ВХОДНО - ИЗХОДНА СИСТЕМА

- Входно-изходната система на компютъра осигурява интерфейса му с външния свят.
- Тя включва множество входно-изходни модули.
- Всеки В/И модул осигурява интерфейса към системната шина и управлява едно или повече периферни устройства.
- Компютърната система може да обхваща широк спектър от периферни устройства, всяко от които има различни електромеханични характеристики и различно бързодействие.

- Скоростта на трансфер при периферните устройства е значително по-малка от тази на процесора или на паметта.
- Затова те не се свързват към високо-скоростната системна шина, а към значително по-бавна входно-изходна шина.
- Освен това обикновено периферните устройства използват различен формат на данните и различна дължина на основните информационни единици от компютъра, към който те са свързани.

- Интерфейсът на В/И модул включва сигнали за управление, статус (състояние), данни и адреси. От една страна, той трябва да осигури интерфейса към компютърната система, а от друга страна – интерфейса към конкретното периферно устройство.
- Управляващите сигнали определят конкретната операция, която трябва да извърши периферното устройство като например, изпращане на данни към В/И модул (INPUT, READ), приемане на данни в периферното устройство (OUTPUT, WRITE), индициране на състоянието на периферното устройство или специфично действие (напр., позициониране на главата на диска).
- Състоянието на периферното устройство може да бъде например готово/неготово за трансфера на данни.

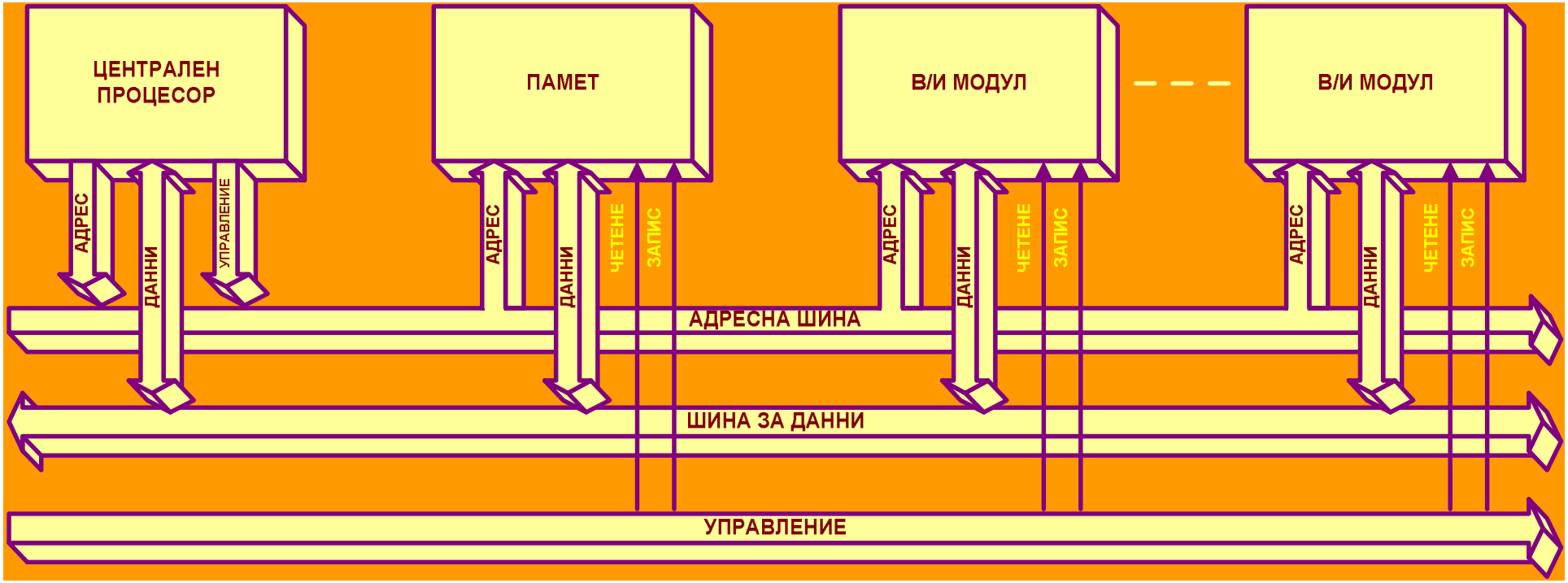
Основните функции на В/И модул са:

- Управление и синхронизация;
- Комуникация с процесора;
- Комуникация с периферното устройство;
- Буфериране на данните;
- Откриване на грешки.

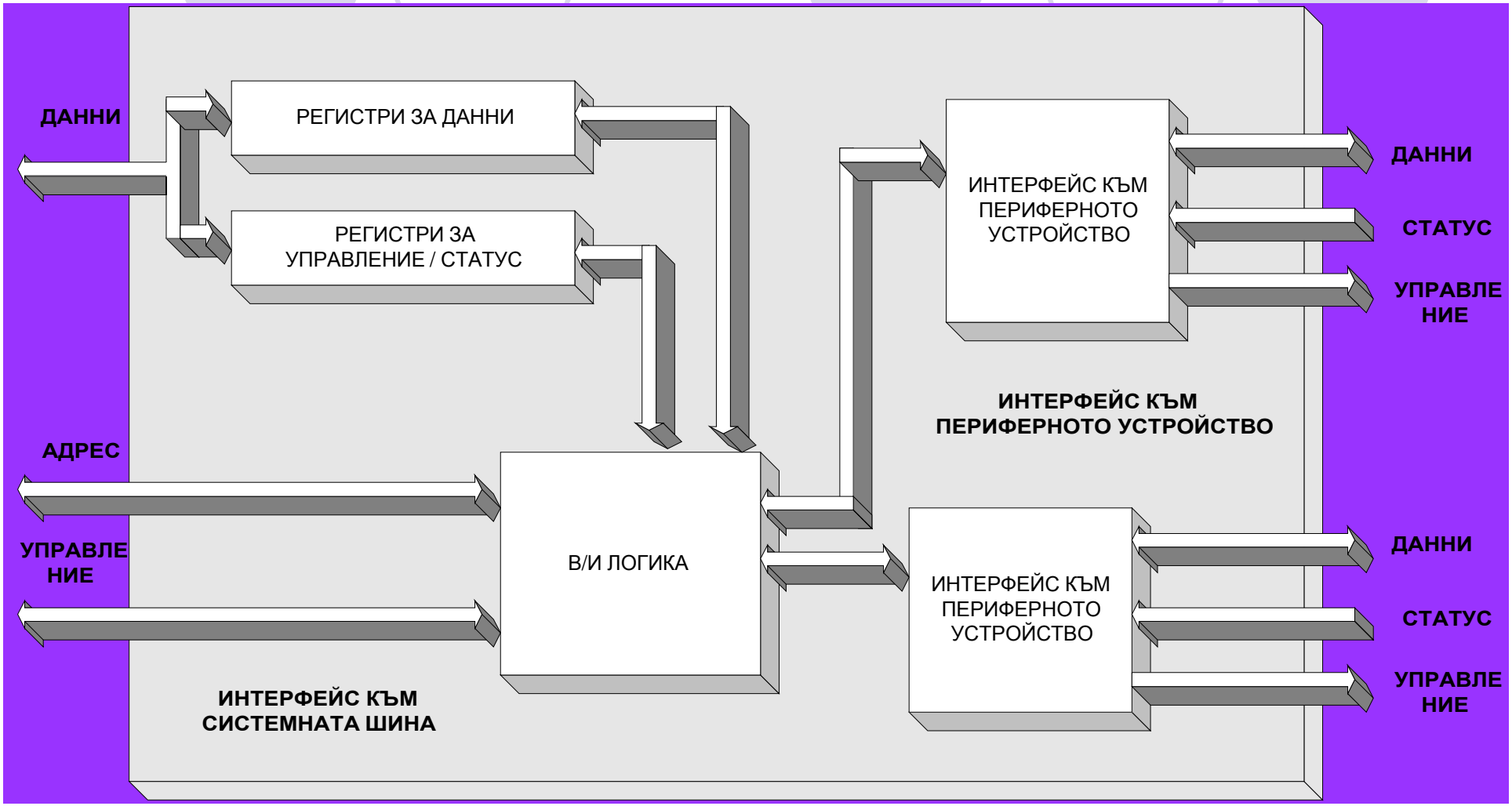
Комуникацията между процесора и В/И модул включва следните стъпки:

- *Декодиране на командата*. В/И модул получава команди от процесора под формата на сигнали по управляващата шина. Някои от тези команди са съпроводени от параметри, подавани по шината за данни. Така например, устройството на диска може да получи команда за четене на сектор, запис на сектор, и др.
- *Данни*. Обмена на данните между процесора и В/И модул се осъществява по шината за данни.
- *Индициране на статуса*. Поради ниското бързодействие на периферните устройства е важно да се знае във всеки момент какво е неговото състояние: заето с изпълнението на задача (BUSY), готов за изпълнението на задача (READY) или възникнала грешка (ERR).
- *Разпознаване на адреса*. В/И модул трябва да може да разпознава уникалните адреси на периферните устройства, които управлява.

