собствените параметри на обслужващата програма;
2. Анализирание на състоянието на източника, подава ЗП, и вземане на решение за конкретни действия;
3. Изпълнение на конкретните действия;
4. Възстановяване на параметрите на прекъснатата програма;
5. Изпълнение на операция "връщане от прекъсване".

# Организация на изчислителния процес

## Система за прекъсване – общи положения

Какъв е общият брой възможни за подаване заявки за прекъсване и какво е тяхното отношение спрямо изчислителния процес?

Колкото броят е по-голям, толкова по-гъвкава и развита може да бъде периферната система на процесора.



# Организация на изчислителния процес

## Система за прекъсване – общи положения

Максимално възможна **дълбочина на вложените прекъсвания**, която системата за прекъсване може да поддържа - това е максималният брой програми, които могат да стоят прекъснати, без да е загубена възможността да бъдат завършени.

Ако например, след прехода към обслужващата програма и по време на нейното цялостно изпълнение, приемането на други ЗП е забранено, това определя дълбочината на единица.

Дълбочината е цялото число  $N$ , ако е възможно да бъдат взаимно прекъснати  $N$  на брой програми.



## Обслужване на вложени прекъсвания



# Организация на изчислителния процес

## Видове прекъсвания

1. *Вътрешни* прекъсвания (програмни и апаратни) ;
2. *Външни* прекъсвания.

Според вероятните за поява **причини**, заявките за прекъсване се класифицират в три основни групи:

1. *Външни* прекъсвания:
  - От периферни устройства, изискващи обслужване;
  - От таймер;
2. *Аварийни* прекъсвания:
  - Поради грешки при изпълнение на програмите - при несъществуващ КОП, при аритметични операции, при обръщение към паметта, при защита на операционната система;
  - От диагностичната система - при загуба на информация в предаваните данни, при отпадане на захранването, при нарушаване на работните условия на отделни ИС.

# Организация на изчислителния процес

## Видове прекъсвания

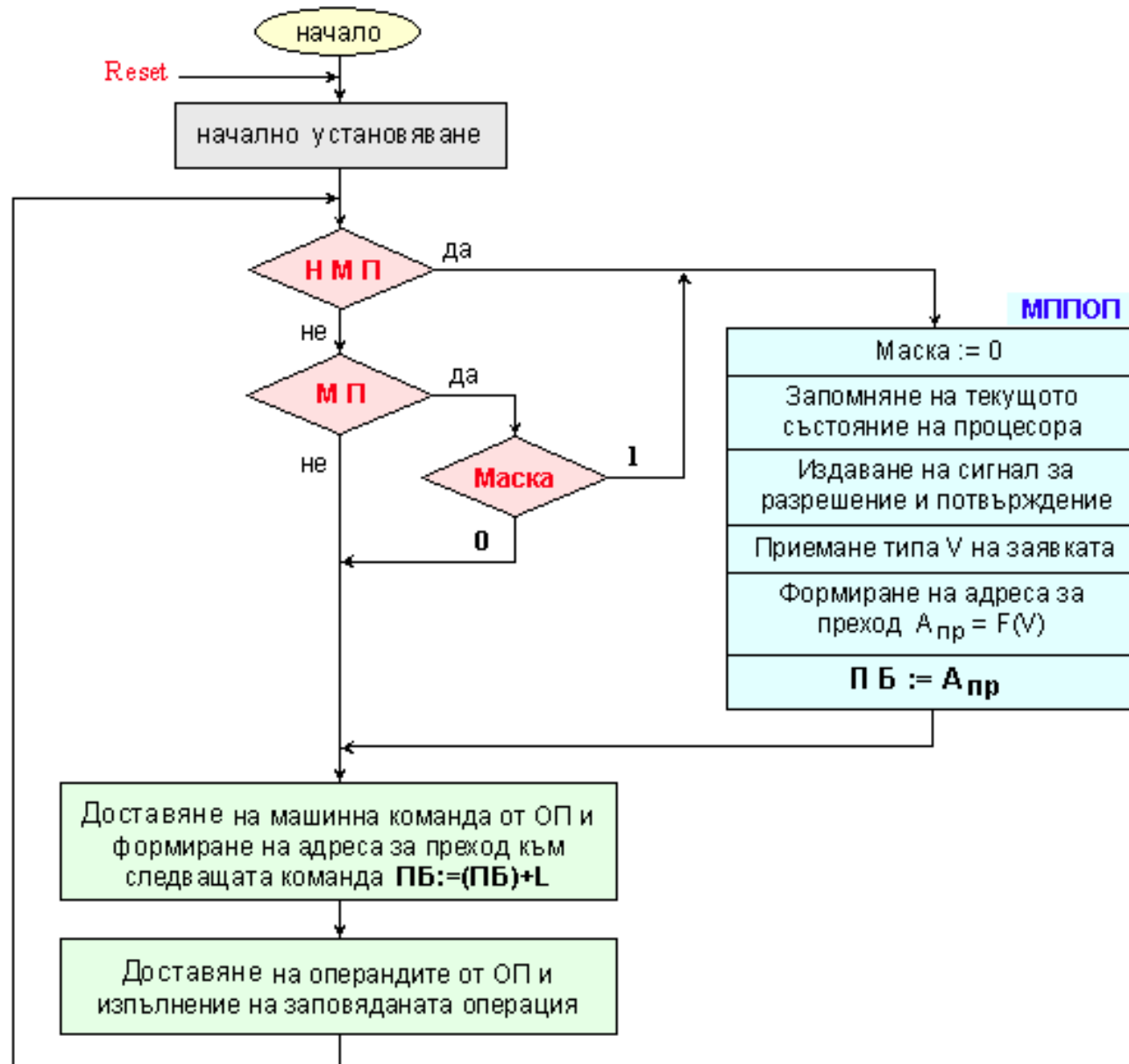
3. Програмни прекъсвания – при преход към подпрограма или към процедура, при преход към друго задание (процес), при връщане от прекъсване.

