

Компютърна архитектура III

Публикувано от legolas на 24.02.2010

Компютърна архитектура III. Микропроцесорна фамилия 80x88/86

1. Обща характеристика на I8088/I8086.

16 битов процесор - I8086 - структура 16/16/20, I8088 - структура 8/16/20

16 битова вътрешна шина за данни

16(8) битова външна шина за данни

20 битова адресна шина (*max 1 Mb системна памет*).

Максимална тактова честота - ~ 5 MHz (базов модел).

Адресация на I/O - отделно I/O адресно пространство.

Прекъсвания:

Векторна организация с фиксирани векторни адреси в паметта

до 255 източника

апаратни(външни), програмни (софтуерни), вътрешни

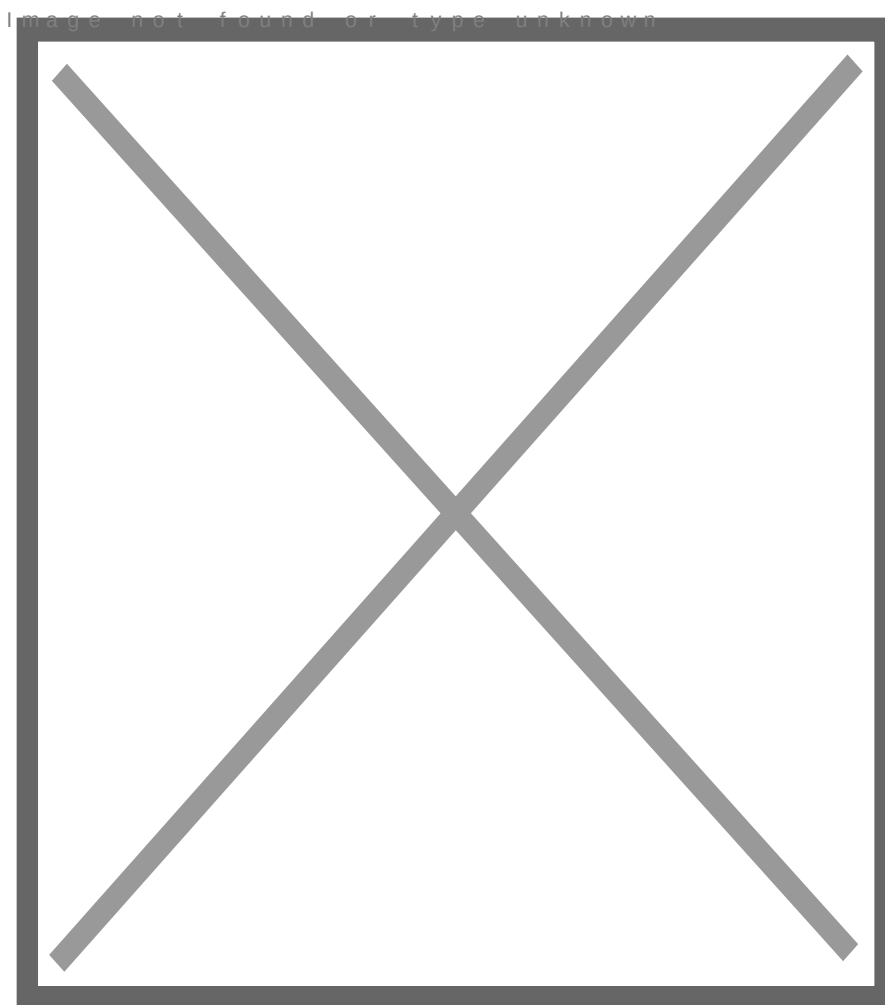
Опашка от команди - FIFO

Извличането (Fetching) се припокрива с изпълнението (Execution) на предишната инструкция:



2. Програмен модел.

Изпълнение Извличане



Регистри - вътрешно съхранение на данни + специфични цели: напр.