КОМПЮТЪРНА СИСТЕМА С ОБЩА ШИНА



В/И модул



Обобщена структура на паралелен В/И порт

В/И процесор

- В/И модул, който осигурява интерфейс на високо ниво за централния процесор, се нарича *В/И канал* или *В/И процесор*.
- В/И модул, който е по-примитивен по отношение на структурата и действието си, и изисква по-детайлизирано управление, се нарича *В/И контролер* или *контролер на устройството*.
- В общия случай В/И контролери се използват в микрокомпютрите, а В/И канали – в по-мощните компютри.

Техниката на *програмирания вход-изход*

- процесорът изпълнява програма, която директно управлява изпълнението на В/И операция включително тестване на състоянието на устройството, изпращане на команда за четене или запис, и трансфера на данните. Тази техника води до големи загуби на процесорно време поради ниските скорости на работа на периферните устройства.
- Възможни са два режима на адресиране на В/И устройства: В/И адресно пространство заема част от адресното пространство на компютъра и изолирано В/И адресно пространство, отделно от адресното пространство на компютъра.
- Процесорът третира регистрите на В/И модули (управляващи/статус и за данни) като клетки от паметта и използва едни и същи машинни инструкции за достъп както до клетките в паметта, така и до В/И устройства.

В/И с прекъсване

- процесорът подава В/И команда за изпълнение към В/И модул, превключва към изпълнението на други задачи и получава сигнал за прекъсване от В/И модул, когато В/И обмен завърши
- В края на всеки инструкционен цикъл процесорът проверява постъпилите заявки за прекъсване.
- Когато е подадена заявка за прекъсване от В/И модул, процесорът съхранява контекста на текущата програма (програмния брояч и регистрите си) и обслужва прекъсването.
- Процесорът чете думата от В/И модул и я записва в паметта или обратното, след което възстановява контекста на прекъснатата програма и възобновява изпълнението й.

имплементиране на В/И с прекъсване

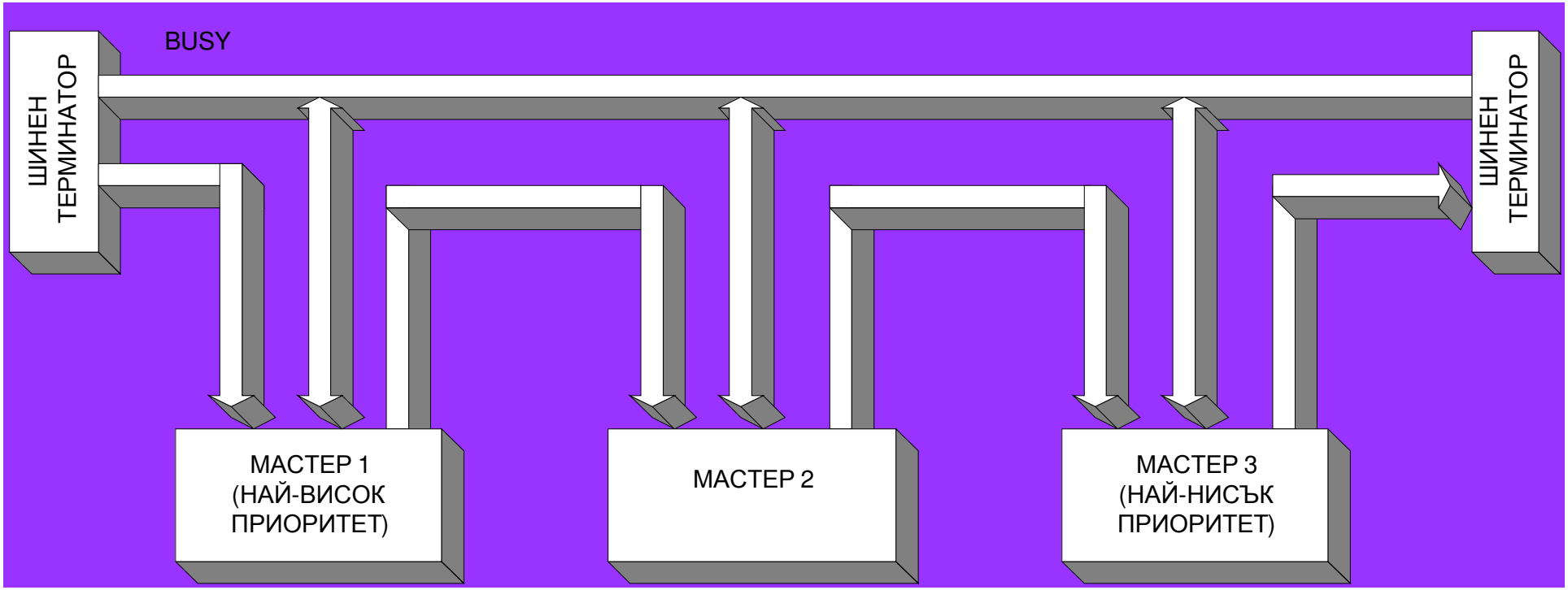
за идентифицирането на В/И модул, подал заявката за прекъсване, се използват няколко подхода:

- множество линии за прекъсване,
- софтуерно “запитване” (polling),
- хардуерно “запитване”
- шинен арбитраж.

софтуерно запитване

- когато процесорът получи заявка за прекъсване, той извиква обслужващата програма, която проверява последователно всички В/И модули за да установи кой В/И модул е подал заявка за прекъсване.
- Тази проверка се осъществява като се четат последователно регистрите на състоянието на В/И модули.
- След като се идентифицира В/И модул, предизвикал прекъсването, се извиква съответната подпрограма, която да го обслужи.
- Очевиден недостък на софтуерното запитване е голямото време за реализация.