Според изискванията си за обслужване, прекъсванията могат да се класифицират в две групи:

1. Прекъсвания, които изискват безусловно обслужване – наричат се **немаскируеми (NMI)**;
2. Прекъсвания, които се обслужват при определени условия – наричат се **маскируеми (IRQ)**.

# Организация на изчислителния процес

## Команден цикъл и МППОП



# Организация на изчислителния процес

## Команден цикъл и МППОП

Маската може да се управлява според нуждите на потребителя.

За целта в командната система на всеки процесор се реализират машинни команди за присвояване стойност на маската.

В командната система на процесорите IA-32 това са двойката безадресни команди **CLI** (изчистване на маската), т.е. “забрани външни прекъсвания” и **STI** (установяване на маската), т.е. “разреши външните прекъсвания”.

Стойността на маската се съхранява в тригер, който обикновено е включен в състава на регистъра на състоянието на процесора.



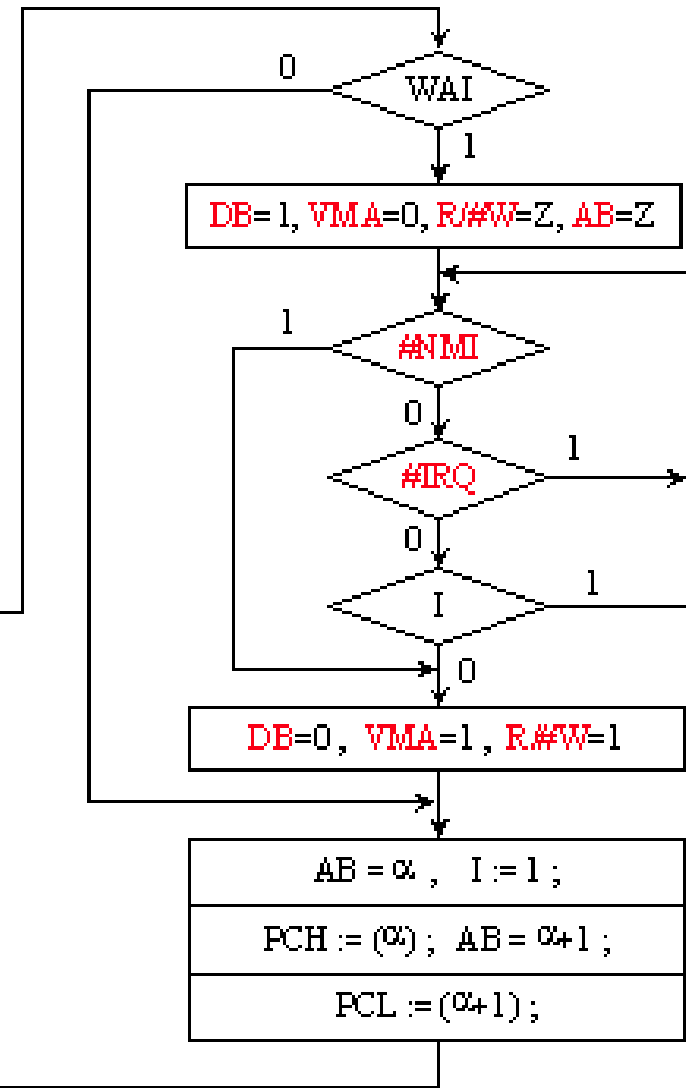


# Организация на изчислителния процес

## Команден цикъл и МППОП - МППОП на 8-битови микропроцесори *Motorola*

От командния цикъл.  
При условие че:  
 $NMI \cup (IRQ \cap \bar{I}) = 1$

$DB=0, VMA=1, R/\bar{W}=0,$ $SP := (SP) - 1, (SP) := (PCL);$
$SP := (SP) - 1, (SP) := (PCH);$
$SP := (SP) - 1, (SP) := (IXL);$
$SP := (SP) - 1, (SP) := (IXH);$
$SP := (SP) - 1, (SP) := (AccA);$
$SP := (SP) - 1, (SP) := (AccB);$
$SP := (SP) - 1, (SP) := (CCR);$



