**АРХІТЕКТУРА ПРОЦЕСОРА 8086.**

**ОРГАНІЗАЦІЯ АДРЕСНОГО ПРОСТОРУ.**

Мінімальною одиницею пам’яті, що має власну адресу, є **БАЙТ**.

Чарунки ОЗП розміром у байт мають послідовні адреси, що починаються з нуля (0, 1, 2 і т.д.). Розмір шини адрес ( тобто кількість розрядів, що відведені для адреси чарунки пам’яті) для процесора 8086 складає двадцять двійкових розрядів. Тому максимально можлива адреса 220 – 1 (1048560 – 1)(10) або дванадцять одиниць у двійковому коді або FFFFF(16).

Може адресуватись також **СЛОВО** (два байти). Слово має адресу свого молодшого байта. Молодший байт слова зберігається за молодшою адресою, старший – за старшою.

**СЕГМЕНТ** – це ділянка пам’яті, що розпочинається на межі параграфу (по адресі, кратній 16(10) або 10(16)), має розмір до 64 Кбайт та може знаходитись у будь якому місці ОЗП. Реальний розмір сегменту визначається кількістю інформації, що міститься в ньому.

Є три головних сегмента.

1. **Сегмент кодів**, у якому зберігається програма, що буде виконуватись. Звичайно перша команда, що буде виконуватись, знаходиться на початку цього сегменту і операційна система передає керування за адресою цього сегменту для виконання програми.
2. **Сегмент даних**, де зберігаються дані, константи, робочі ділянки, необхідні програмі.
3. **Сегмент стеку.** Стек це ділянка оперативної пам‘яті, що організована за принципом записаний останнім читається першим. Стек можна уявити собі як купу тарілок, з якої можна брати тільки верхню тарілку, потім наступну і так далі. Додавати тарілку до купи можна тільки зверху купи. Сегмент стеку використовується для зберігання адрес повернення з процедур та переривань.
4. Може бути ще **додатковий сегмент (даних).**

Адреси початку сегментів зберігаються у спеціальних сегментних регістрах. Сегментні регістри мають шістнадцять розрядів. Тому що початок сегменту встановлено на межі параграфу, молодший розряд адреси (у шістнадцятьковому коді) завжди дорівнює нулю. Він не зберігається. Таким чином у сегментному регістрі зберігаються старші 4 розряди у шістнадцятьковому коді або 16 розрядів у двійковому адреси початку сегменту або **БАЗА**сегменту.

В середені програми усі адреси відносні до початку відповідного сегменту (**ЗМІЩЕННЯ** відносно початку сегменту).

Зміщення складається з двох байтів (від 000016 доFFFF16 або від 0 до 6553510 (64К)).

Записується адреса чарунки пам’яті “база : зміщення”. Наприклад, запис 1000h:0200h означає, що база відповідної чарунки пам’яті 1000h, а зміщення – 0200h.

Адреса у програмі формується як сума вмісту відповідного сегментного регістру та зміщення.

Наприклад, у сегментному регістрі зберігається адреса 045Fh, а зміщення дорівнює 0032h.

Відповідна фізична адреса (20 двійкових розрядів) створюється таким чином:

Адреса у сегментному регістрі 045F0

Зміщення + 0032

04622

(До вмісту сегментного регістру додається ноль, що не зберігається).

Таким чином, маючи 16 бітів у сегментному регістрі та 16 бітів зміщення, можна адресувати 1Мбайт пам’яті.

**Прикладна архітектура процесора 8086(8088)**

ОП – операційний пристрій ШІ - шиний інтерфейс

AX AH AL

BX BH BL

CX CH CL

DX DH DL

SP CS

BP DS

SI SS

DI ES

Крув.