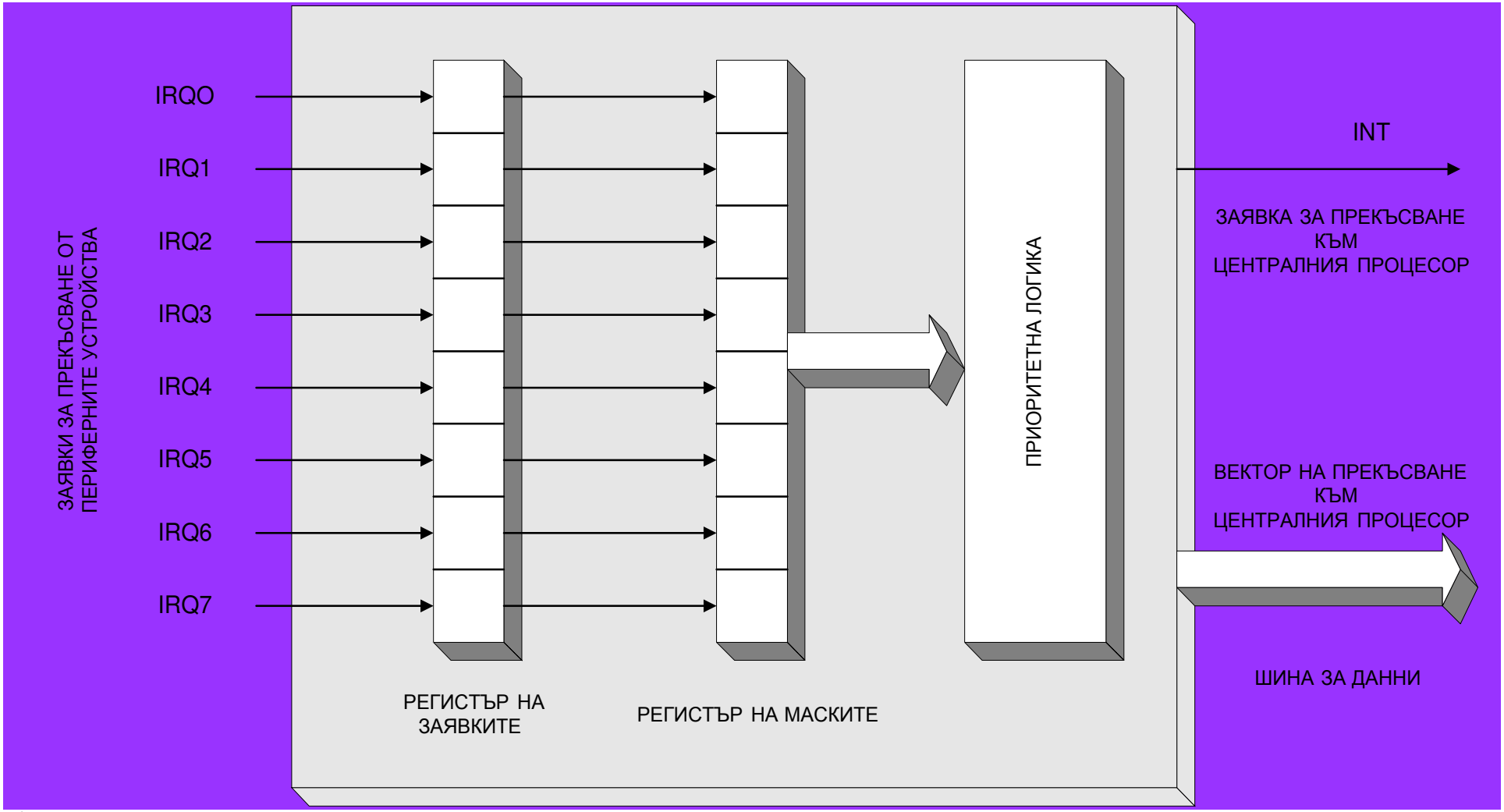
Хардуерен полинг (маргаритка)

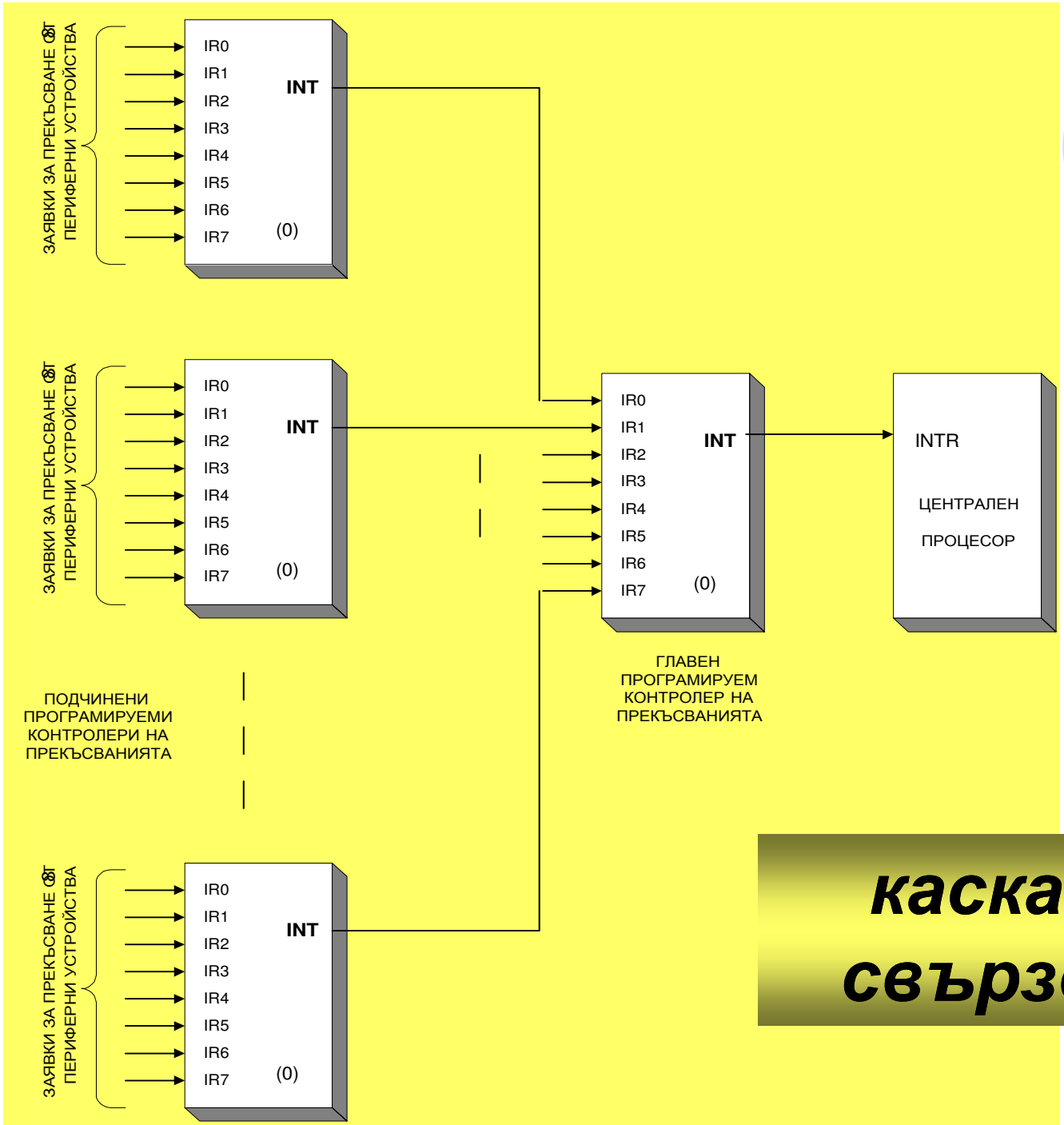


Възможните режими на обслужване на прекъсванията са:

