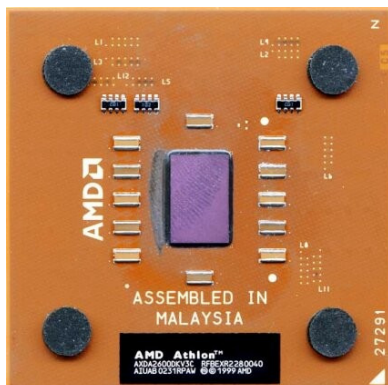


# ПРОЦЕСОР.ФУНКЦИОНАЛНА БЛОК СХЕМА.ПРОИЗВОДИТЕЛНОСТ.ОРГАНИЗАЦИЯ.

## 1.СЪЩНОСТ НА ПРОЦЕСОРА



### ФУНКЦИОНАЛНА БЛОК СХЕМА

Централният процесор е програмируемо устройство, което се явява основен движещ инструмент в компютърната система. Изпълнявайки стъпка по стъпка инструкциите записани в конкретната програма, той решава задачите, поставени пред компютъра. Определя неговата производителност и ефективност.

#### Задачи на процесора

Изпълнява всички необходими математически, логически и мултимедийни операции;

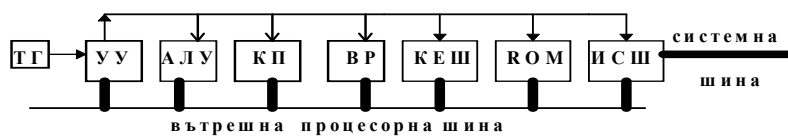
Осъществява управление на устройствата на компютъра;

Контролира изправността на устройствата на компютъра;

Обработва постъпващите от различни устройства данни и формира резултатите от тази обработка.

#### Компоненти на процесора

В съвременните персонални компютри централният процесор конструктивно е изпълнен като микропроцесор на базата на свръх голяма интегрална схема. Това е полупроводников кристал, в който се реализират компонентите на процесора.



#### Тактов генератор (ТГ)

Генерира електрически импулси през определен интервал от време, които осигуряват тактовата честота на процесора.

#### Тактова честота

Определя скоростта на извършваните от микропроцесора операции и се измерва в Hz. Оказва влияние върху бързината на компютъра, но не е определяща. Оказва по - голямо влияние при програмни приложения, извършващи операции в паметта, например преизчисляване на електронни таблици. При приложения свързани с дискови операции влиянието е незначително, защото дисковите устройства са с голямо време на достъп. Тук определящ критерий е употребата на цикли за изчакване.

## Цикъл на изчакване

Времето, през което процесора изчаква по-бавното устройство, за да се синхронизира с него.

### Управляващо устройство (УУ)

Управлява работата на всички основни възли на процесора, като им изпраща съответни управляващи сигнали.

### Аритметико-логическо устройство (АЛУ)

Извършва аритметичните и логически операции, зададени със съответните инструкции на изпълнимата програма.

### Копроцесор

Предназначени са за изпълнение на математически операции. До 486DX копроцесорите се добавяха като опция, а след него те се вграждат в процесора. Копроцесорите притежават мощна аритметика с плаваща запетая и набор от специални константи и трансцендентни функции, които са полезни при математически и инженерни изчисления,

### Постоянна памет (ROM)

Постоянна памет, в която са записани програмите за изпълнение на процесорните инструкции (набор от микроинструкции).

### Вътрешни регистри (BP)

Изпълняват важна роля при съхранение моментното състояние на компютъра и изпълнимата програма, при обработка на прекъсванията и при управление процеса на адресация.

Тип на регистрите	Брой/бита	Предназначение
Общи	8/32	Временна свръхоперативна памет за съхранение на 8, 16 и 32 битови думи.
Сегментни	6/16	Подпомагат процесора при адресиране на секции или сегмент (1B до 4GB) от паметта
Регистри за състояние и управление	2/32	EFLAGS-регистър на флаговете, указващи на програмите състоянието на компютъра - резултати от аритметични операции, прекъсвания и др. подобна информация. EIP - указател на инструкциите, запомня мястото до което е стигнал процесора при изпълнение на програмата.