шиною

АЛП 4

ПК 3

Регістр прапорців 2

Вказівник команд 1

Черга команд

Мал.1. Прикладна архітектура процесора 8086 (8088)

Процесор складається з двох основних частин: операційного пристрою (ОП) та шиного інтерфейсу (ШІ). ОП виконує команди. ШІ підготовує команди та дані для виконання. ОП містить арифметично-логічний пристрій (АЛП), пристрій керування (ПК) та десять регістрів. Ці пристої забезпечують виконання команд, арифметичні обчислювання та логічні операції. ШІ складається з блоку керування шиною, черги команд та сегментних регістрів. ШІ виконує три основні функції:

1. керує передачею даних на ОП, у пам’ять, та на зовнішні пристрої вводу - вивіду.
2. чотири сегментних регістри керують адресуванням пам’яті об’ємом до 1 Мбайту.
3. виконує добування команд з ОЗП, черга команд зберігає чотири команди наступні за тїєю, що виконується.

У процесорах 8086/8088 застосована конвеєрна архітектура, яка дозволяє виконувати вибирання кодів інструкцій з пам’яті та їх декодування під час виконання внутрішніх операцій. Конвеєр підвищує продуктивність процесора за рахунок зменшення часу простою його операційних узлів. Конвеєр 8086 має 6 байтову внутрішню чергу інструкцій. Блок попереднього вибирання за наявності двох вільних байтів у черзі намагається заповнити її в той час, коли зовнішня шина процесора не зайнята операціями обміну. Черга у процесорі 8088 зменшена до 4 байт, а попереднє вибирання виконується за наявності 1 вільного байту. Ці відмінномті оптимізують конвеєр з врахуванням розрядності шини даних. Черга обнуляється під час будь якої команди передачі керування (навіть при переході (JMP) на наступну адресу).

Процесори 8086/8088 мають 14 регістрів, що використовуються для керування програмою, адресування пам’яті, забезпечення арифметичних обчислень.

Регістри мають довжину одне слово (16 біт).

СЕГМЕНТНІ РЕГІСТРИ CS,DS,SS,ES.

Зберігають адреси початку відповідного сегменту (без молодшего шістнадцатькового нуля):

CS – сегменту кодів;

DS – сегменту даних;

SS – сегменту стеку;

ES – додаткового сегменту.

РЕГІСТРИ ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ (РЗП) AX,BX,CX,DX.

Можуть адресуватись як 16-бітові, так і як два однобайтових.

АХ - 16-бітовий регістр, АН – його старша частина (лівий байт), АL – його молодша частина (правий байт). Аналогічно BX складається з BH та BL, CX – з CH та CL, DX – з DH та DL.

Призначення регістрів загального призначення.

AX - використовується як суматор, в усіх операціях вводу – вивіду, деяких операціях над рядками і деяких арифметичних операціях.

BX – базовий регістр. Єдиний РЗП, який може використовуватисьяк як база для розширеної адреси. Використовується також задля обчислювань.

CX – регістр – лічильник. Використовується для керування кількістю циклів та зсувів. Також використовується для обчислювань.

DX – регістр даних. Використовується у деяких операціях вводу – вивіду, у операціях множення та ділення великих чисел.

РЕГІСТРОВІ ВКАЗІВНИКИ SP, BP.

Забезпечують доступ до даних у стеку.

SP – вказує на зміщення відносно SS вершини стеку.

BP – вказує на зміщення відносно SS даних усередені стеку.

За допомогою регістру SP організується принцип “записаний останнім – читається першим”. Як працює стек у процесорі 8086? Стек оперує **тількі зі словами**. Заповнюється стек від старших адрес до молодших (зверху униз). Тобто спочатку записується слово за максимальною адресою у стеці, потім за адресою на два меншою (тому що стек працює зі словами) и так далі. Регістр SP зветься вказівником на вершину стеку. Вершина стеку – це останнє записане у стек слово. Вказівник на вершину стеку - це адреса ( точнійше зміщення відносно сегментного регістру SS) останнього записаного у стек слова. Як виконується запис слова у стек. Під час запису слова у стек вміст регістра SP зменшується на два і по отриманному зміщенню записується слово (на мал. 2.б - це слово A0B5). Під час повторного запису вміст SP знову зменшується на два і по отриманному зміщенню записується слово (на мал.2.в – це слово C601).

SP

SS

…………..