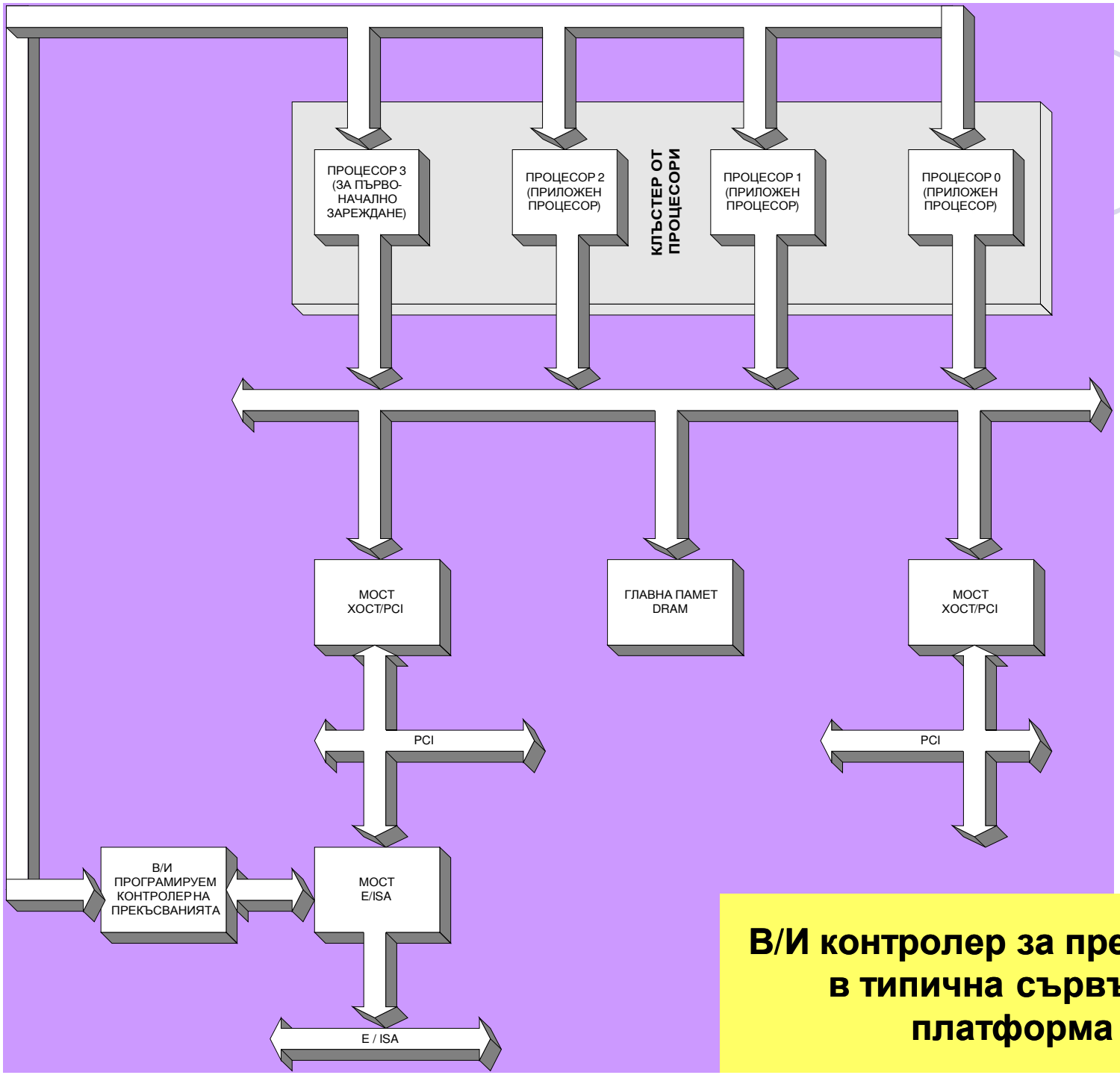
- *С фиксиран приоритет* - на всяко периферно устройства се задава приоритет от 0 (IR0) до 7 (IR7).
- *Ротация на приоритетите*
- *Маскиране* – процесорът забранява обслужването на заявки за прекъсване от определени от него периферни устройства

Вътрешна архитектура на програмируемия В/И контролер за прекъсване





каскадно свързване

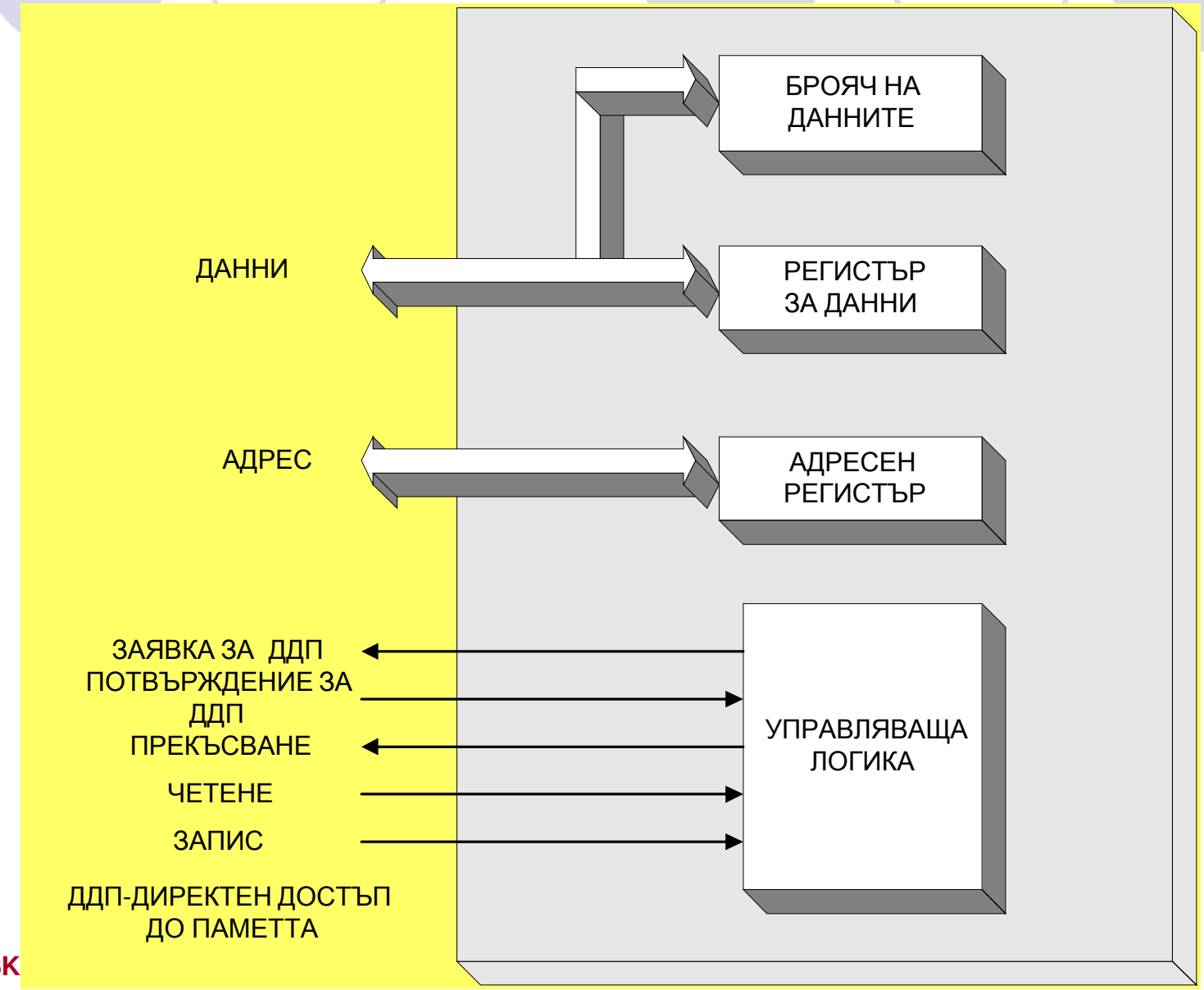


В/И контролер за прекъсване в типична сървърна платформа

директен достъп до паметта (*DMA* – Direct Memory Access)

- трансферът на данни между В/И модул и главната памет се осъществява директно, без намесата на централния процесор.
- Той е особено препоръчителен при трансфера на големи блокове данни.

контролер за директен достъп до паметта



В/И процесор

- има собствена система инструкции, специализирана за изпълнението на В/И операции.
- За иницииране на В/И обмен централният процесор дава директиви на В/И процесор да изпълни програма за В/И след което продължава с друга обработка.
- В/И процесор извлича инструкциите на В/И програма и ги изпълнява без участието на централния процесор.
- Следователно, централният процесор може да зададе последователност от В/И операции за изпълнение на В/И процесор, след изпълнението на които той ще бъде информиран чрез получаване на сигнал за прекъсване.

- В/И канал представлява разширение на концепцията за директен достъп до паметта.
- При компютърни системи с В/И канал централният процесор изобщо не е ангажиран с изпълнението на В/И, а само с неговото иницииране.
- Изпълняваната програма от В/И канал определя периферните устройства и областите от главната памет, между които ще се реализира обмена, приоритета на обслужване и спецификата на обработката на евентуално възникнали грешки.
- В/И канал следва тези инструкции и управлява В/И обмен.

В/И канали

- *Селекторните В/И канали* се използват за управление на трансфера с високоскоростни периферни устройства и в даден момент този тип канал управлява обмена само с едно устройство.
- *Мултиплексните В/И канали* се използват за управление на трансфера с периферни устройства с различно бързодействие и в даден момент този тип канал управлява обмена с множество устройства.
- За управление на бавните устройства се използват *байт-мултиплексни канали*,
- за управление на бързи периферни устройства се използват *блок-мултиплексни В/И канали*.