### Стек

Временна памет, където процесорът съхранява цялата текуща информация за това, което компютърът върши в момента. Когато трябва да изпълни подпрограма или някаква друга задача, процесорът запомня в стека временното състояние и какво е правил до този момент, така че по-късно да може да се върне към изпълняваната програма без затруднения. При всяка нова задача компютърът помества на върха на стека информация за задачата и какво е текущото и състояние. При приключване на задачата информацията на върха на стека се заменя с предишната.

### КЕШ (L1)

Малка буферна памет, в която се съхраняват често използваните от процесора инструкции и операнди.

### Интерфейс със системната шина (ИСШ)

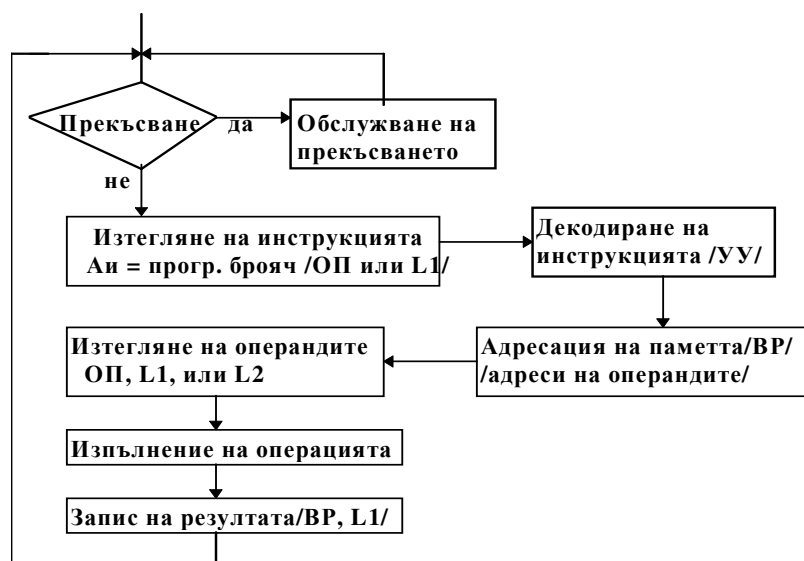
Осъществява връзката на процесора с останалите компоненти на компютърната система.

### Вътрешна процесорна шина

Вътрешна магистрала за обмен на данни между отделните компоненти на процесора. Работи с тактовата честота на процесора

## Работен цикъл на процесора

### Алгоритъм на обработка на инструкция



## 2.ПРОИЗВОДИТЕЛНОСТ НА ПРОЦЕСОРА

### Тенденции, влияещи върху производителността

- До 1990г. - компютрите в масовите си приложения се занимаваха с аритметическа и логическа обработка на данни. Основен критерий за оценка на производителността бе тактовата честота. Колкото е по-висока тактова честота, толкова е по-голям броя изпълнени операции за секунда и съответно по-висока производителността на процесора.
- След 1990г. се появиха нови приложения, като мултимедията, многопроцесорни системи, многозадачни операционни системи и др., които изискват нов подход и критерии при оценка производителността на процесора.

### Поддръжка на операционната система

Операционната система очевидно е неразделна част от компютъра. В някои приложения взаимовръзката операционна система - процесор може да се окаже доста критична за избягване времето на бездействие между отделните превключвания.

### Поддръжка на мултимедия

Развитието на мултимедийни приложения - видео конференции, дву и тримерни изображения, анимация и други, изискват качествено нови процесори. Специални инструкции за изпълнение на сложни графични операции значително увеличават производителността на процесора. Видео приложенията с високо качество натоварват процесора да работи с големи количества информация и да изпълнява много операции по едно и също време(плавна анимация и интерактивно видео в реално време).

### Многопроцесорна поддръжка

Многопроцесорната поддръжка стана много важна през това десетилетие. Многопоточните приложения са вече реалност и масово навлизат в различни области.