Връщане в командния цикъл

# Организация на изчислителния процес

## Команден цикъл и МППОП - МППОП на 8-битови микропроцесори *Motorola*

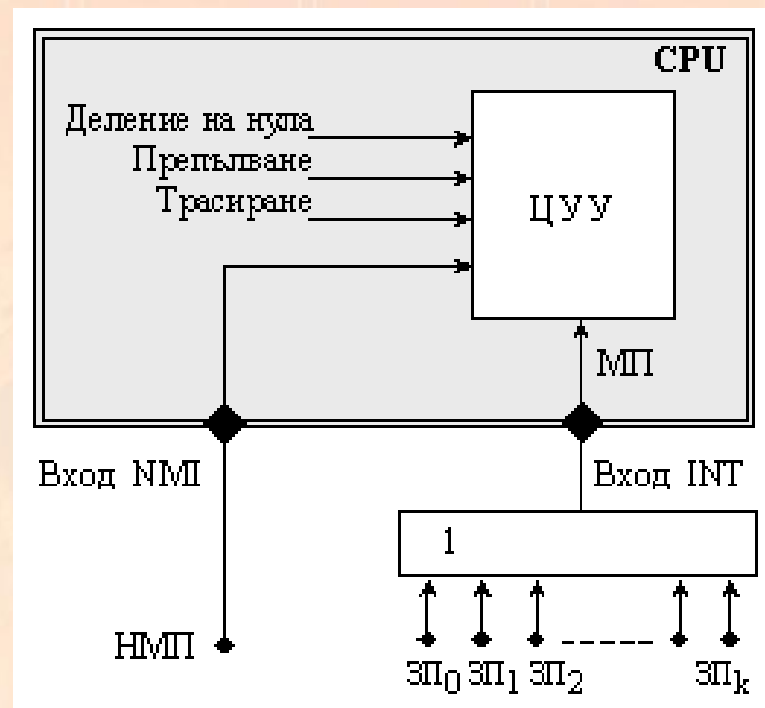
Характерното за тази твърде елементарна 8-битова архитектура е, че адресът  $\alpha$ , от който в края на МППОП се извлича новото съдържание на програмния брояч, се формира **апаратно** върху адресната шина и е **константен**. Стойността му зависи от вида на прекъсването. Началните адреси на обслужващите програми са постоянно записани в тези константни адреси :

Причина за прекъсване		Физически адрес	Начален адрес на обслужващата програма
IRQ , заявка за МП	$\alpha$	FFF8	Старши байт
	$\alpha+1$	FFF9	Младши байт
Команда SWI	$\alpha$	FFFA	Старши байт
	$\alpha+1$	FFFB	Младши байт
NMI , заявка за НМП	$\alpha$	FFFC	Старши байт
	$\alpha+1$	FFFD	Младши байт
Reset , сигнал за рестарт	$\alpha$	FFFE	Старши байт
	$\alpha+1$	FFFF	Младши байт

Разположение на векторите за прекъсване

# Организация на изчислителния процес

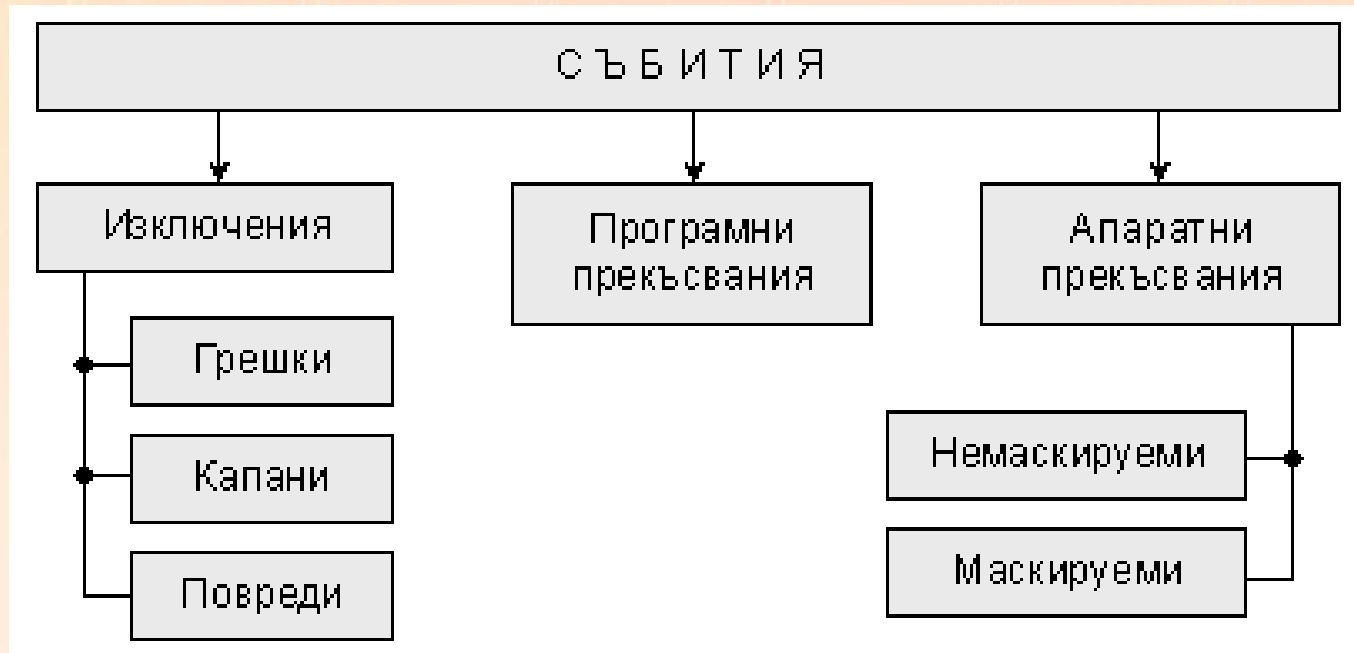
## Включване на външни и вътрешни ЗП



# Организация на изчислителния процес

## Видове прекъсвания

В съвременните компютърни системи се е наложило понятието *събитие*. Събитията основно се делят на вътрешни и външни. Вътрешните събития се наричат още *изключения*, а външните – *прекъсвания*.



# Организация на изчислителния процес

## Видове прекъсвания

Вътрешните изключения се делят на три вида:

1. Грешки (*faults*). Грешките се откриват преди или по време на изпълнението на машинната команда. Най-често се явяват в системата на виртуалната памет, когато е заявен адрес към несъществуваща страница или сегмент. В този смисъл изпълнението на командата е опорочено и с нея може да се постъпи по два възможни начина – да се рестартира или да се организира нейното продължение.