BX - базов регистър за адрес

CX - неявен брояч

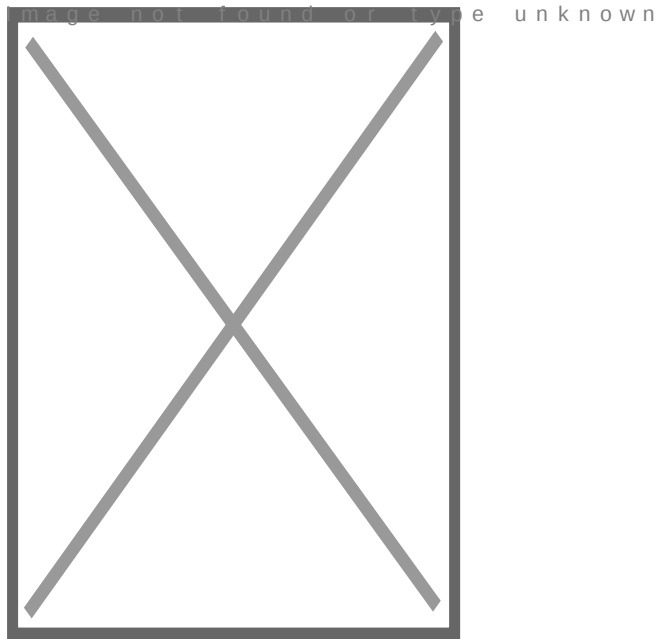
DX - в I/O команди - адрес на порт

Указатели и индексни регистри - формиране на адрес

Сегментни регистри - формиране на адрес

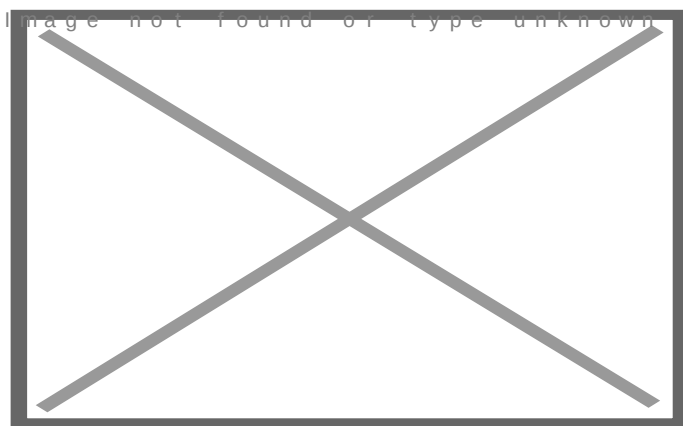
3. Формиране на физически адрес, Сегментна организация на паметта.

Формиране на физически адрес



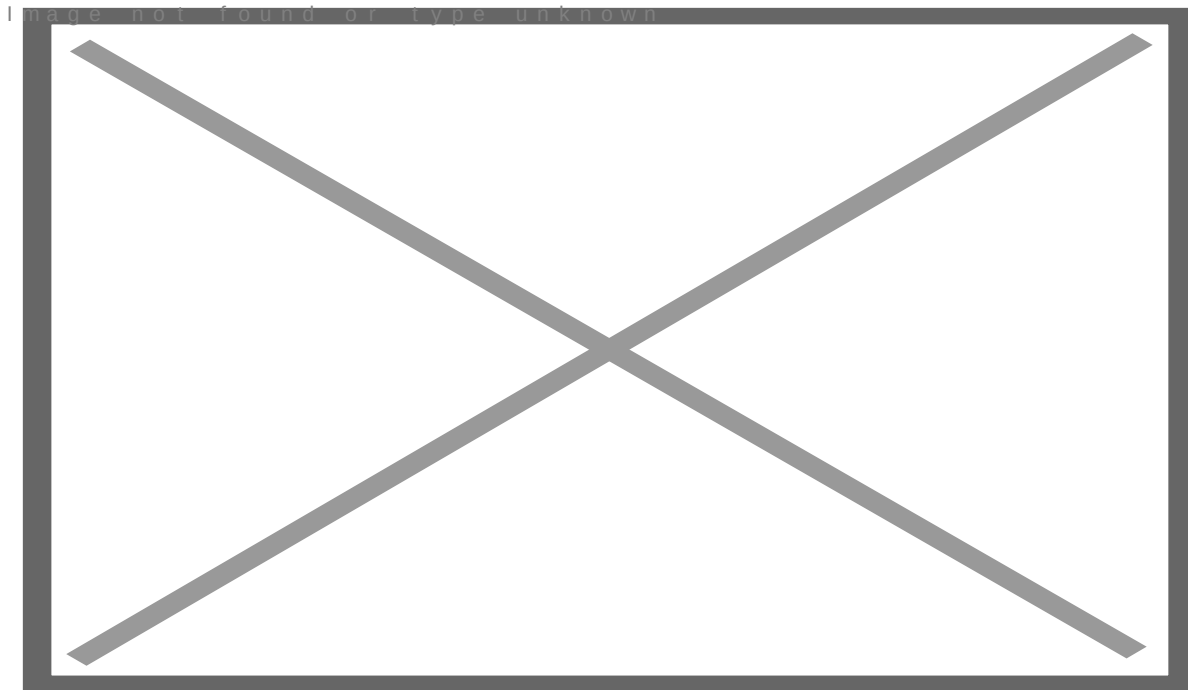
Формиране на физически адрес от segment:offset