### Фактори, оказващи влияние върху производителността:

#### Разрядност на процесора

Колкото по-вече бита данни може да обработи процесорът едновременно, толкова е по-голямо бързодействието му.

#### Тактова честота

Колкото е по висока тактовата честота на процесора, при равни други условия, толкова е по-висока производителността му. Основно тактовата честота зависи от дебелината на силициевата подложка на процесорния чип.  
( 0.25 микрона - над 300 MHz; 0.18 микрона - над 700 MHz; 0.13 микрона - над 1000 MHz;).

### **Архитектура**

Преминването от стандартна към суперскаларна и многопроцесорна архитектури значително увеличи производителността на процесорите. В съвременните процесорни архитектури се използват техники като паралелната и конвейерна обработка, предсказване на преходи, двойна кеш памет и др., които пълно натоварват процесорното време и значително ускоряват скоростта на обработка на данните.

### **Оценка на производителността**

Оценката производителността на различните процесори и компютри е сложен процес и зависи от много фактори - производители, архитектура, компоненти и т.н. Съществуват различни методи за оценка, като те се променят с развитието на компютрите.

#### **Чрез сравнение на някои основни параметри:**

Когато се сравняват два компютъра е необходимо да се включат и трите параметъра, за да се оцени относителната производителност.

□ Такт (или честота) на синхронизация - при съвременните процесори скоростта на обработката на данни в резултат на взаимодействието на различните компоненти на компютъра се определя не от времето за изчакване между компонентите, а се задава чрез единна система от синхросигнали, изработвани от генератора на тактови импулси

Дискретните временни събития се наричат тактове за синхронизация (clock ticks), само такт (ticks), периоди на синхронизация (clock periods), цикли (cycles) или цикли за синхронизация (clock cycles). Обикновено се говори за период на синхронизация, който се определя или от своята продължителност (например, 10 наносекунди), или честота (например, 100 МГц).

Честотата на синхронизация се определя от технологията на апаратните средства и функционалната организация на процесора;

□ Среден брой тактове на команда CPI (clock cycles per instruction) - важна характеристика, често публикувана в отчетите за процесори.

При известно количество изпълнявани команди в програмата, този параметър позволява бързо да се оцени време ЦП за дадената програма.

Средния брой тактове на команда зависи от функционалната организация на процесора и архитектурата на системните команди.

□ Брой изпълнени команди в програмата - определя се от архитектурата на командите и технологията на компилаторите.

**Чрез измерване на времето за изпълнение** на операция, програма или нещо друго. На практика единствената подходяща и надеждна единица за измерване на производителността се явява времето за изпълнение на реални програми.

### **Време**

Единица за измерване на производителността на компютъра.

Времето за изпълнение на всяка програма се измерва в секунди. Често производителността се измерва като скорост на появяване на брой събития за секунда. Времето може да има различни значения, в зависимост от конкретната ситуация - астрономическо време, време за отговор (response time), време за изпълнение (execution time) или изминало време (elapsed time).

### **Време ЦП (CPU time)**

Специален параметър за измерване времето на работа на процесора с дадена програма. В това време не се включва времето за очакване на вход/изходна операция или времето за изпълнение на друга програма.

**Време ЦП = Потребителско време ЦП + системно време ЦП**

**Потребителско време ЦП**

Времето изразходвано непосредствено за изпълнение на потребителска програма.

**Системно време ЦП**

Времето изразходвано от операционната система за изпълнение на потребителската програма.

**Стандартни единици за оценка**

Съществуват няколко стандартни единици за измерване производителността.

**MIPS (милион команди в секунда)**

Една от алтернативните единици за измерване производителността на процесора (по отношение към времето за изпълнение).

Съществуват няколко различни варианти за интерпретация на определението MIPS:

MIPS = количество на команди за единица време.

MIPS = количество команди в програмата/времето за изпълнението и.

Положителни страни:

Характеристиката е лесна за разбиране;

По бърз компютър се характеризира с по-голямо число MIPS.

Недостатъци:

MIPS зависи от набора команди на процесора, което затруднява сравнение по MIPS на компютри, имащи различни системни команди;

MIPS дори на един и същи има различни стойности за различни програми.