# Организация на изчислителния процес

## Видове прекъсвания

2. Капани (*traps*). Тази ситуация възниква на границата между две команди. Капаните са контролни точки, които могат да бъдат залагани умишлено от потребителя, с цел контрол на изчислителния процес, обикновено при тестване на софтуера, чрез вмъкване в кода на програмата на еднобайтова команда **INT**.

# Организация на изчислителния процес

## Видове прекъсвания

3. Повреди (*aborts*). При такива ситуации не винаги се разполага с адреса на командата, по време на която е настъпило изключението. По тази причина не винаги е възможен рестарт на командата и следователно на хода на изчислителния процес. Типични изключения от този вид са различни апаратни грешки, откривани от контролиращи логически схеми, или при противоречиви (недопустими) стойности в системните таблици. След обработка на прекъсването изчисленията продължават нормално от следващата команда.

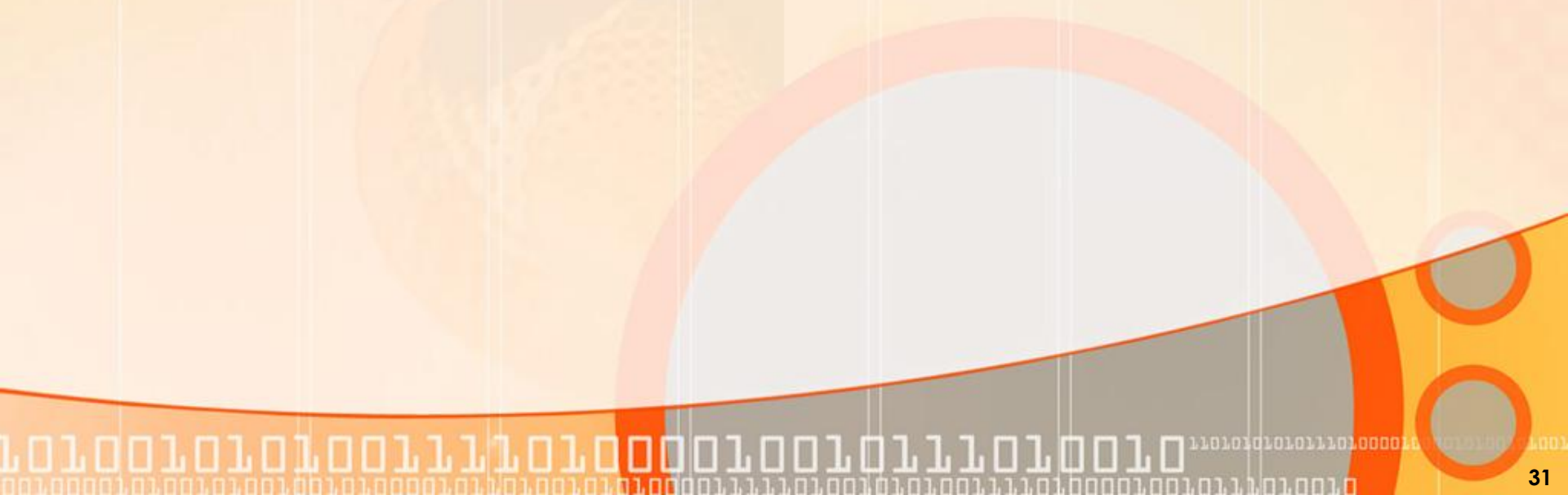
# Организация на изчислителния процес

## Видове прекъсвания

Заявките за обработка на изключенията се генерират от вътрешните схеми на процесора.

В рамките на една команда могат да съществуват причини за няколко изключения, но в даден момент ще възниква само едно изключение.

Процесорът ще обработва възникващите изключения последователно и ще рестартира машинната команда до нейното пълно изпълнение.



# Организация на изчислителния процес

## Обслужване на прекъсвания

## Обслужване на прекъсвания

Обслужването на немаскируемите заявки за прекъсване по определение трябва да бъде **безусловно**, но в същото време те са няколко и тяхното едновременно появяване не е изключено.

От това положение следва, че тяхното обслужване не може да бъде друго освен последователно и за целта следва да съществува някакъв ред.

Този ред е фиксиран от конструкторите на процесора.

Редът за обслужване на вътрешните и външните немаскируеми прекъсвания за всеки процесор е фиксиран в командния цикъл чрез съответното подреждане на условните им преходи.



# Организация на изчислителния процес

## Обслужване на прекъсвания

Ред, в който МППОП проверява всички едновременно възникнали изключения и прекъсвания:

1. Проверка за капан след току що завършилата команда (постъпков режим – трасировка, или прекъсване по стойност на резултата) ;
2. Проверка за грешки на следващата команда, според съдържанието в регистрите за трасиране на изпълнението;
3. Проверка за наличие на заявки за външни прекъсвания;
4. Проверка за грешка при липса на сегмент;
5. Проверка за грешка при липса на страница;
6. Проверка за грешка при дешифриране на командата;
7. Проверка за наличие на FPU;
8. Проверка за грешки при операции в FPU, ако проверката е включена.



# Организация на изчислителния процес

## Насищане на системата за прекъсване

Насищане настъпва в онзи момент, когато една ЗП застига все още необслужена заявка от същия тип. В този случай застигнатата заявка може да бъде загубена, което не е допустимо.

Насищането е параметър, който може да се свърже с максимално допустимата честота на следване на заявките за прекъсване, която е функция от параметъра  $T_r$  – време за реакция.

