

плоскост. Образът се проектира върху носител (4), фокусирайки се в точка от него. Към тази точка под определен ъгъл се насочва допълнителен лъч (5), наречен опорен, който е кохерентен на обектния лъч. Получената от пресичането на двата лъча интерферентна картина се регистрира като холограма. Размерът ѝ може да се регулира чрез диафрагма, а продължителността на съхраняване зависи от свойствата на регистриращата среда.

Прочитането на записаната холограма става по подобен начин, но в обратна посока. Холограмата върху носител (4) се осветява чрез насочване на падащ лъч (6), който е копие на опорния лъч при записа. В резултат от преминаване на четящия лъч през холограмата се получават няколко светлинни компоненти. Основната група компоненти се разпространява по посока на четящия лъч и формира т. нар. нулев лъч (5) - компоненти от нулев порядък. Другата компонента формира точно, но обърнато изображение на записаната в холограмата "картина" (при отсъствие на леща се създава мнимо изображение). Това изображение се използва за възстановяване (прочитане) на информацията чрез детектиране от матрица фотоприемници, поставена в позиция (2).

Организацията на холографска памет е по страници от данни и с адресен достъп чрез лазерен лъч. Лазерът генерира лъч от кохерентна светлина и се насочва чрез прецизен дефлектор към носител, преминавайки през система от холографски лещи и оптични призми. Целта е един и същ изходен лъч да се използва за формиране на двата лъча при записа – обектен и опорен, като при четене чрез поляризатор се блокира разпространението на обектния лъч. Въпреки големия обем съхранявана информация, масовото приложение на холографските паметни се ограничава от технологични трудности и сложност на реализацията им.

4.4. Понятие за виртуална памет

Виртуалната памет е такава организация на системната памет, която позволява независимост на потребителските приложения от физическата адресация на инструкции и данни в паметта на компютъра. За това е въведен механизъм на достъпа до информационните ресурси, осигуряващ бързото им доставяне, тяхната защита и ефективното им обменяне между отделните нива.

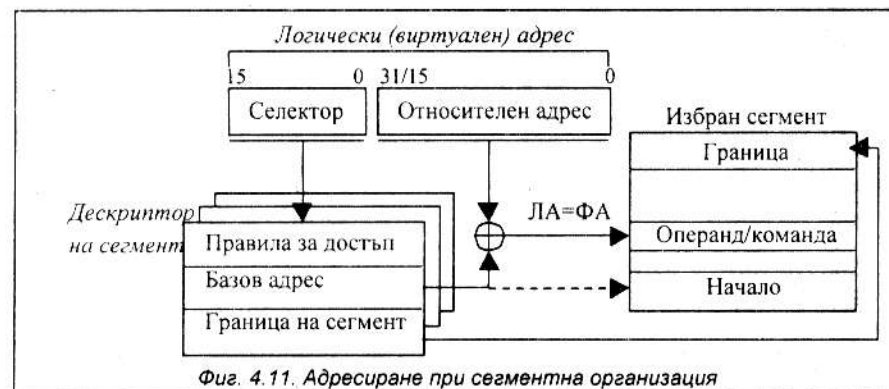
Известно е, че за да бъде изпълнена една програма, тя трябва да се намира в първичната памет. Тъй като обемът на ОП е ограничен, не винаги е възможно всички необходими информационни ресурси да са съхранени в нея. Обикновено там са актуални части от тях, а основната информация се съхранява във външна памет. Актуален пример е извеждане на справки от

база данни, която е съвкупност от файлове, съхранявани във външна дискова памет. Това налага непрекъснато обновяване на съдържанието на ОП чрез прехвърляне на блокове данни от вторичното към първичното ниво на системната памет, което се нарича *планиране на паметта*.

За организацията на виртуалната памет се използват допълнителни апаратни и програмни средства, позволяващи цялата памет да се приеме от потребителя като памет с едно ниво, независимо от йерархичната структура. Така той я възприема като собствена и с неограничено адресно пространство.