- Защо сегментна организация?
 - Съвместимост

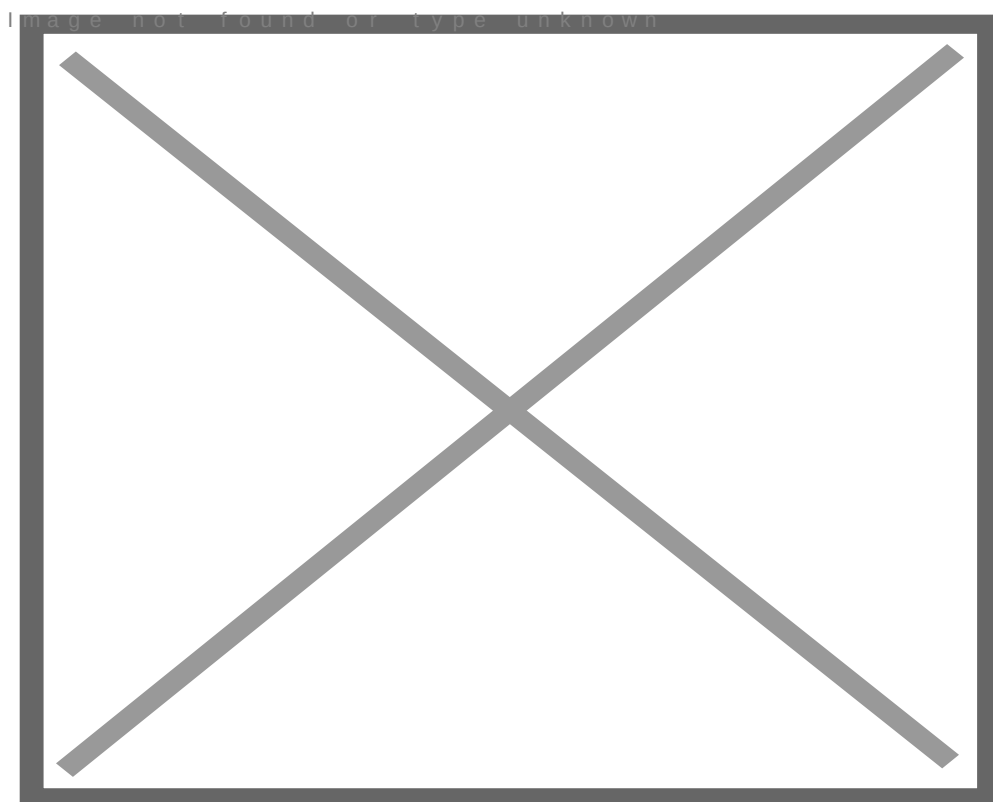


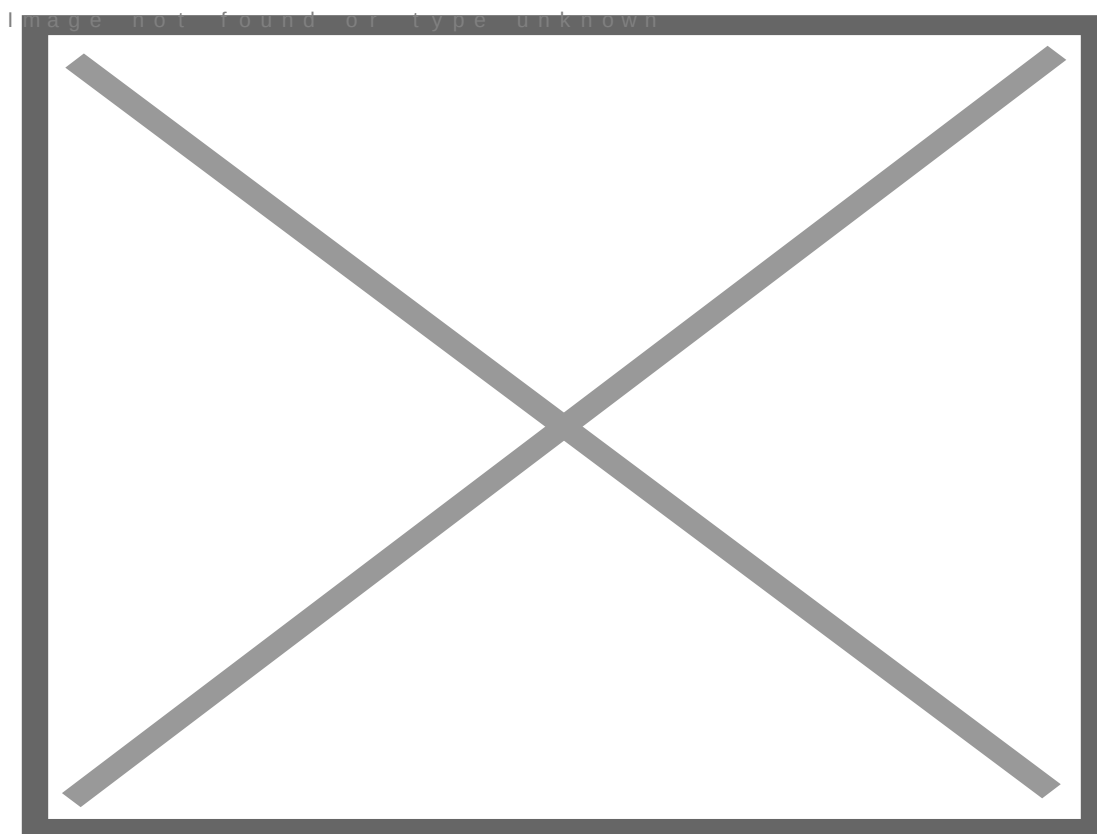
- - Осигурява съответствие между ширината на адресната шина и шината за данни
 -
- Нормално (но може да се форсира друго):
 - IP работи с CS -> CS:IP - извличане на инструкции
 - Обмен на данни - с DS (или ES)
 - Стекови операции - с SS (или ES)

- Пример :



4. Хардуерна структура.





В блока на регистрите за данни (**Data registers**) са включени четири 16-битови регистъра, означени като AX, BX, CX и DX. Във всеки от тях може да се записват операндите и резултата от операциите. Възможно е половинките на тези регистри да се използват поотделно като 8-битови регистри, означени съответно като AH, AL, BH, BL, CH, CL, DH и DL. Всеки от тези регистри може да се използва като **акумулатор** за операндите и резултата от аритметичните и логически операции. Най – често за тази цел се използва регистър AH (16 бита) или AL (8 бита). Регистрите BX, CX и DX освен аритметически имат и следните специфични предназначения:

- **BX** се използва като базов регистър при изчисляване на адресите;
- **CX** се използва в някои команди като неявно адресиран брояч
- **DX** в някои операции с входно – изходни устройства съдържа адреса на клетката от интерфейса, чрез която се извършва обмен на данни с конкретно устройство.