Влизането на системата за прекъсване в състояние на насищане води до зацикляне, от което е невъзможно да се излезе без допълнителна външна намеса.

Такова състояние е възможно при некоректно поведение от страна на външното устройство, подаващо заявка за обслужване.

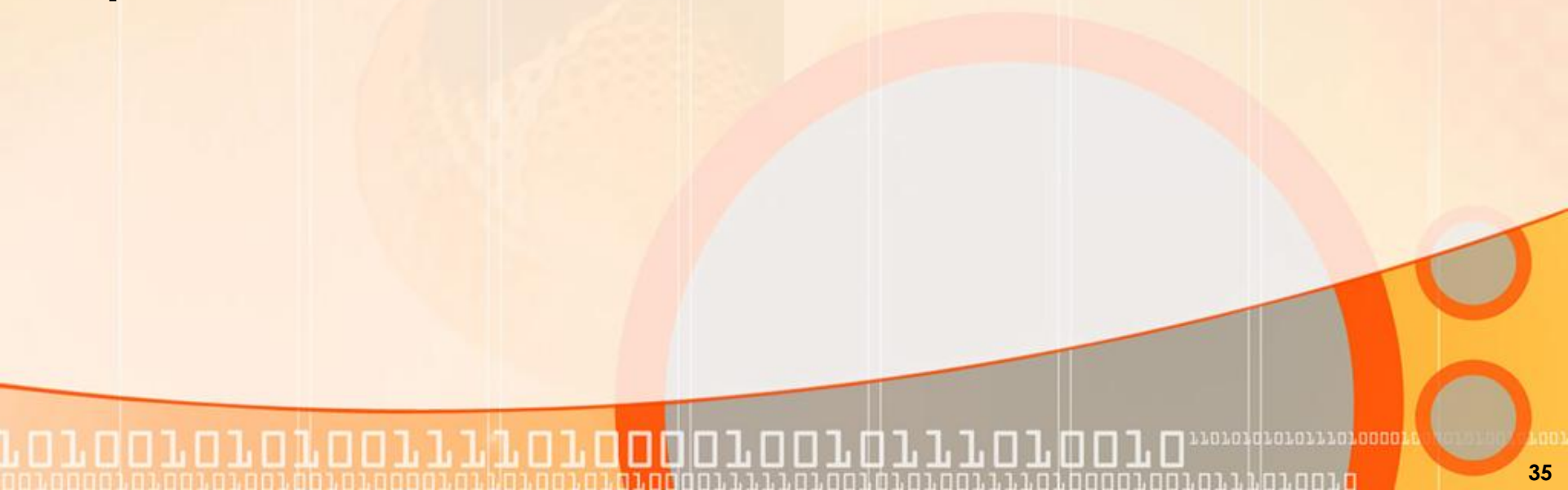
# Организация на изчислителния процес

## Обозначаване на прекъсванията

В съвременните компютърни системи възможните източници на заявки за прекъсване (или накратко – прекъсванията) получават **уникални номера**.

Използваните числа като номера на прекъсванията, са числата от естествения ред в интервала  $[0,255]$ , които като двоични имат дължина  $8[b]$ .

Номерът на прекъсването се нарича още **код за прекъсване**.

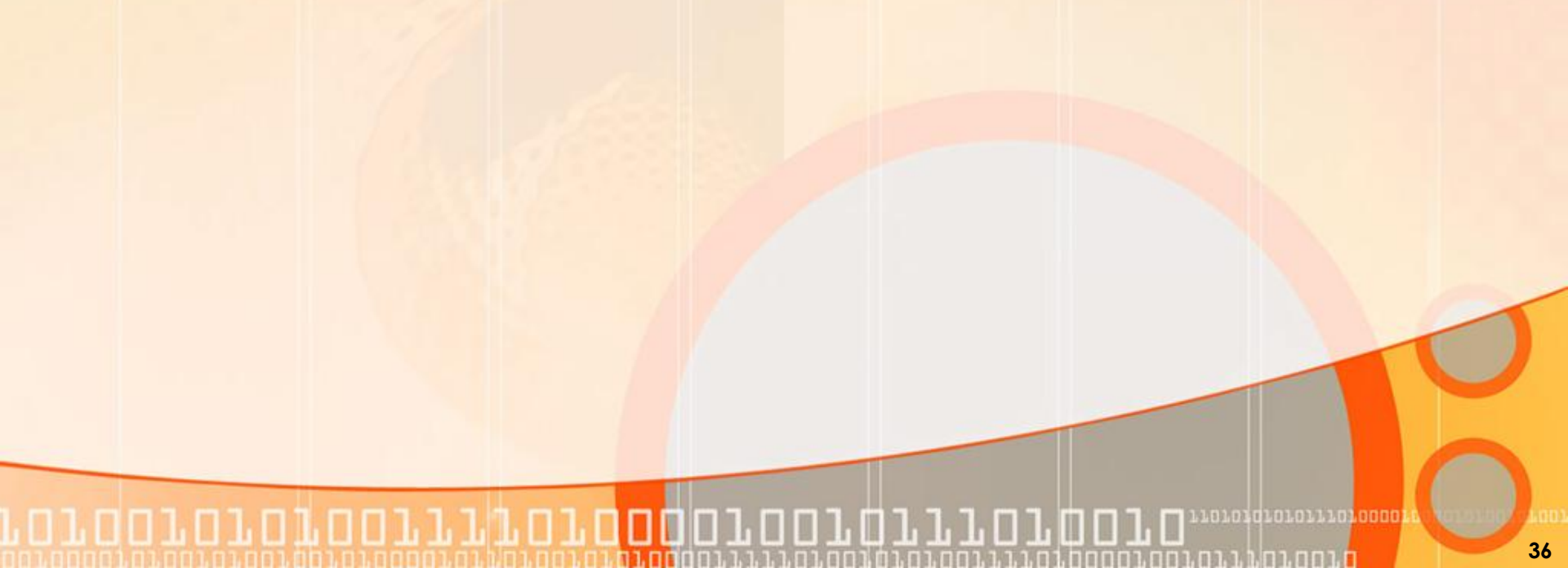


# Организация на изчислителния процес

## Обозначаване на прекъсванията

Общоприети конвенции за разпределение на номерата на прекъсванията.