Това се постига чрез създаване на приложенията във виртуално адресно пространство без никакви ограничения, а при тяхното изпълнение механизмите и средствата на виртуалната памет *преобразуват виртуалните адреси (ВА) във физически адреси (ФА)*. За ефективна организация на това преобразуване и за автоматичния обмен на информацията между нивата на паметта се използват информационни блокове с еднакъв (страници) или различен (сегменти) размер.

Сегментната организация на паметта се реализира чрез разделяне на адресното пространство на отделни сегменти, характеризиращи се с определени атрибути (разположение, размер, тип, защита). В системната памет на процесорите от Р6 могат да се поддържат до 8192 сегмента с размер до 4 Гбайта всеки, което осигурява възможност за използване до 64 ТВ виртуална памет.



Фиг. 4.11. Адресиране при сегментна организация

За достъп до клетките на сегментната памет се използва съставен (логически) адрес, който се състои от селектор, задаващ базовия адрес (началото) на сегмента и относителен адрес на клетката (байт или дума) в сегмента (фиг.4.11). Така чрез аритметично сумиране на базов и относителен адрес се определя реалният ФА, освен ако не е използвана и допълнителна (странична) организация. При последното, формираният

адрес се третира като линеен адрес (ЛА) и се подлага на следваща обработка.

Параметрите на сегмента се представят в структура от 8 байта, наречена дескриптор. Дескрипторите на всички сегменти се съхраняват в паметта като масиви от данни, оформени в примерна таблица. Тази таблица може да бъде от 8 байта до 64 Кбайта (т.е. може да съдържа до 8192 дескриптора).

При изпълнение на приложения за 80386, 80486, Pentium и P6, баозвият и относителният адреси съдържат по 32 разряда. Границата на сегмента L указва максимално допустимата стойност на относителния адрес, която може да се използва при достъп до сегмента. Стойността (L+1) определя размера на сегмента в байтове или страници. Адресирането на клетка от паметта, която е извън границите на сегмента, се санкционира от защитата на виртуалната памет.

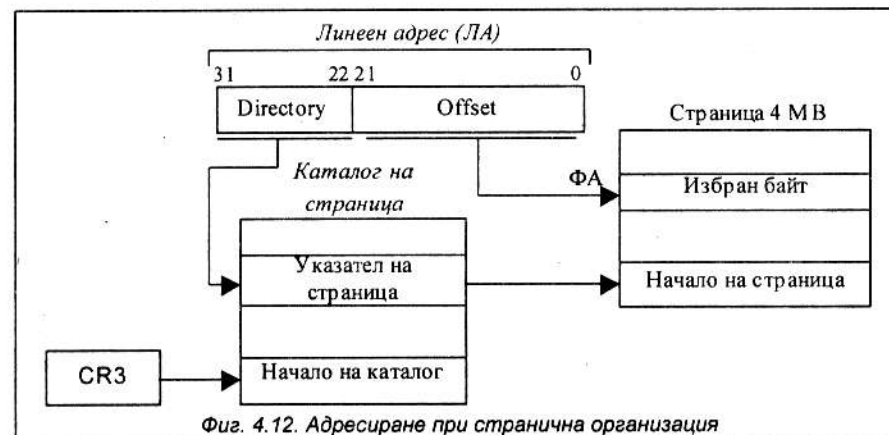
Страничната организация на паметта я разделя на равни по обем информационни блокове, които се характеризират със свой начален адрес на страницата. Обикновено тези адреси се съхраняват в памет на страниците, като достъпът до тях се извършва чрез старшата част от ВА и след това чрез присъединено адресиране се определя ФА на клетка от страницата.

При процесорите от P6 странична организация е предвидена за самите сегменти. Размерът на страниците може да бъде 4 KB, 2 MB или 4 MB и за разлика от размера на сегментите (от 1 B до 4 GB) е строго фиксиран. Страниците могат да се разполагат на различни места в реалното адресно пространство и да се групират в раздели. Ако размерът на страницата е 4 KB, то в един сегмент могат да се формират до $2^{10}=1024$ раздела, като във всеки раздел може да има до $2^{10}=1024$ страници от по 4 KB. Ако размерът на страниците е съответно 4 MB или 2 MB, то сегментът се разбива само на отделни страници, съответно 1024 или 2048 на брой.