Интерфейсът между В/И модул и конкретното периферно устройство

- При паралелния интерфейс съществуват множество линии за данни, по които едновременно се предават битовете.
- Паралелният интерфейс се използва при по-високоскоростните периферни устройства като дисковете.
- При серийния интерфейс съществува само една линия за данни и по нея последователно във времето се подават битовете.
- Серийният интефейс се използва за по-бавни устройства като клавиатурата и принтерите.

Свързването между В/И модул и периферните устройства

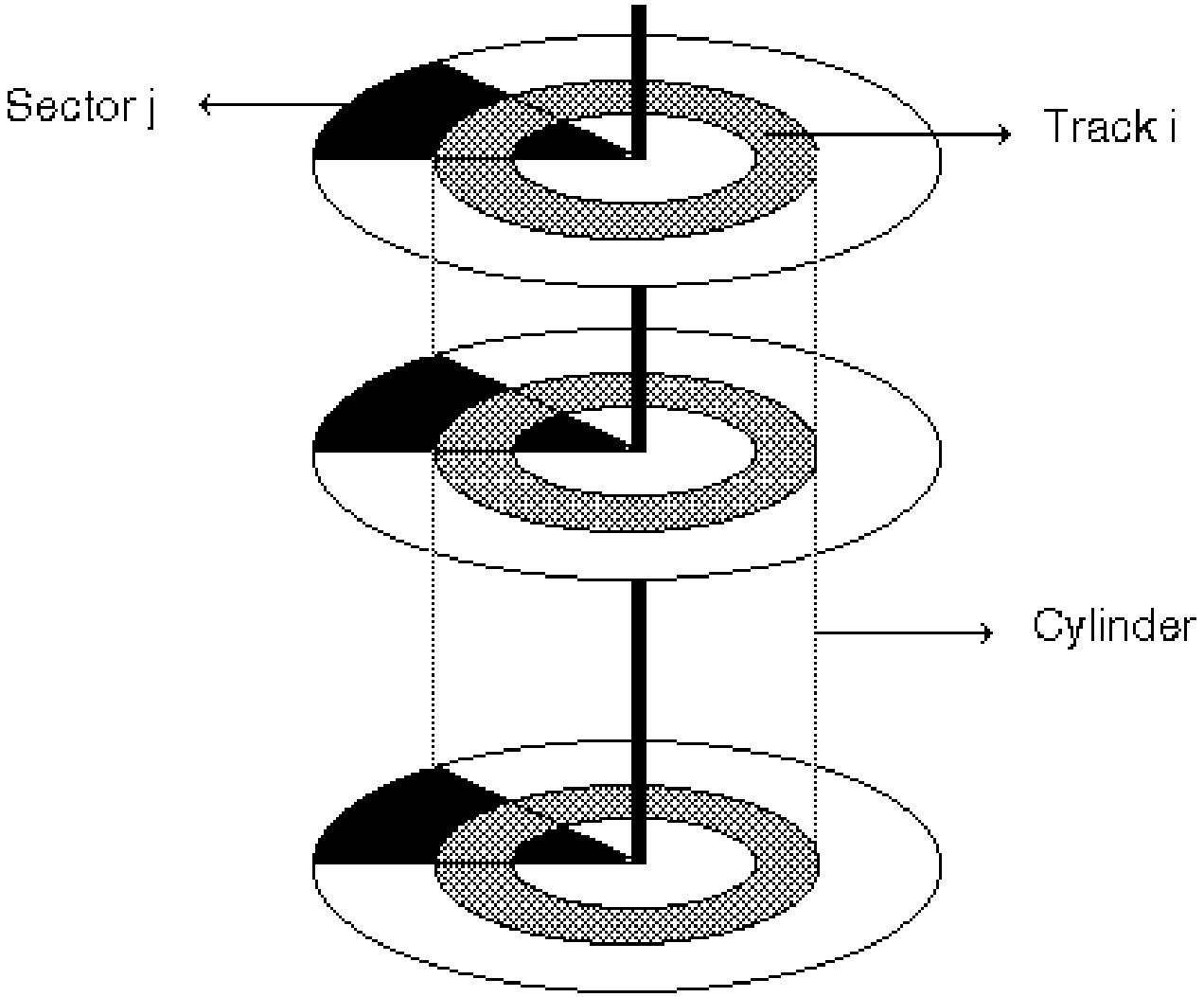
- може да бъде точка-с-точка (point-to-point) или с множество точки.
- Интерфейсът от тип ***точка-с-точка*** се използва за монополизирана линия между В/И модул и периферното устройство.
- При малките компютърни системи (персонални компютри и работни станции) този тип интерфейс се използва за клавиатурата, принтера и модема.
- Външните интерфейси с ***многоточково свързване*** се използват за външни масови памети (дискови устройства) и устройства за мултимедия (CD-ROM, видео, аудио) като на практика представляват външни шини.

ВЪНШНА ПАМЕТ

- При магнитния диск записът и четенето на информацията се осъществява от глава, която е стационарна, а дискът се върти под нея.
- Механизмът за запис се основава на факта, че електрическият ток, протичащ през главата, индуцира магнитно поле, което в зависимост от посоката на протичащия ток създава различни магнитни отпечатъци, кодиращи съответно двоичната 1 и 0.
- Механизмът за четене се основава на факта, че при ротацията на диска под главата, движещото се магнитно поле предизвиква протичането на електрически ток през главата с посока, при която е осъществен записът. Главата може да бъде фиксирана или подвижна.

- При дисковите системи с фиксирани глави за всяка пътечка е осигурена по една глава за четене – запис.
- Тези глави са монтирани на рамо, което покрива всички пътечки на диска.
- При дисковите системи с подвижна глава има само една глава за четене – запис, движеща се по направлението на радиуса на диска и която може да се позиционира над всяка пътечка.

Формат на диска



- ***Данните се организират в диска в концентрични окръжности, наречени пътечки***.
- В рамките на всяка пътечка се записват един и същ брой битове независимо от нейното разположение, така че гъстотата на битовете се увеличава в посока от външните към вътрешните пътечки.
- Пътечките са разделени на сектори.
- ***Секторът е най-малката информационна единица, до която може да се осъществи достъп в диска.***
- За всяка пътечка се записва допълнителна служебна информация (недостъпна за потребителя), идентифицираща нейното начало и край.

- Големите дискови системи осигуряват пакет от множество дискови повърхности.
- При тях цилиндърът включва пътечките с еднакъв радиус, принадлежащи на различните дискови повърхности.
- За да се осъществи достъп до даден сектор е необходимо да се зададе номера на дисковата повърхност, номера на пътечката и номера на сектора.

- дисковата система може да се рзглежда като тримерен масив от сектори.
- На практика тя се разглежда от операционната система като едномерен масив от блокове.
- Всеки блок представлява един сектор.
- Следователно, размерът на блока се определя от размера на сектора.

- В общия случай рамерът на физическия запис не съвпада точно с дължината на логическия запис.
- Освен това, дължините на логическите записи варират.
- *Блокът представлява базовата единица за разпределяне на дисковото пространство.*
- Това означава, че известно място в последния блок може да остане неизползвано (празно).
- Това явление се нарича ***вътрешна фрагментация***. Очевидно, колкото е по-голям размерът на блока, толкова е по-голяма вътрешната фрагментация.

- Файловете са основния механизъм за съхраняване на информацията в компютърната система.
- ***Файлът представлява абстрактен тип данни, дефиниран и имплементиран от операционната система.***
- Всеки диск има каталог на файловете, който се съхранява в нулевата му пътечка.
- Всяка позиция в този каталог съдържа символично име на файла, неговия тип, указател към местоположението на файла в диска, размера на файла, текущия указател на файла, защита, време.