В микропроцесора има 6 регистъра за инструкции (**Instruction queue**) в който се записва (6

байтова) опашка от инструкции за изпълнение. Тези регистри се запълват в моментите кога микропроцесорът изпълнява вътрешни операции. В този момент системната магистрала е свободна и може от паметта да се извлича следващата инструкция. В момента на завършване на командата следващата вече се намира в МП и започва изпълнение. Има случай, в който е нижен преход, в този момент съдържанието на опашката се изтрива и продължава изпълнение от новия адрес.

ALU(аритметико – логическо устройство)

Има две входни 16 битови магистрали – една за първия операнд и една за втория. Освен това има и вход за пренос (заем) от преходни изчисления. Резултатът се подава към изхода на АЛУ.

16 битови – събуране, изваждане

32 битови – умножение, деление

За запис се използват 1 или 2 16 – битови регистъра (AX и DX)

PSW – регистър на състоянието

Той е 16 – битов, но се използват само 9 – бита, означени като флагове на състоянията. Ако в резултат на операцията е възникнал пренос (заем) се активират определени битове от регистъра

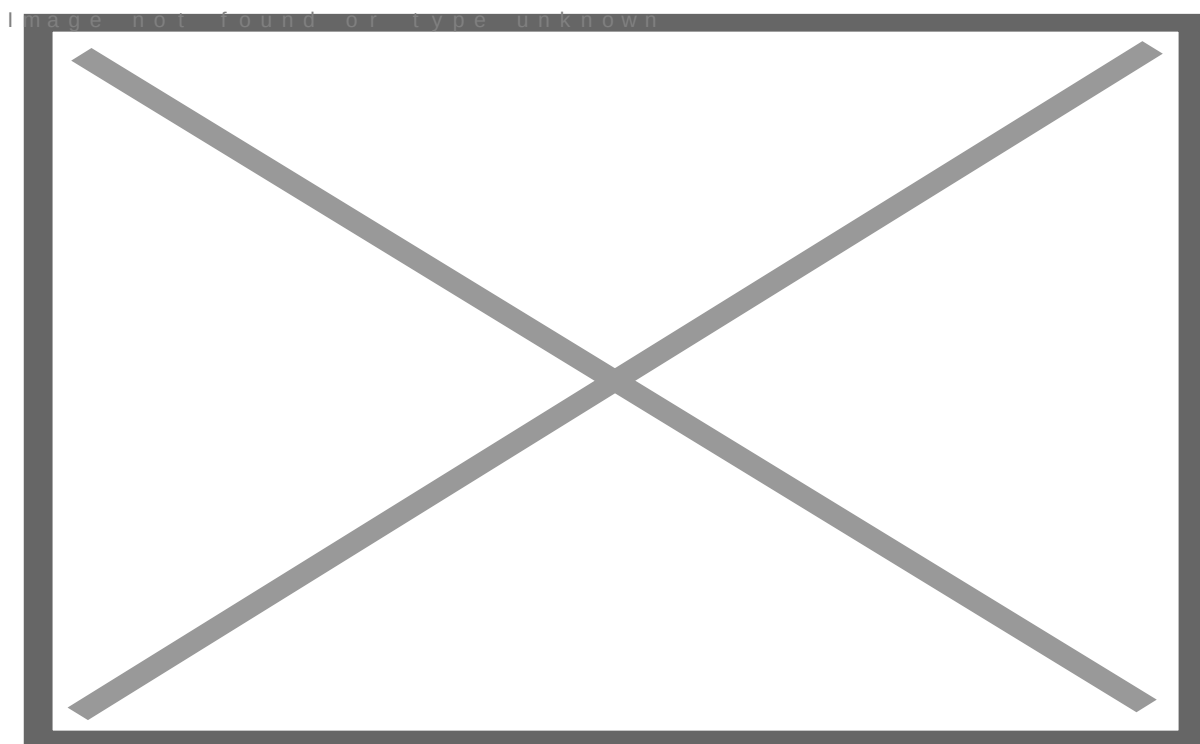
PSW

40 извода - DIP40

22 управляващи сигнали

16 - младша част на адреса **мултиплексирана** с данните

4. Организация на стека.