MIPS не отчита възможностите за използване на помощни процесори - копроцесор, графичен процесор и други процесори.

**MFLOPS**

Милион елементарни аритметически операции над числа с плаваща запетая, изпълнявани за секунда. Използва се за измерване производителността на компютрите при решаване на задачи, в които активно се използва аритметика с плаваща точка.

MFLOPS се базира на количеството изпълнявани операции, а не на количеството изпълнявани команди. Положителното в сравнение с MIPS е, че една и съща програма, работеща на различни компютри, ще изпълнява различно количество команди, но едно и също количество операции с плаваща точка.

### **3.КЛАСИФИКАЦИЯ НА ПРОЦЕСОРИТЕ**

**Според използваните команди**

Различните процесори, използват различен набор инструкции.

В съвременните компютри се използват няколко основни архитектури за набори команди.

**CISC Complex Instruction Set Command** (сложен набор команди)

Те имат класическа архитектура, характеризираща се с голям набор команди. Благодарение на това, процесорите изпълняват разнообразни задачи по обработката на данни.

Основни характеристики:

- ❑ Сравнително малък брой регистри с общо предназначение;
- ❑ Голямо количество машинни команди, изпълнявани за много тактове;
- ❑ Голямо количество методи за адресация;
- ❑ Голямо количество формати за команди с различна дължина, като преобладава двуадресен формат на командите
- ❑ Наличие на команди за обработка от типа регистър-памет.

**RISC Reduced Instruction Set Command** (съкратен набор бързо изпълнявани команди).

Опростена архитектура с оптимизиран набор от машинни инструкции, което означава че наборът включва само онези елементарни операции, изпълними от хардуера с максимална бързина и ефективност. Всяка друга по - сложна операция се изпълнява като комбинация на няколко прости инструкции.

Първоначално създадени като научен продукт, днес RISC се използват от IBM, HP, SUN и др. производители в производството на работни станции и суперкомпютри.

Основни характеристики:

- Имат не по-вече от 128 бързо изпълняващи се команди, извършващи елементарни операции. Когато е необходимо да се извърши по сложно действие, то се описва със сравнително голям брой от тези елементарни операции, но в общия случай те се изпълняват по бързо от съответните CISC инструкции.
- Инструкциите са с фиксиран формат.
- Всяка инструкция се изпълнява за един процесорен такт, което позволява да се предсказват по-точно преходите в програмите и да се изпълняват повече от една инструкция за такт.
- Инструкциите могат да се изпълняват и конвейерно /докато дадената инструкция се изпълнява, следващата се разшифрова/.
- Произвеждат се по лесно, защото в имат по малко транзистори и по-просто организирана вътрешна логика.

**MISC Minimal Instruction Set Command** (минимален набор дълги команди)

Увеличаването разрядността на процесорите, довежда до идеята за събиране на няколко команди в една дума с размер 128 бита. По този начин процесорът има възможност да изпълнява едновременно няколко команди, съответно няколко потока от данни.

MISC, както и процесорите RISC се характеризират с неголямо число често срещани се команди.

**MISC обединява в едно цяло супер скаларната (много поточна) и VLIW концепцията.** Компонентите на процесора са прости и работят с високи скорости.

**VLIW** (набор от команди със свръхголяма разрядност)

Идеята се заключава в това, че се създава специален компилатор за планиране, който преди изпълнение на приложната програма я анализира и по множеството клони от последователности определя групата команди, които могат да се изпълняват последователно. Всяка такава група образува една свръх дълга команда. Това позволява да се решат две важни задачи:

- В течение на един такт се изпълняват група кратки ("обикновени") команди.
- Опростява се структурата на процесора.

Принципът на "много дълги команди" осигурява изпълнението на група команди за един цикъл на процесора.

Порядъкът на изпълнение на командите са разпределя така, че в максимална степен да се натоварят маршрутите, по които преминават потоците от данни.

