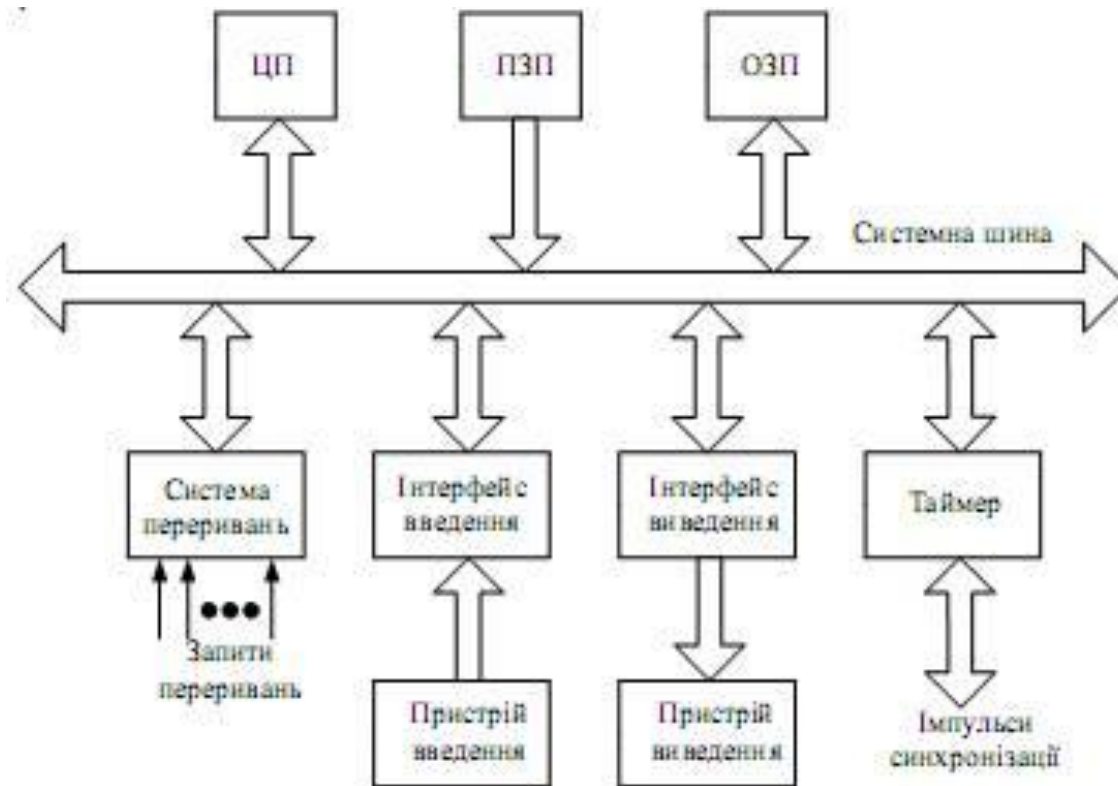


Структура мікропроцесорного пристрою. Архітектурні особливості мікропроцесорів. Поняття про систему команд МП. Типова архітектура універсального МП. Робота та призначення внутрішніх елементів. Основні способи адресації у мікропроцесорах

## Структура мікропроцесорного пристрою



Структура МПП

Узагальнену структурну схему МП пристрою (МПП) чи системи показано на рис. 1. До складу МПП входять: центральний процесор (ЦП), ПЗП, ОЗП; система переривань, пристрої вводу/виводу (ПРВВ), таймер. ПРВВ приєднані до системної шини через інтерфейс вводу/виводу.

Шина - це інформаційний канал, який об'єднує всі функціональні блоки МПП і забезпечує обмін даними. Конструктивно шина являє собою  $n$  провідників. У паралельній шині  $n$  бітів передаються по окремих лініях одночасно. Усі основні блоки МПП з'єднують з єдиною паралельною шиною, яка називається системною шиною (СШ, System Bus). Системна шина містить три шини: адреси, даних і керування.