Разполагането на разделите и страниците в ОП се извършва от диспечер на паметта, включен в състава на ОС. Диспечерът задава базовите (началните) адреси на разделите и страниците, които се съхраняват в паметта в специални адресни таблици (каталози). Освен базов адрес в таблиците се съхраняват и правила за достъп и друга служебна информация.

На фиг.4.12 е показан начинът на формиране на ФА при странична организация с размер на страницата 4 MB (разширен размер). Линеиният адрес (ЛА) е формиран на базата на ВА при адресиране на сегмента. Чрез него се формира ФА, като процедурата зависи от размера на използваната страница и разрядността на адреса. В показания случай, ЛА се състои от две полета: Directory – за адресиране на адресната таблица (каталога);

Offset – съдържа отместването (младшата част) за адресиране на байт в избрана страница. В разглеждания случай се поддържа само един каталог, понеже няма формирани раздели. При страници от по 4 KB са необходими няколко такива каталози, за адресирането на които се въвежда трето поле Table между полетата Directory и Offset.



За адресиране на каталога се използва служебен регистър за управление CR3, в който е зареден базов адрес (старшата част от 20 бита) на каталога. Младшата част се допълва от съдържанието на поле Directory от ЛА. Чрез формирания адрес от таблицата се извлича указателят на търсената страница, който представлява нейния базов адрес (това е старшата част ФА[31+22] на 32-битовия реален ФА). Останалата младша част ФА[21+0] се определя от съдържанието на поле Offset в ЛА.

При страничната организация ВА се преобразува в ЛА, от който чрез обръщане към адресните таблици на страниците се определя ФА. За да се увеличи бързодействието при достъп до страниците, базовите адреси на последните адресирани 32 страници се съхраняват в т.нар. вътрешни буфери за трансляция на адреса, от където директно могат да бъдат използвани.

Освен преобразуването на ВА във ФА, при виртуалната памет трябва да се решат още две важни задачи.

Първата от тях е свързана с *планиране на виртуалната памет*, т.е. избор и реализация на стратегия за планиране на присъствието на определени информационни блокове в първичната памет и периодичното ѝ обновяване с такива от вторичната памет, в зависимост от актуалната компютърна обработка. Решаваният проблем е по-скоро кой от блоковете в ОП да бъде временно прехвърлен във вторичната памет, за да се освободи място за необходимия в момента въвеждан блок. Традиционните стратегии

за избор са свързани с най-голямо време за престой в ОП, с максимално време за неизползване на блок, оценка на вероятности и пр.

Другата задача е *защита на виртуалната памет*, т.е. защита на информационните масиви от чужда намеса (при грешка или при неправомерен достъп). За целта се въвеждат механизми и средства за ограничаване на възможностите за адресиране извън предоставеното адресно пространство – с гранични адреси, с регистър на достъпните страници, чрез защитни ключове за потребителските програми и пр.

В процесорите от Р6 това се поддържа от съдържанието на дескриптора на сегмента, който освен правилата за достъп, определя и размера L на сегмента. Системата за защита при тези процесори предвижда:

- контрол на достъпа до сегменти и страници, базиран на система от привилегии;

- контрол на използването на сегменти и страници с въвеждане на различни ограничения: забрана за запис в сегменти (страници), които са достъпни само за четене; забрана за прочитане на програмни сегменти (страници), които са предназначени за изпълнение; забрана на достъпа до незаредени в ОП сегменти (страници) и пр.;

- ограничаване на множеството от изпълнявани инструкции (команди) в зависимост от нивото на привилегии за изпълняваната програма.

В МПФ на Intel са установени четири нива на привилегии (PL – Privilege Level):

PL0 (най-привилегировано ниво) – ядро на ОС;

PL1 – утилити на ОС;

PL2 – служебни програми (драйвери, системи за управление на бази от данни, програмни системи и пр.);

PL3 – потребителски приложения.

В зависимост от предоставените права и защитеността са установени следните правила:

1. Данните от сегмент с ниво PL_j могат да бъдат четени от програми, имащи същото или по-високо ниво на привилегии.