Достъпът до файловете

- При файловете с *последователен достъп* се прилага моделът на магнитната лента. Най-често използваните операции при файловете са четене и запис.
- При операция четене се прочита следващата част от файла като автоматично се придвижва текущият указател на файла (*read next, write next*).
- При операция запис се прибавя новата информация към края на файла и автоматично се придвижва текущият указател на файла към новия край на файла.

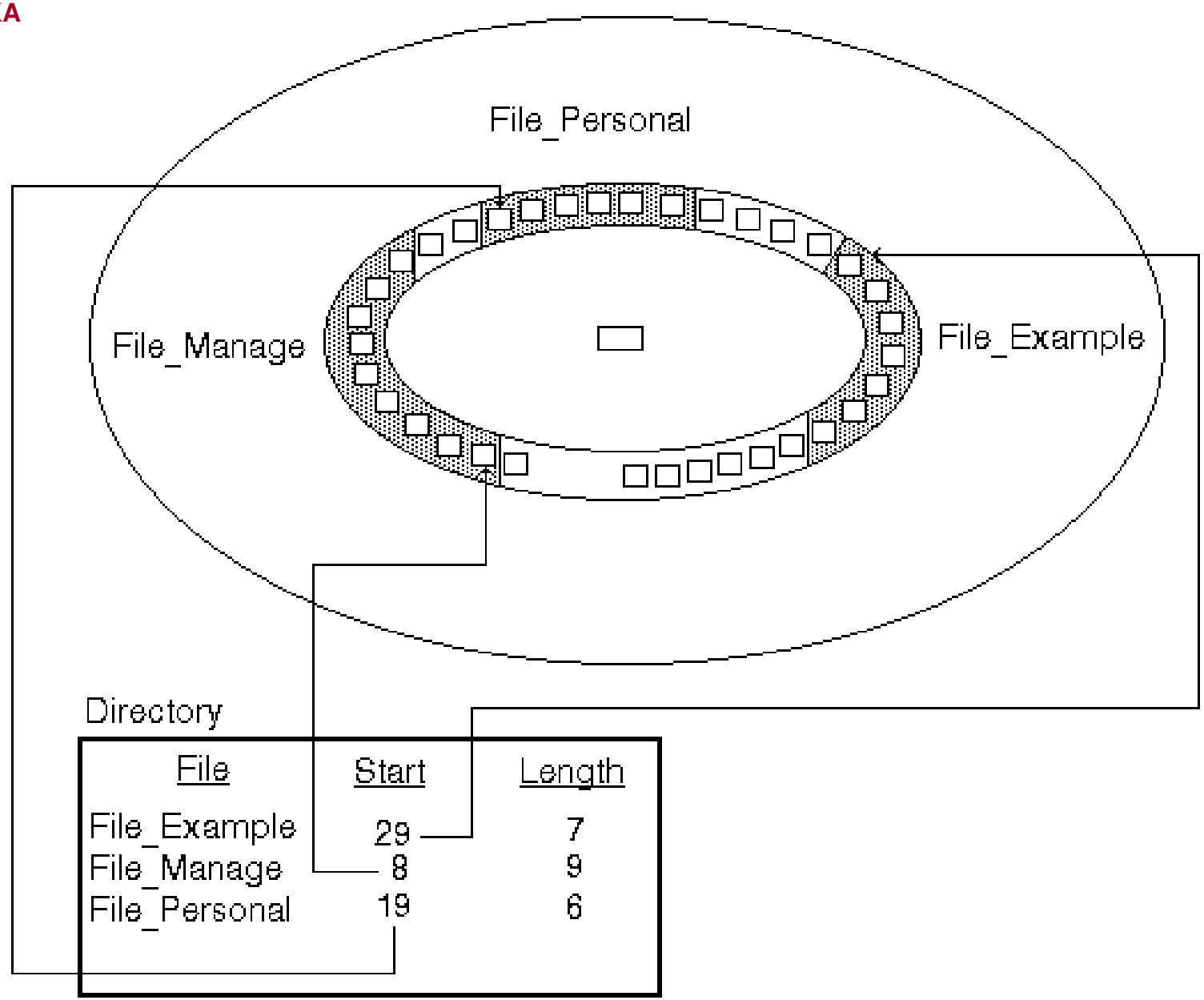
Достъпът до файловете

- При файловете с *директен достъп* се прилага моделът на диска.
- При него съществува възможност да се четат или записват произволни блокове. За тази цел операциите се модифицират по такъв начин, че да включват номера на блока като параметър (*read n, write n*).
- Операциите използват относителен номер на блока, представляващ индекс относно началото на файла. Очевидно не представлява трудност симулирането на последователен достъп при файловете с директен достъп.

Методи за разполагане на файловете върху диска

- При *метода с непрекъснато разполагане* всеки файл заема множество последователни адреси на диска.
- Местоположението на файла се определя от адреса на първия блок и дължината на файла.
- Позицията на файла в каталога на устройството съдържа тези два параметъра.

- Непрекъснатото разполагане на файловете върху диска поддържа както файлове с последователен достъп, така и файлове с директен достъп.
- Основният проблем е да се намери свободно място за разполагане на нови файлове, а също така за разширяващи се във времето файлове.

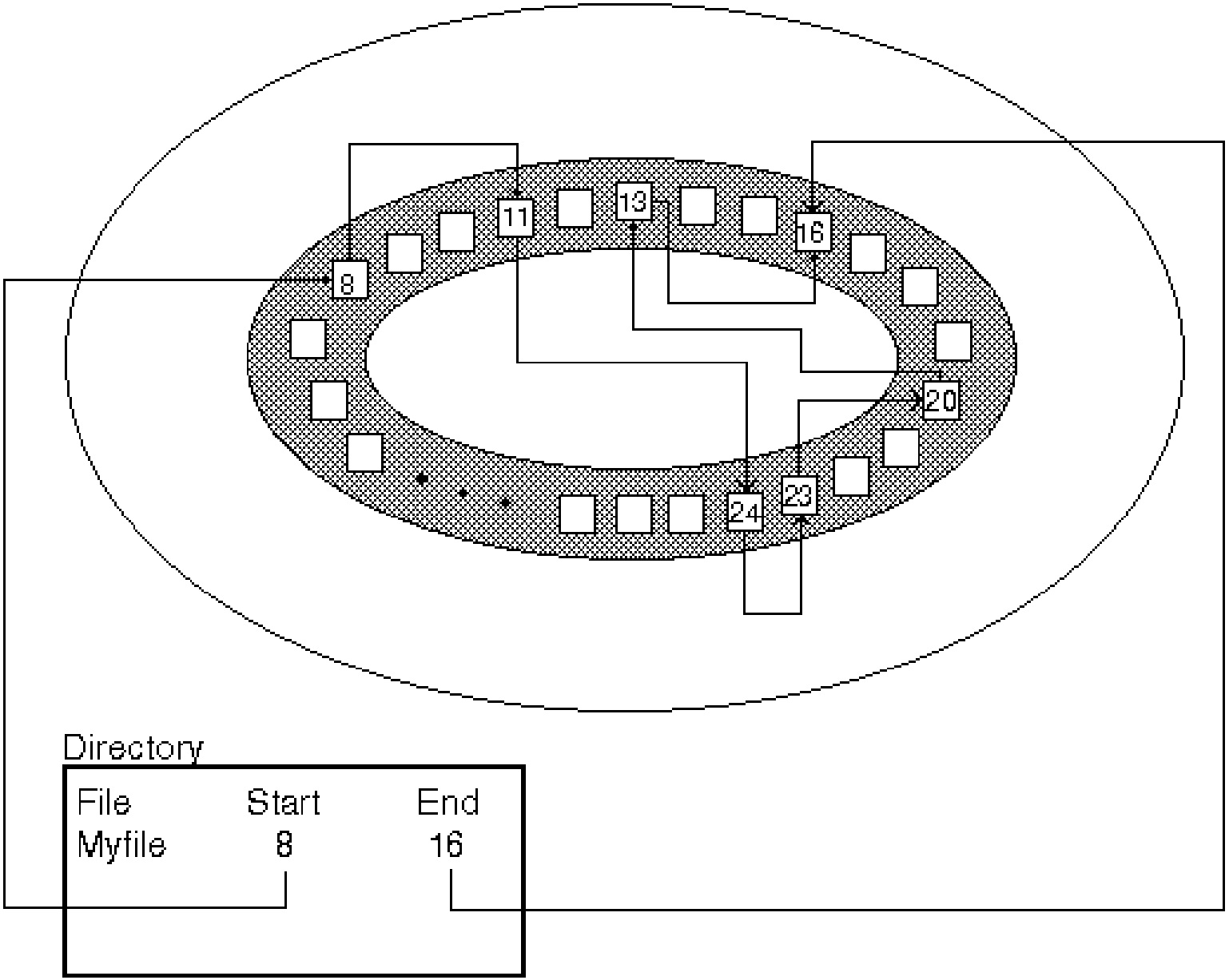


Непрекъснато разполагане на файл върху диска

списъчното разполагане на файлове

- всеки файл представлява свързан списък от блокове при което блоковете могат да бъдат разхвърляни на произволни места в диска. Каталогът съдържа указател към първия (и последния) блок на файла.
  - Всеки блок съдържа указател към следващия блок.
- При четенето на файла блоковете се четат последователно блок след блок следвайки указателите.