Шина адреси (ША, Address Bus) - є однонаправленою. Вона призначена для передавання адреси комірки пам'яті або ПРВВ. Напрямок передачі по ША - від МП до зовнішніх пристроїв (ЗП). Позначається



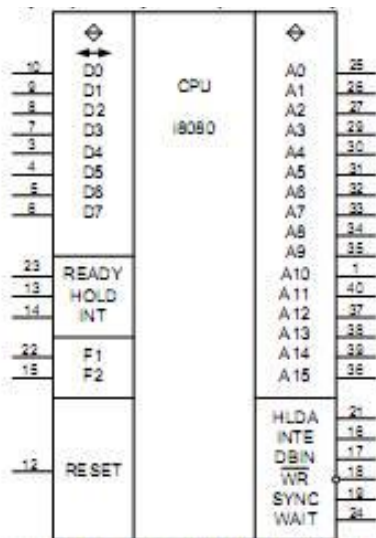
Шина даних (ШД, Data Bus) - є двонаправленою. Вона призначена для передавання даних між блоками МПП. Інформація по ШД передається у двох напрямках по одних і тих самих лініях: як до МП так і від нього. Позначається



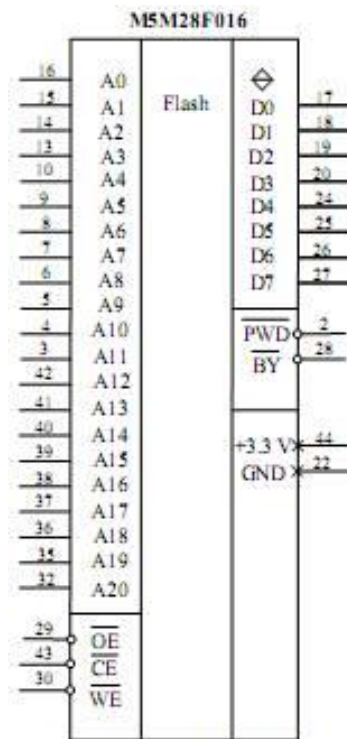
Шина керування (ШК, Control Bus) - призначена для передавання керувальних сигналів. Хоча напрям керувальних сигналів може бути різним, однак ШК не є двонапрямленою, оскільки для сигналів різного напрямку використовуються окремі лінії.

ПЗП та ОЗП складають систему пам'яті, призначену для збереження інформації. ПЗП призначений для збереження програм, таблиць, констант, ОЗП - проміжних результатів обчислень. Пам'ять організовано у вигляді масиву комірок, кожна з яких має свою адресу і містить байт (8 біт) або слово (може мати будь-яку довжину, переважно - 16 біт).

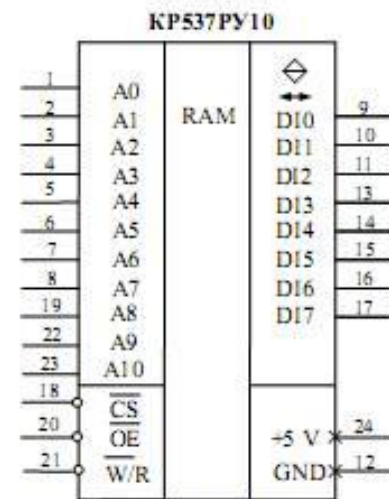
Для звернення до комірки пам'яті треба видати її адресу на ША. На рис. зображено структуру пам'яті з 4 однобайтових комірок



Приклад МП з 16-бітною шиною адреси А0-А15 та 8-бітною шиною даних D0-D7.



Мікросхеми Flash-пам'яті (ПЗП) та статичної оперативної пам'яті (ОЗП)



Робота МПП полягає в опрацюванні даних та інтенсивному обміні інформацією між ЦП, пам'яттю та ПРВВ. Режимми роботи, напрямки передачі інформації задаються сигналами управління, що передаються від ЦП по ШК. При виконанні програми, записаної в ПЗП, мікропроцесор виймає команди з ПЗП, розшифровує їх та виконує дії, що приписуються тією чи іншою командою. При виконанні команд, що включають читання або запис МП звертається до пам'яті, а при виконанні команд вводу-виводу - до ПРВВ. Ефективність вирішення задачі обміну інформацією в МПП в значній мірі визначається організацією цього обміну та структурою зв'язків між мікропроцесором, пам'яттю та ПРВВ.