Така например, за платформи с общо предназначение с архитектура IA-32, разпределението на номерата е известно и широко описано в литературата. За такива системи фирма *Intel* е резервирала номерата от 00 до 31 за обозначаване на изключенията.



# Организация на изчислителния процес

## Обозначаване на прекъсванията

№	Смисъл на събитието	Тип	Възможен рестарт	Код на грешката	Източник на събитието
0	Грешка при деление	Грешка	Да	Няма	Команди за деление DIV, IDIV.
1	При тестване	Грешка / Капан	Да	Няма	Всяка команда
2	Немаскируема ЗП (NMI)	Прекъсване	От следващата команда	Няма	Сигнал ЗП по вход NMI
3	Среща на контролна точка	Капан	От следващата команда	Няма	Команда INT 3
4	Препълване	Капан	От следващата команда	Няма	Команда INT 0
5	Нарушение на границите на масив	Грешка	Да	Няма	Команда BOUND
7	Недостъпно устройство	Грешка	Да	Няма	ESC, команда WAIT
16	Грешка при плаваща запетая	Грешка	Да	Няма	ESC, команда WAIT
32 ÷ 255	Прекъсвания, определяни от потребителя	Прекъсване	Да		Команда INT n или външен сигнал



# Организация на изчислителния процес

## Обозначаване на прекъсванията

Номерът на прекъсването е функционално свързан с началния адрес на съответната обслужваща програма –  $A_{\text{пр}} = F(V)$ .

Тази функционалност се реализира в микропрограмната процедура за осъществяване на прекъсването МППОП.

Функцията  $F(V)$  изразява преобразуването на номера на прекъсването  $V$  в табличен адрес  $TA$ ,  $TA = F(V)$ .

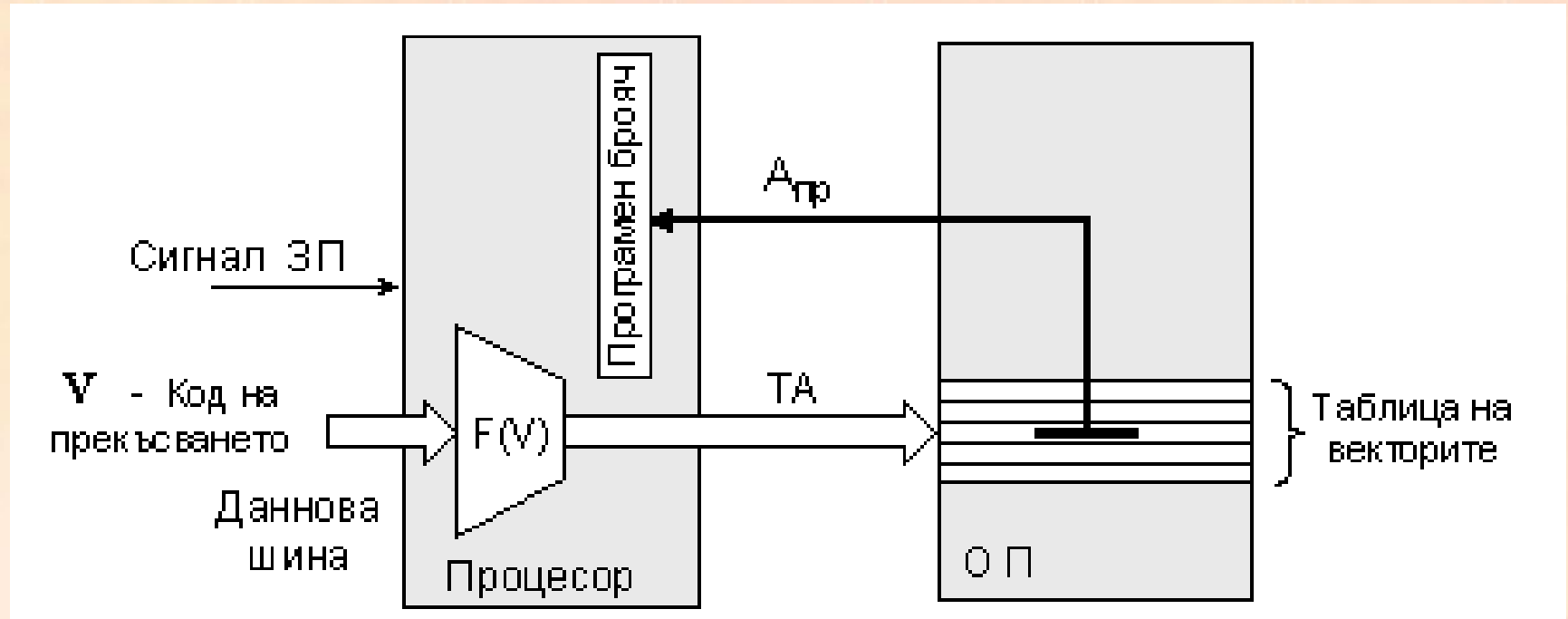
Този табличен адрес е косвен и неговото съдържание ( $TA$ ) следва да се извлече и използва като адрес за преход към съответната обслужваща програма.