2. Програмен сегмент (процедура) с ниво на защита PL_j може да бъде извикан от програма, имаща същото или по-ниско ниво на привилегии.

За поддържане на описаната система за програмите с различни нива на привилегии са организирани отделни стекове. Така в системата функционират четири различни стека, всеки обслужващ съответното ниво.

Глава пета

Класическа компютърна архитектура

Класическият архитектурен модел на фон Нойман е базовата концепция за изграждане на универсални и специализирани компютри с традиционна структурна организация. Включва относително самостоятелни подсистеми (процесорна, входно-изходна, програмна) и се основава на следните принципи:

а) адресируема системна памет;

б) фиксирани формати за данни и инструкции;

в) последователно изпълнение на инструкции от едно приложение, като текущата инструкция определя еднозначно прехода към следващата;

г) разделяне на управлението от изпълнението на операциите, както и от другите функции на обработката;

д) изпълнение на операциите в АЛУ, като операндите се съхраняват временно в регистри;

е) управлението е на базата на съхранявана в паметта програма и анализ на признаците на състоянието на изчислителния процес.

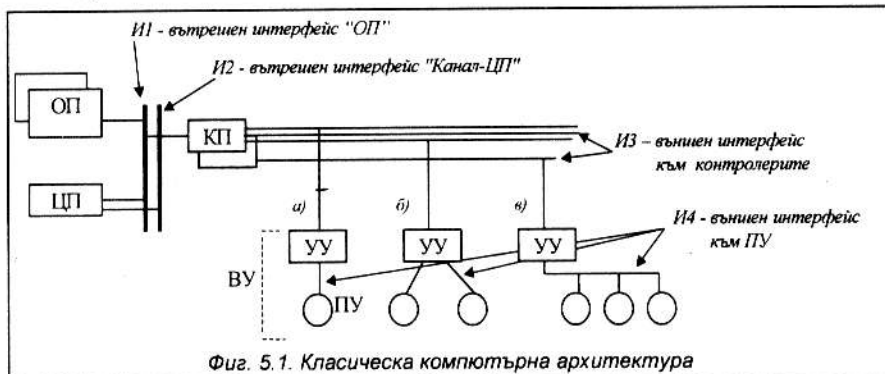
Моделът на фон Нойман е универсална машина и позволява развитие в архитектурната концепция, базирана на дадена операционна система (ОС). *Усъвършенстването* на архитектурния модел на фон Нойман е свързано с придаване на по-голямо значение на програмното управление (операционната система) и въвеждане на елементи на *псевдопаралелизъм* в работата - канална организация на КС с развита система за поддържане на прекъсванията. Това определя една КС като съвкупност от апаратни и програмни средства за обща организация на изчислителните процеси при оптимално разпределение на ресурса между активните в текущия момент задачи. Такава система се характеризира с модулна организация, която допуска разширяемост на структурата (фиг. 5.1). Изпълнението на тази цел зависи от: операционните и програмни среди за компютърни приложения; режимът на работа на КС, поддържан от ОС; стратегията за планиране изпълнението на задачите в КС. В структурно отношение класическата архитектура предвижда самостоятелност на работата на ядрото (ЦП, ОП) от входно-изходната (В/И) система, но под общото управление на функционалния алгоритъм (централното програмно управление). Последното се осигурява от елементите на системното ПО и в частност от ОС. В състава на В/И система влизат каналния процесор (КП), външните

устройства (ВУ) и интерфейсите средства за връзка между тях. Повишаването на бързодействието се постига със следните прийоми:

а) съвместяване на работата на ЦП (обработващата система) с тази на периферията чрез въвеждане на каналната организация на В/И система;

б) разделяне на фазите "извличане на инструкцията" и "изпълнение на операция", позволяващо конвейеризация на ниско ниво при обработката на инструкциите;

в) развитие на машинния език към окрупняване на операциите, изпълнявани от една машинна инструкция - включване в машинния език на векторни и матрични операции или усложняване на езиковите му конструкции за доближаване до езиците от високо ниво.