Основою ЦП є МП, але крім ЦП ще можуть входити мікросхеми тактового генератора, системного контролера, мікросхеми, призначені для буферування СШ. Модуль ЦП обробляє дані та керує всіма іншими модулями системи. ЦП вибирає коди команд з пам'яті, дешифрує їх і виконує. Протягом часу виконання команди - командного циклу ЦП виконує такі дії:

- виставляє адресу команди на ША;
- отримує код команди з пам'яті та дешифрує його;
- обчислює адреси операндів та зчитує їх;
- виконує операцію, визначену командою;
- сприймає зовнішні керуючі сигнали (напр., запити переривань);
- генерує сигнали стану і керування, потрібні для роботи пам'яті та ПРВВ.

ПРВВ або зовнішні пристрої (ЗП) - це пристрої, призначені для введення інформації у МП або виведення інформації з нього. Прикладами ПРВВ є дисплеї, клавіатура, АЦП, ЦАП, реле. Для з'єднання ПРВВ з СШ їх сигнали мають відповідати певним стандартам. Це досягається за допомогою інтерфейсів вводу-виводу.

Інтерфейси вводу-виводу (ІВВ) виконують функцію узгодження сигналів ПРВВ із сигналами СШ. Їх називають також контролерами або адаптерами. Поняття інтерфейсу включає систему шин, допоміжних технічних засобів (наприклад контролери вводу-виводу) та алгоритмів, реалізованих за допомогою програмних та технічних засобів і призначених для організації обміну між МП, пам'яттю та ПРВВ.

Система переривань дозволяє МПП реагувати на зовнішні сигнали - запити переривань, джерелами яких можуть бути: сигнали готовності від ЗП, сигнали з виходу датчиків. Із появою запиту переривання ЦП перериває основну програму і переходить до виконання підпрограми обслуговування запиту переривання.

Таймер призначений для реалізації функцій, пов'язаних з відліком часу. Після того, як МП завантажує в таймер число, яке задає частоту, затримку, або коефіцієнт ділення, таймер реалізує потрібну функцію самостійно.

## Архітектурні особливості мікропроцесорів

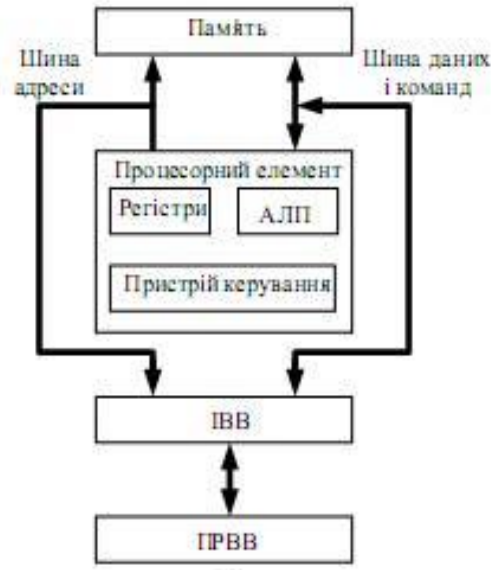
Особливість самого МП, як пристрою з програмованою логікою, полягає у підпорядкованості його структури програмі. Це означає, що функції, які реалізує МП, визначаються не стільки його структурою, скільки послідовністю команд, тобто керуючих інструкцій чи директив, що поступають з пам'яті на виходи МП. При зміні цієї послідовності змінюється функція, яку виконує МП.

Оскільки можливості МП залежать не тільки від його структури, але й від програмного забезпечення, для комплексної характеристики можливостей МП ввели таке поняття як архітектура МП.

Архітектура МП - це поняття, що включає в себе структуру, систему (набір) команд МП, призначення вхідних та вихідних сигналів, формати даних і команд, способи адресації до даних.

Тобто, поняття архітектури об'єднує як апаратні, так і програмні засоби МП. Існують два основних типи архітектури - Прінстонська або Фон-Нейманівська та Гарвардська.