# Организация на изчислителния процес

## Обозначаване на прекъсванията

Схемата на преход от сигнал към обслужваща програма



# Организация на изчислителния процес

## Проблемът “избор на заявка за прекъсване”

### Системи за реализация на избора

- **Програмен полинг**

Микропрограмната процедура за осъществяване на прекъсването е крайно опростена. При решение за обслужване на ЗП МППОП съхранява текущото състояние на процесора и зарежда в програмния брояч фиксиран адрес. Прекъснал текущата програма, процесорът стартира винаги една и съща програма – програмата "Полинг". Тази програма трябва по чисто програмен път:

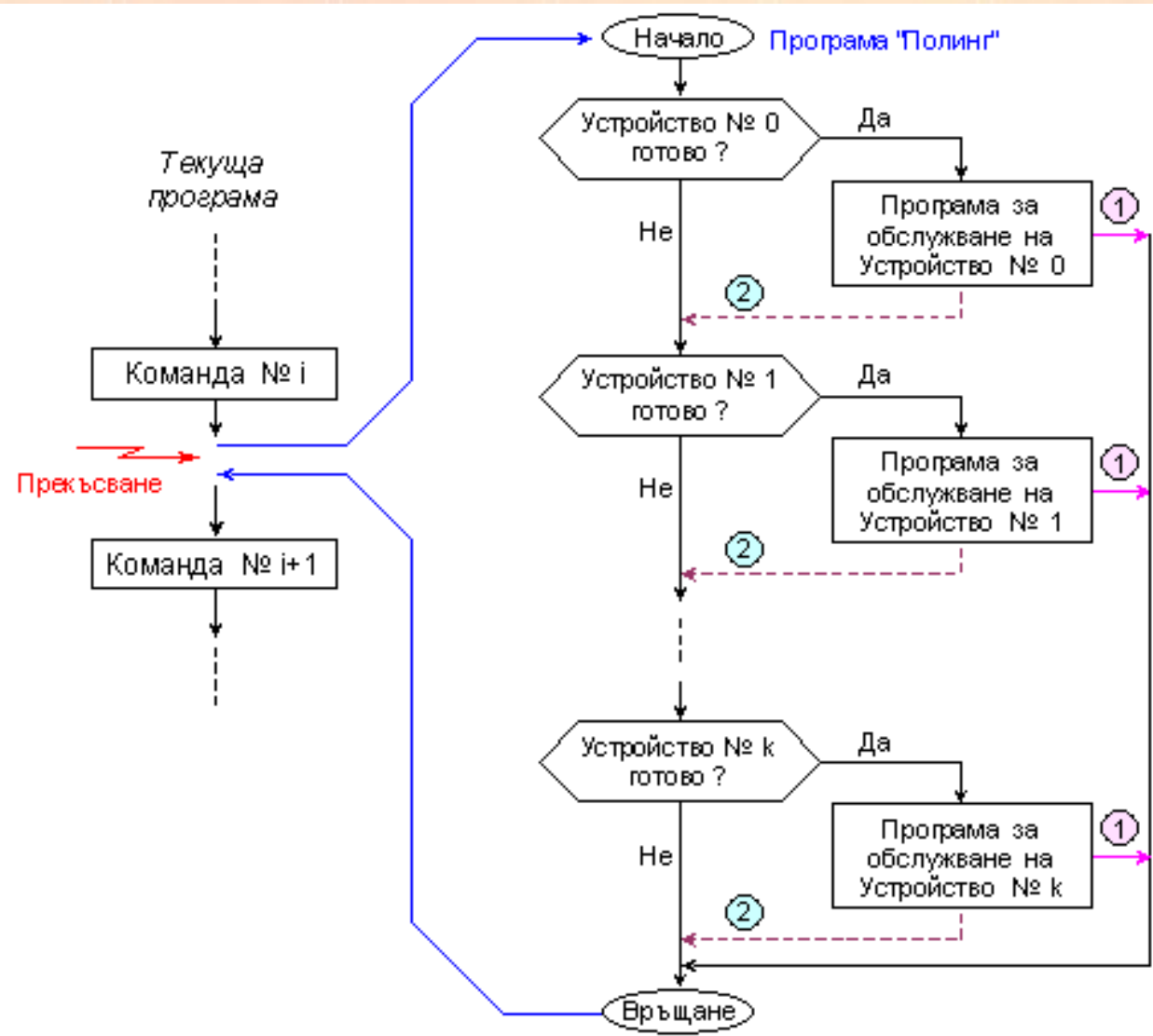
1. Да обходи всички източници на ЗП ;
2. Да избере най-приоритетния от тях ;
3. И да стартира съответстващата му обслужваща програма.