5. Прекъсвания.

Три вида

Апаратни

маскируеми

немаскируеми

програмни (Пример: INT #)

вътрешни (Пример : деление на 0, TRAP - TF)

RESET - всички reg=0 с изключение на CS=0FFFFh

256 вектора на прекъсване - адреси 00000h - 003FFh (1 Kb) - x4 bytes на вектор - segment:offset

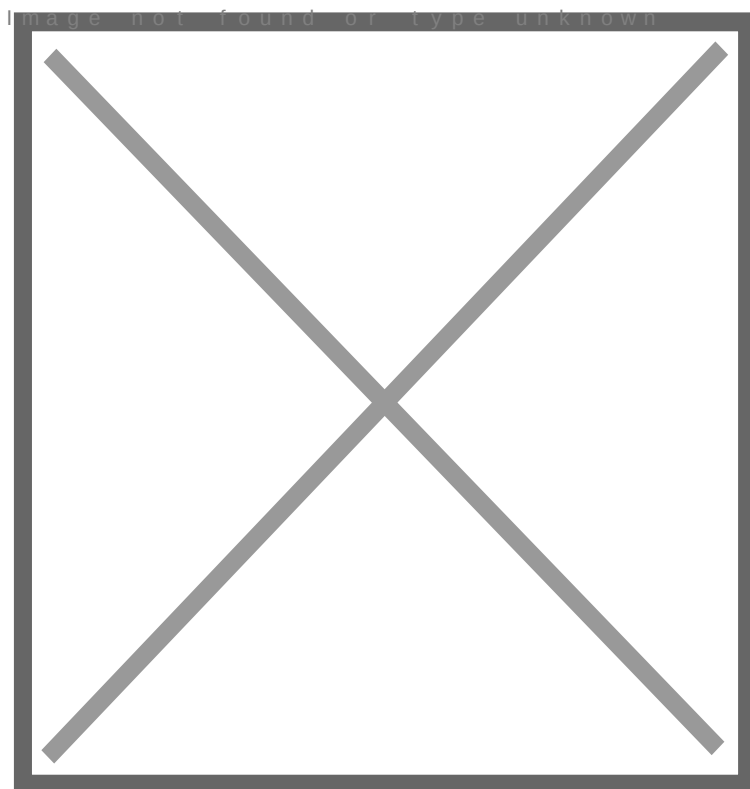
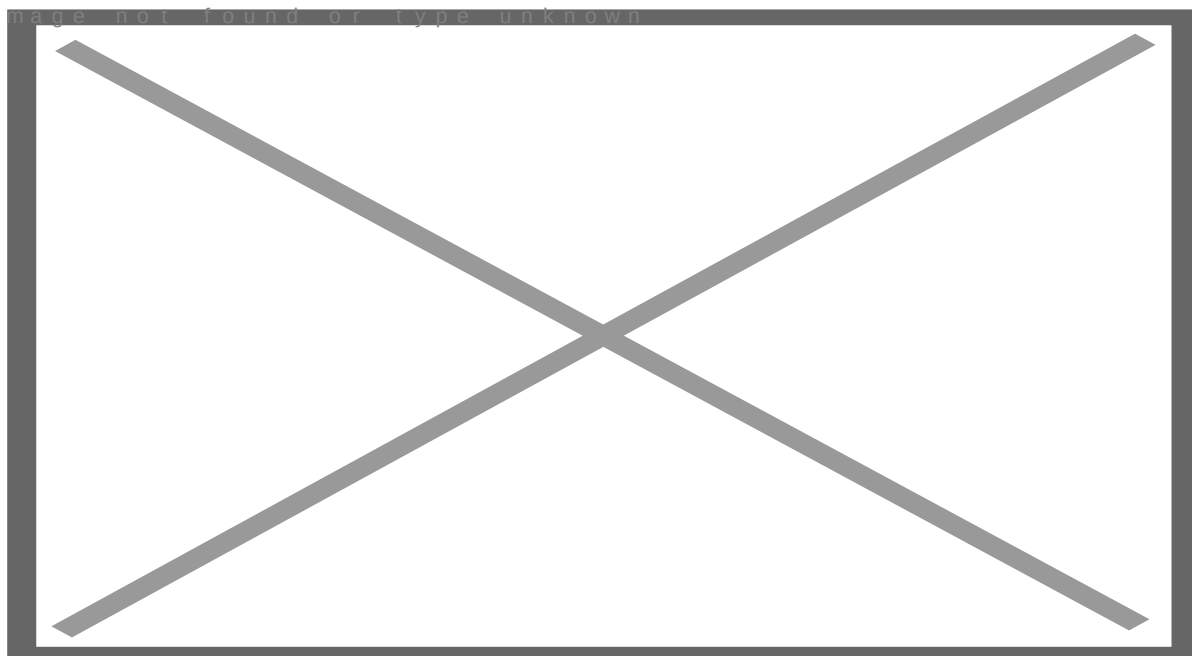
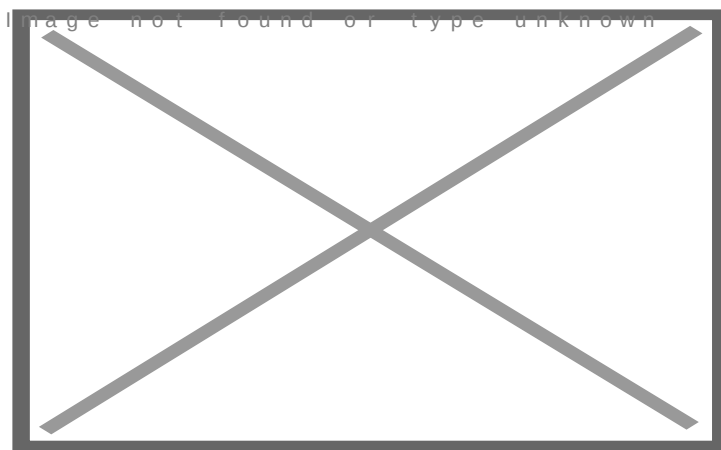