Стек до запису слова

SP

SS

A0B5

……………

Стек після запису першого слова

SP

SS

A0B5

C601

……………

Стек після запису другого слова

а б в

# Мал. 2. Запис слів до стеку

При читанні слова зі стеку слово читається за зміщенням, що знаходиться у регістрі SP, а потім вміст SP збільшується на два, тобто вказує на попереднє записане у стек слово (мал. 3.а - б). При читанні наступного слова читається слово, зміщення якого знаходиться у SP, і вміст SP знову збільшується на два (мал. 3. б-в). Таким чином SP завжди містить зміщення останнього записаного до стеку слова. Зміна SP виконується автоматично під час запису та читання слів.

SP

SS

…………..

Стек після читания першого слова

SP

SS

A0B5

……………

Стек після читания другоого слова

SP

SS

A0B5

C601

……………

Стек перед читанням другого слова

а б в

# Мал. 3. Читання слова зі стеку

Регістр BP дозволяє прочитати або записати слово за довільним зміщенням усередені стека. Він містить зміщення слова відносно регістра SS.

ІНДЕКСНІ РЕГІСТРИ SI, DI.

Використовуються для розширеного адресування та у операціях з рядками.

SI - є індексом джерела і використовується сумістно з регістром DS для деяких операцій з рядками.

DI - є індексом приймача і використовується сумістно з регістром ES для деяких операцій з рядками.

ВКАЗІВНИК КОМАНД (ЛІЧИЛЬНИК КОМАНД) IP.

Містить зміщення (відносно СS) команди, яка повинна бути виконана (тобто наступної за тією, що в цей час виконується). За допомогою IP реалізується керування послідовністю виконання команд вашої програми. Коли команда завантажується в пристрій керування для виконання, IP збільшується на число, що доівнює кількості байт, що займає ця команда, і таким чином формує адресу наступної команди (при послідовному виконанні команд). Під час виконання команд перходу зміщення команди, на яку треба передати керування, заноситься до IP.

РЕГІСТР ПРАПОРЦІВ.

Зберегає у вигляді бітових ділянок результати виконання операцій (ознаки нульогвого, відємного результату, переповнення розрядної сітки тощо).

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

X X X X OF DF IF TF SF ZF X AF X PF X CF

Мал.4. Формат регистру прапорців

CF (Carry Flag) – прапорець переносу. Містить значення переносів (“0” або “1”) із старшого розряду під час арифметичних операцій та операціях ссуву та циклічного ссуву.

PF (Parity Flag) – прапорець парності. Перевіряє молодші 8 бітів операцій над даними. Непарна кількість одиниць викликає встановлення цього прапорця в “0”, парна – в “1”. Не слід плутати цей прапорець з бітом контроля на парність.

AF (Auxiliary Carry Flag) – допоміжний прапорець переносу. Встановлюється в “1”, якщо внаслідку виконання арифметичної операції виник перенос з третього біту в регістровій однобайтній команді. Цей прапорець використовується в арифметичних операціях над числами в ASCII- та BCD-форматах.

ZF (Zero Flag) – прапорець нуля. Встановлюється за результатом арифметичних команд та команд порівняння. Якщо результат дорівнює нулю, прапорець встановлюється в “1”, в протилежному випадку – у “0”.

SF (Sign Flag) - прапорець знаку. Встановлюється в відповідності зі знаком результату (старшого біту) арифметичних операцій. У разі додатного результату – “0”, у разі від’ємного –“1”.

TF (Trap Flag) – прапорець трасування. Якщо цей прапорець встановлен в одиничний стан, то процесор переходить у режим покрокового виконання команд.

IF (Interrupt Flag) – прапорець переривання. Якщо стан цього прапорця нульовий, переривання заборонені, якщо одиничний – дозволені (керує тільки маскованими апаратними перериваннями).

DF (Direction Flag) – прапорець напряму. Використовується в рядкових командах для визначення напряму обробки рядка.

OF (Overflow Flag) – прапорець переповнення. Фіксує арифметичне переповнення, тобто перенос у (з) старший (знаковий) розряд під час знакових арифметичних операцій.