Фиг. 5.1. Класическа компютърна архитектура

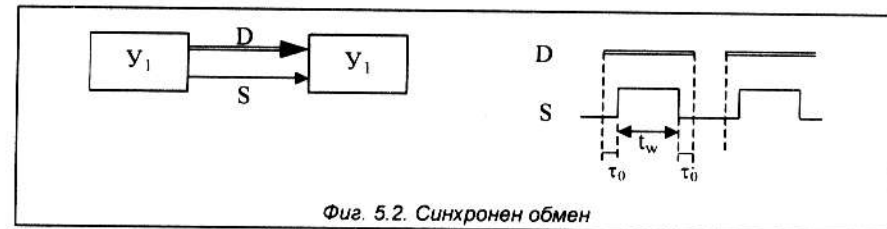
5.1. Организация на входно-изходна система

5.1.1. Информационен обмен и интерфейси

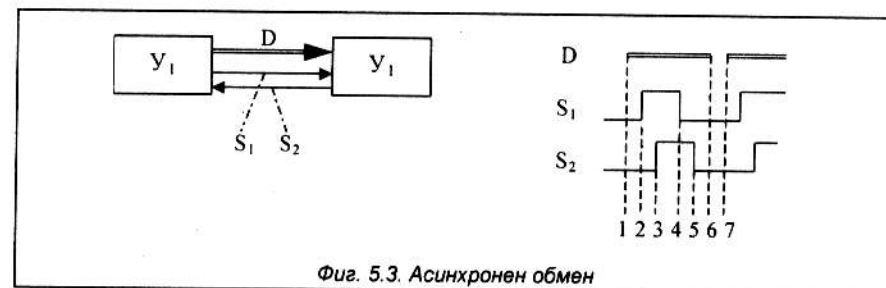
Информационният обмен между две устройства в КС се извършва в паралелен (обикновено 8 бита) или последователен (1 бит) код по линии (шини) за връзка. Основните принципи при предаване са:

а) **Синхронен обмен** (фиг.5.2) – данните (D) по информационните линии между устройствата Y_1 и Y_2 се съпътстват с допълнителен синхронизиращ сигнал (S), определящ начало и край на валидност за данните. Основното време за предаване/приемане на данните е продължителността на тактовия импулс t_w , което е достатъчно за изпълнение на обмена. За по-голяма сигурност е предвиден запас от време τ_0 , през което данните са в наличност върху шината.

б) **Асинхронен обмен** (фиг.5.3) - допълнителните управляващи сигнали са два (S_1 , S_2), чрез които се организира запитване ("разговор") за готовност на всяко устройство за провеждане на обмена.



Фиг. 5.2. Синхронен обмен



Фиг. 5.3. Асинхронен обмен

Прочитането на този обмен може да се опише чрез следните стъпки:

- (1) Y_1 подава готовите за изпращане данни на информационната магистрала (шина за данни);
- (2) Y_1 изпраща управляващ сигнал S_1 за изпратени данни;
- (3) Y_2 връща отговор S_2 за приет сигнал за начало на обмена и прочита данните от шината;
- (4) след получаване на S_2 устройство Y_1 нулира своя сигнал S_1 ;
- (5) промяната на S_1 води до нулиране и на S_2 от устройство Y_2 ;
- (6) приключва обменът на тази "порция" данни и след време Δt се подготвят данни за следващ обмен – (7).

Интерфейсът осъществява връзката между устройствата в КС и представлява съвкупност от шини и управляващи сигнали, осигуряващи единен принцип на обмен на информацията. Могат да се дефинират следните типове (виж фиг. 5.1):

а) **вътрешни**: И1 (интерфейс на оперативната памет) - осигурява обмен на адреси и данни към и от ОП в зависимост от избрания формат на обменяната дума или информационен блок; И2 (интерфейс "канал-процесор") - за обмен на адреси и инструкции между централния процесор (ЦП) и каналите (каналния процесор) за стартиране на В/И операция;

б) **външни**: И3 (интерфейс към контролерите) - за връзка между съответния канален процесор и адресираното ВУ и осъществяване на обмен на адреси, команди и данни (обикновено по 1 байт); И4 (интерфейс към ПУ) - за връзка между управлението (контролера - УУ) и самото периферно устройство (ПУ).