- не допуска външна фрагментация. Даден файл може да се разширява дотогава докато има свободни блокове в диска.
- Никога не се налага компактиране на дисковото пространство.
- Основният недостатък на списъчното разполагане е, че не се поддържат файлове с директен достъп.
- Друг недостатък е допълнителното пространство за съхраняване на указателите.
- Ако даден указател се загуби или повреди това може да доведе до свързване на празно пространство или друг файл.

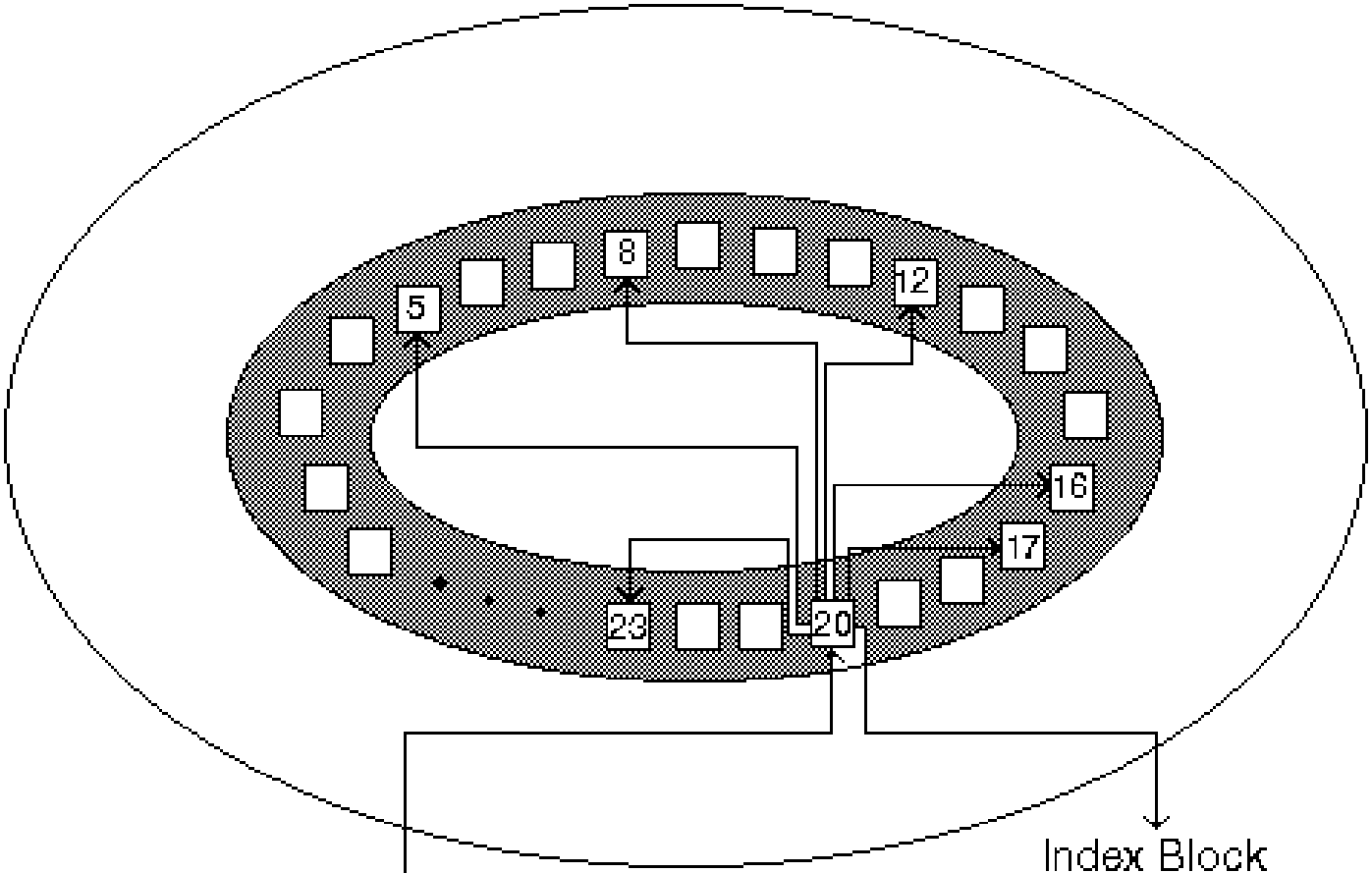


Списъчно разполагане на файловете върху диска

индексно разполагане на файла върху диска

- всички указатели се съхраняват в един блок, наречен индексен блок (блок с индексите).
  - Всеки файл разполага със свой собствен блок на индексите, който съдържа фактически масив от адресите на заеманите блокове в диска.
  - В таблицата на индексите i-тата позиция съдържа адреса на i-тия блок на файла.

- Метода с индексно разполагане на файловете не допуска външна фрагментация и поддържа файлове с директен достъп.
- Допълнителните разходи за поддържане на блока на индексите са по-големи от тези за поддържане на указателите при списъчното разполагане.
- Блокът на индексите обикновено заема един блок в диска.
- За поддържане на големи файлове, няколко файлове с индексно разполагане могат да бъдат свързани.
- В този случай за индексните блокове се използва списъчно разполагане. В повечето случаи за големите файлове се оказват достатъчни две нива на индексите.



Directory

File	Index Block
Myfile	20

Index Block

23
17
16
8
12
5

Индексно разполагане на файловете върху диска

Сравнителен анализ

- списъчното разполагане е ефективно при файлове с последователен достъп, но не може да се използва при файлове с директен достъп.
- Някои системи използват непрекъснато разполагане за да поддържат файловете с директен достъп и списъчно разполагане за файлове с последователен достъп.
- Добра алтернатива е да се използва непрекъснато разполагане за малки файлове и автоматично да се превключва към индексно разполагане ако файлът стане голям.

механизми за управление на свободното дисково пространство

- Поради ограничения капацитет на диска и непрекъснатото създаване на нови файлове и разширяване на съществуващите файлове, освободеното пространство при изтриване на файлове трябва да бъде използвано отново.
- Файловата система поддържа списък на всички свободни блокове на диска наречен списък на свободното пространство.

списък на дупките

- Дупката се дефинира като последователност от свободни блокове с произволна дължина.
- Всяка позиция в списъка на дупките съдържа адреса на първия сектор на дупката и нейната дължина.
- Очевидно, дължината на списъка на дупките и системните разходи за неговото поддържане зависи от разпределението на свободното пространство в диска, което се променя динамично във времето.
- Динамичното удължаване и скъсяване на този списък трудно се управлява от файловата система.
- Предимството на този подход е, че намирането на дупка със зададен размер е сравнително проста задача.

Списък на дупките

ДУПКА	НАЧАЛЕН АДРЕС		ДЪЛЖИНА
НОМЕР	ПЪТЕЧКА	СЕКТОР	(БРОЙ СЕКТОРИ)
1	2	4	3
2	4	3	6
3	5	1	8
4	6	8	3
5	8	5	9

битова карта

- Предимството на алтернативния подход с поддържане на битова карта е постоянния размер на битовата карта и простотата на нейната поддръжка.
- Промените в статуса на даден блок се отразяват чрез съответна промяна на състоянието на бита използван / свободен.
- Откриването на дупка със зададен размер е трудна задача.