По това VLIW се отличава от супер скаларната архитектура, при която подбора на групата едновременно изпълнявани команди става едновременно в хода на изпълнение на приложната програма, което усложнява структурата на процесора и забавя скоростта на работата му.

Технологията VLIW е резултат от изследванията на корпорациите HP и Intel.

## 4.ОРГАНИЗАЦИЯ НА ПРОЦЕСОРА



### КОМАНДИ

#### Основни понятия

##### *Програмен език*

Изкуствен език, състоящ се от ключови думи и правила /синтаксис/, който се използва за създаване на инструкции, които компютърът да следва.

##### **Задача**

Управление на процесите по обработка и обмен на данните.

##### *Инструкция*

Специализирана команда, при изпълнението на която процесорът извършва конкретно действие, заложено в програмата.

##### **Формат на командата**

Определя структурата на командата, кода, дължината и метода за разположение на адресите.

Код на операцията	Адресен код
-------------------	-------------

Код на операцията /машинен код/

Кодирана инструкция, част от набора инструкции, специфични за всеки тип процесор. Съобщава къде и какво трябва да се направи, както и къде да се съхрани получения резултат.

Адресен код

Код с помощта на който се адресират клетките от паметта, където се намират операндите, върху които трябва да се изпълни операцията.

##### *Операнди*

Данни обработвани вътре в процесора..

##### **Изпълнение на командата**

Всяка команда се изпълнява от процесора за един или няколко команди в пет (класика) или повече фази.

##### **Макрокоманда**

Последователност от взаимосвързани команди.

#### Типове команди

##### *Аритметични и логически*

Целочислени аритметически и логически операции: събиране, изваждане, логическо събиране, логическо умножение и т.н.



### ***Прехвърляне на данни***

Прехвърлят данни от едно устройство на компютъра към друго: процесор - оперативна памет; процесор - вход или изход и т.н.

### ***Управление на потока команди***

Безусловни и условни преходи, активиране на процедури и т.н.

#### **За безусловен преход**

Предава управлението на инструкция, която не е поредна. Адреса на прехода се записва в брояча на инструкции.

#### **За условен преход**

Променя реда на изпълнение на инструкциите, само ако е изпълнено определено логическо условие. Пример: ако  $x = 200$ , изпълни инструкция 3040.

#### **Системни операции**

Системни повиквания, команди за управление на виртуалната памет и т.н.

## **МЕТОДИ ЗА АДРЕСАЦИЯ**

Адресация - процес на формиране изпълнимите адреси на клетки от паметта, където се намират операндите, необходими за изпълнение на текущата инструкция. За формиране на изпълнимите адреси се използват вътрешните регистри на процесора.

□ Регистри с общо предназначение - компоненти на процесора, с помощта, на които се извършва процеса на адресация.

□ Метод на адресация - начин за манипулиране на командата, чрез който могат да се формиране на адреси на константа, регистър или клетка от паметта. За адресиране на клетка от паметта процесорът трябва да изчисли действителен или ефективен адрес, чрез зададения в командата метод за адресация.

## **КОМПЮТЪРНИ ПРЕКЪСВАНИЯ**

Механизми, които заставят процесора да прекъсне за кратко време изпълнението на текущата програма и да обработи прекъсването. Когато се появи прекъсване, информацията за това, което е правил процесорът за момента се записва в стека.

### **Машинни**

Хардуерните прекъсвания са предизвикани от различни компоненти и се предават на процесора от контролера на прекъсванията/часовник, таймер, диагностика и др/.

### **Програмни**

Прекъсвания, явяващи се инструкции на изпълнимата програма. Процесорът разглежда извиканата функция като подпрограма. След изпълнението и продължава с текущата програма.

### **Вътрешни (от операционната система)**

Използват се за осигуряване на мултипрограмна работа на компютъра и за други управляващи функции на ОС.

### **Външни**

Прекъсвания, предизвикани от периферните устройства. Периферните устройства непрекъснато се обръщат към процесора чрез заявки за прекъсване. Всяка заявка за прекъсване идентифицирана по номер се предава на процесора чрез два контролера за прекъсване. Заявките се обработват по ред и значимост.