Таблица на векторите на прекъсване в паметта

Протокол при прекъсване - 1. INTR (вход - заявка); 2. INTA (изход - потвърждение) 3. DB (входове - # на прекъсв



- Микропроцесора и контролера на прекъсванията



6. Набор от инструкции (Instruction Set).

Основни групи

Преместване на информация - MOV dest, src, XCHG dest,src - единия операнд е задължително регистър

Преобразувания - XLAT ---- AL<- [[BX]+AL]

Стекови - PUSH src, POP dest

Аритметични - ADD AX, src

DIV -- DX,AX : AX|BX|CX|DX -> DX(остатък)AX(частно)

Логически - AND , XOR

Ротации и премествания (shifts)

Проверка - TEST AX

Обработка на низове -

LODS AL(AX) -> (DS:SI)->AL(AX), SI++ (or--) [see DF)

STOS AL(AX) -> (DS:SI)<-AL(AX), SI++ (or--) [see DF)

Префикси - REP - loop operation CX-- until CX=0

Работа с подпрограми - JSR dest, RET

Условни преходи - JZ dest

Безусловни преходи - JMP dest

Флагове - CLC

Прекъсвания - INT num

Методи на адресация

Непосредствена - MOV AX, #\$FFFA

Пряка - MOV AX, [\$FFFA]

Регистрова - INC AX

Регистрова косвена - MOV AX, [BX]

Регистрова относителна

$EA = [BX|BP|SI|DI] + \{8|16 \text{ бита отместване}\}$

Базова индексна

$EA = [BX|BP] + [SI|DI]$

Относителна базова индексна

$EA = [BX|BP] + [SI|DI] + \{8|16 \text{ бита отместване}\}$

Относителна - условни преходи

Безусловна - безусловни преходи