Прінстонська архітектура, яку часто називають архітектурою Фон-Неймана, характеризується використанням спільної пам'яті для зберігання команд (програм), даних, а також для організації стеку. Для звертання до цієї пам'яті використовується спільна системна шина, по якій в мікропроцесор поступають і команди, і дані (рис.).



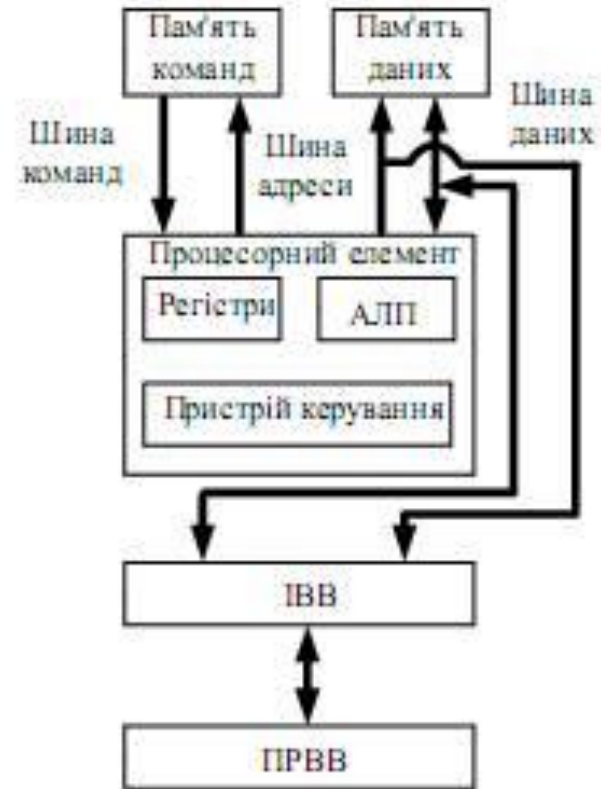
### Фон-Нейманівська архітектур МП

Переваги Прінстонської архітектури. Наявність спільної пам'яті дозволяє оперативно перерозподіляти її об'єм для зберігання команд, даних і реалізації стеку в залежності від виду задач. Таким чином більш ефективно використовується оперативна пам'ять. Використання спільної шини для передачі команд і даних значно спрощує відладку, тестування і поточний контроль функціонування системи, збільшує її надійність. Тому Прінстонська архітектура на протязі довгого часу домінувала в обчислювальній техніці.



Недоліки Прінстонської архітектури. Основний недолік - необхідність послідовної вибірки команд і даних по спільній системній шині. При цьому шина стає вузьким місцем, яке обмежує продуктивність системи. Постійно зростаючі вимоги до продуктивності мікропроцесорних систем викликали в останні роки більш широке застосування Гарвардської архітектури при створенні багатьох типів сучасних мікропроцесорів.

Гарвардська архітектура характеризується фізичним розділенням пам'яті команд (програм) і пам'яті даних (рис.). Кожна пам'ять з'єднується з процесором окремою шиною, що дає змогу одночасно з читанням-записом даних при виконанні поточної команди проводити вибірку і декодування наступної команди. Завдяки такому розділенню потоків команд і даних і суміщенню операцій їх вибірки вдається підвищити швидкодію у порівнянні з Прінстонською архітектурою.



Гарвардська архітектур МП

Недоліки Гарвардської архітектури пов'язанні з необхідністю проведення більшого числа шин, а також з фіксованим об'ємом пам'яті, виділеної для команд і даних, призначення якої не може оперативного перерозподілятися. Тому потрібно використовувати пам'ять більшого об'єму, коефіцієнт використання якої при вирішенні різноманітних задач виявляється нижчим, ніж в системах з Принстонською архітектурою. Проте розвиток мікроелектроніки дозволив в значній мірі подолати вказані недоліки, тому Гарвардська архітектура широко застосовується у внутрішній структурі сучасних мікропроцесорів, де використовується окрема кеш-пам'ять для зберігання команд і даних. Разом з цим, у зовнішній структурі більшості мікропроцесорних систем реалізуються принципи Принстонської архітектури.