Битова карта

TRACK	SECTOR NUMBER									
NUMBER	0	1	2	3	4	5	6	7	8	9
0	1	0	0	1	1	0	0	1	1	1
1	0	0	1	1	0	0	1	1	0	0
2	0	1	1	0	0	1	1	1	1	1
3	1	0	0	0	0	1	1	1	0	0

динамично преразпределяне на дисковото пространство

- Предназначението е да осигури механизъм за обслужване на заявка за намиране на свободно пространство с размер n от списъка на дупките.
- За да се открие дупка с оптимален размер е необходимо да се извърши търсене в списъка на дупките.

стратегии

- ***Първо попадение (First-fit)*** – избира се първата дупка, достатъчно голяма да удовлетвори изискванията на заявката.
- Търсенето в списъка на дупките може да започне от началото на списъка или от мястото, където е привършило предходното търсене.

Най-добро попадение (Best-fit)

- Претърсва се целия списък до откриване на най-малката дупка, която удовлетворява заявката.
- Ако списъкът се поддържа сортиран по големината на дупките не е необходимо да се търси в целия списък.
- При тази стратегия новосъздадените дупки са най-малки поради което възниква външна фрагментация.

Най-лошо попадение (Worst-fit)

- за обслужване на заявката се избира най-голямата дупка, при което новосъздадените дупки са най-големи.
- Ако списъкът не е сортиран се налага цялостното му претърсване.

- *Стратегията с първо попадение е най-бързата от трите разгледани стратегии*.
- По отношение на използването на дисковото пространство стратегиите с първо и най-добро попадение са по-ефективни от стратегията с най-лошо попадение.
- Недостатък и на трите стратегии е, че предизвикват ***външна фрагментация*** (празното пространство съдържа множество разхвърляни малки дупки в диска).
- В тези случаи дискът съдържа голямо свободно пространство, което е фрагментирано на множество и разпръснати малки дупки.

За да се преодолее негативния ефект на външната фрагментация е необходимо да се изпълни програма за компактиране, реализираща следното:

- Копира съдържанието на диска, който се компактира, върху друг диск;
- Оригиналният диск се изтрива;
- Файловете се копират обратно като се използват непрекъснати фрагменти.

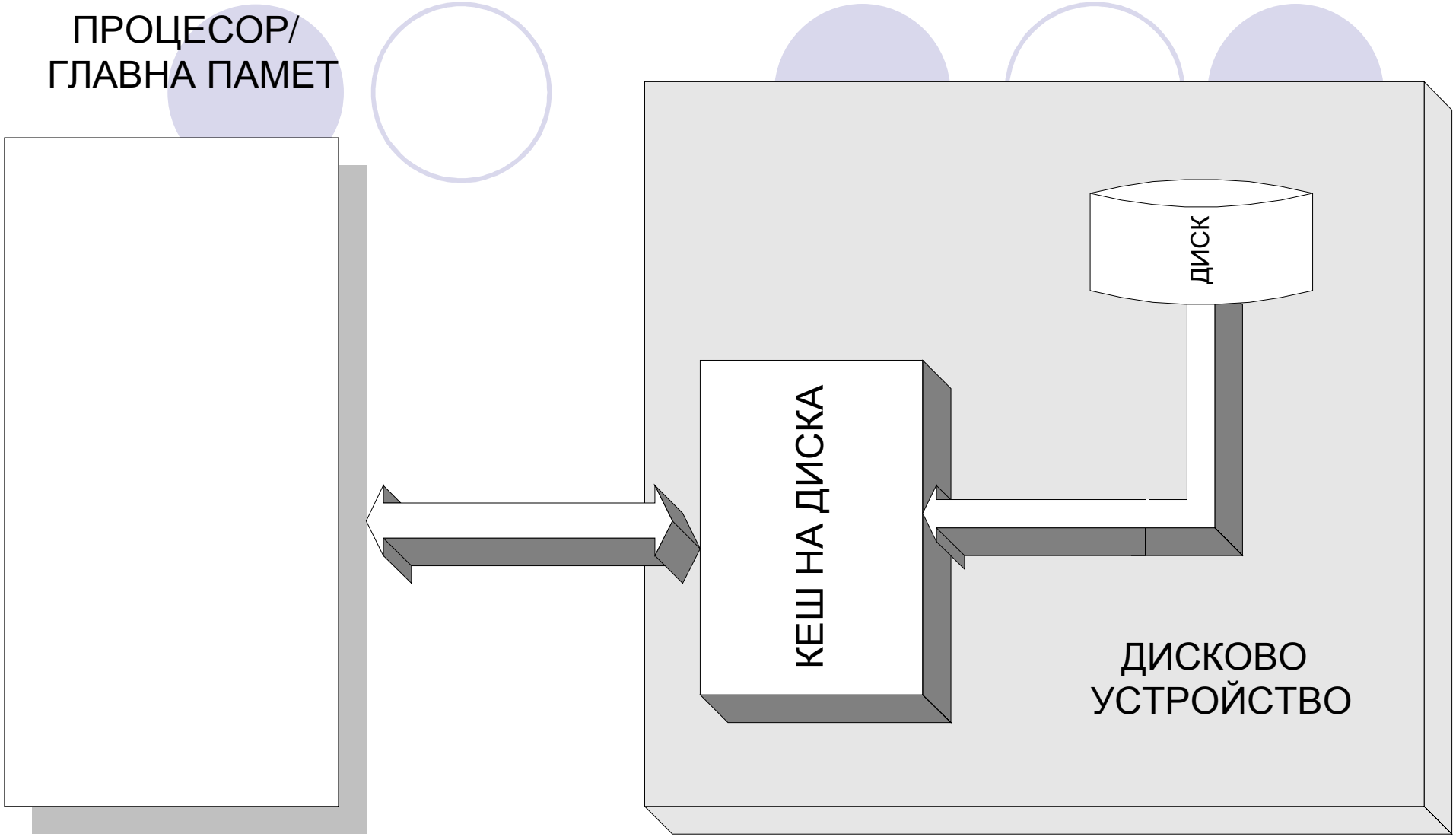
- Компактирането на диска разрешава проблема с външната фрагментация като обединява всички свободни фрагменти в една обща голяма дупка.
- Проблемът с външната фрагментация на диска и необходимостта от компактиране на диска е най-актуален при метода с непрекъснато разполагане.

- Бързодействието на магнитните дискове е значително по-малко (обикновено с няколко порядъка) от бързодействието на главната памет.
- Разликата се увеличава с напредъка на полупроводниковите технологии тъй като времето за достъп при диска и скоростта на трансфер се ограничава от механични фактори.
- Често процесорът не може да продължи изпълнението на текущата програма докато се реализира трансфер между диска и главната памет, дори когато този режим се осъществява от контролер за директен достъп до паметта без участието на процесора.

кеш на диска

- За намаляване на негативните ефекти от голямото закъснение на дисковата памет се използва концепцията за използване на *кеш на диска* в интерфейса главна памет / диск
- Кешът може да бъде включен в дисковото устройство или може да бъде организиран извън него в компютърната система.
- Капацитетът на дисковия кеш е много голям, имплементира се с RAM и съдържа блокове информация, които има вероятност да бъдат използвани в близкото бъдеще.

- Контролерът на кеша на диска активира информационния трансфер.
- Ефективността на дисковия кеш се повишава като се използва принципа на локалността, който редуцира ефективното време за В/И трансфер.
- Някои операционни системи като UNIX използват софтуерни техники за кеширане на диска при поддържането на В/И буфер в главната памет.
- Единицата за трансфер между диска и кеша може да бъде сектор, няколко сектора, пътечка или няколко пътечки.
  - Кешът на диска естествено има вложени механизми за откриване и коригиране на грешки.



Имплементиране на кеш на диска в рамките на дисковото устройство