# Организация на изчислителния процес

## Проблемът “избор на заявка за прекъсване”

### Системи за реализация на избора

- Програмен полинг

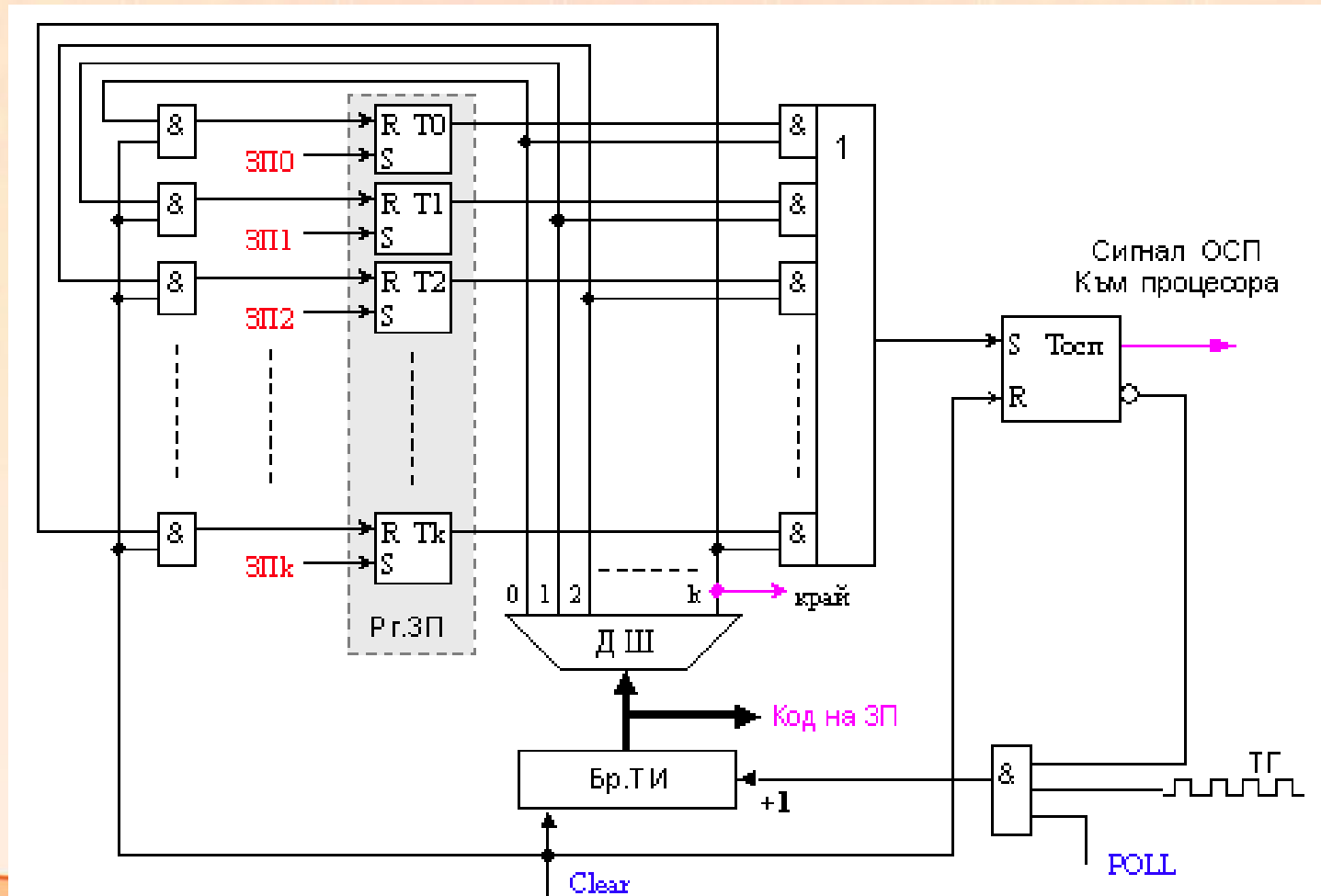


# Организация на изчислителния процес

## Проблемът “избор на заявка за прекъсване”

### Системи за реализация на избора

- Апаратен полинг



# Организация на изчислителния процес

Всички операции във външните устройства, които потребителската програма изисква, се изпълняват практически с помощта и под управлението на **операционната система**.

Тъй като работата на външните устройства се организира най-вече чрез системата за прекъсване, то следва, че операционната система обработва и всички прекъсвания.





# Организация на изчислителния процес

## Проблемът “избор на заявка за прекъсване”

- Въвеждане на **приоритет** (степен на значимост), който оценява както всяка отделна причина за прекъсване, така и характеризира всеки отделен вход за ЗП в системата за прекъсване.
- Приоритетът (оценката на степента на значимост) е **ЦЯЛО ЧИСЛО**.

# Организация на изчислителния процес

## *Plug-and-Play* върху шина PCI

Всяко PCI-устройство има така нареченото конфигурационно пространство. Това пространство е с обем от 256 байта. Зад всеки адрес от това пространство стоят още две конфигурационни полета – за устройство и за функция, така че адресът се интерпретира според следната структура:

**Адрес на шина [8 бита] :: Устройство [5 бита] :: Функция [3 бита]**

Това дава 256 шини, на всяка от тях могат да се мултиплексират до 32 устройства, всяко от които поддържа максимум 8 функции, които могат да бъдат адресирани и конфигурирани. Всяко PCI-устройство трябва да отговаря поне на функция номер 0. Първите 64 байта от конфигурационното пространство са стандартизирани, останалите са на разположение на производителите.



# Организация на изчислителния процес

## Технология Plug-and-Play (PnP)

Тази технология е разработена с цел да улесни потребителя при добавяне на нов хардуер към компютърната конфигурация. Предлага се хардуерът да предоставя на програмното осигуряване възможности за гъвкаво конфигуриране. Така от своя страна софтуерът може да променя голяма част от настройките на хардуера, който става напълно “ВИДИМ”.

Най-съществената част са **номерата на прекъсване и ВХОДНО-ИЗХОДНИТЕ адреси.**



# 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.