Гарвардська архітектура отримала також широке застосування в мікроконтролерах та цифрових сигнальних процесорах.

CISC-процесори (Complete Instruction Set Computing) - процесори з повним набором команд виконують великий набір різноформатних команд з використанням різноманітних способів адресації. Типовим прикладом CISC-процесорів є процесори сімейства Pentium. Вони виконують більше 200 команд різної степені складності, які мають розмір від 1 до 15 байт і забезпечують більше 10 способів адресації. Така різноманітність виконуваних команд і способів адресації дає змогу програмісту реалізувати найбільш ефективні алгоритми вирішення різноманітних задач. Проте при цьому суттєво ускладнюється структура мікропроцесора, особливо його пристрій управління, що приводить до збільшення розмірів і вартості кристалу, зменшенню продуктивності. У той же час багато команд і способів адресації використовуються досить рідко. Також аналіз кодів програм, які генеруються компіляторами мов вищого рівня, показав, що компілятори з усієї системи команд МП використовують тільки обмежений набір простих команд. Це команди типу "регістр-регістр", "регістр-пам'ять". Компілятори не в стані ефективно використовувати складні команди. Все це привело до того, що починаючи з 1980-х років, інтенсивний розвиток отримала архітектура процесорів з скороченим набором команд -RISC-процесори (Reduced Instruction Set Computing).

RISC-процесори використовують обмежений набір команд фіксованого формату (біля 100 команд). Також значно скорочується число способів адресації. Для зменшення кількості звертань до пам'яті RISC-процесори мають збільшений об'єм внутрішніх регістрів - від 32 до декількох сотень, тоді як в CISC-процесорах кількість регістрів загального призначення переважно становить 8-16.

Сприяв розвиткові RISC-процесорів також розвиток конвеєрних процесорів. В них кожна команда розбивається на невелику кількість кроків з однаковим часом виконання (вибирання команди з пам'яті, дешифрування команди, виконання команди, запис результатів і т.п.). Це дозволяє запускати паралельно на виконання з зсувом на один крок багато команд, створюючи таким чином ефективний конвеєр процесора, і за один такт видавати результати виконання команди.

В результаті при використанні RISC-архітектури суттєво спрощується структура МП, зменшуються його розміри та вартість, значно зростає продуктивність.

Зараз в багатьох сучасних CISC-процесорах використовується RISC-ядро, яке виконує обробку даних. В таких МП є вбудований апаратний транслятор, що перетворює складні команди формату CISC в послідовність простих RISC операцій. При цьому одна команда формату CISC може породжувати до  $4^x$  команд формату RISC. Фірма Intel, наприклад, використала цей підхід в МП сімейства Pentium.

## Команди мікропроцесора. Мнемонічна форма запису команд

Обробка даних - одна з основних функцій МП, яка включає арифметичні обчислення, логічні операції та різноманітні маніпулювання з даними.

Елементи структури МП, що виконують цю функцію, складають так званий арифметико-логічний пристрій (АЛП). В результаті роботи АЛП дані змінюються. До операції обробки даних, наприклад, можна віднести:

1. додавання - відповідна команда в наборі (системі) команд МП позначається ADD (від англійського слова (addition));
2. віднімання - SUB (subtraction);
3. логічне множення - ANA (and);
4. порівняння - CMP (compare) та інші.

Програма МП складається з послідовності інструкцій (команд), що зберігаються в пам'яті і виконуються одна за одною. Отже, команди записані в пам'ять визначають майбутню роботу МП.

Всі команди і дані в МПНІ представлені в двійковій системі числення. Програмування в машинних (об'єктних) кодах, тобто в двійкових кодах, є копіткою стомлюючою роботою, оскільки команди представляють собою довгі послідовності одиниць і нулів.

Задача скорочення запису команд, полегшення запам'ятовування та ототожнення їх з фактичним призначенням вирішується шляхом використання мнемонічного позначення команд. Мнемонічне позначення команди означає її скорочений запис. Для цього звичайно використовують перші букви назви операції, що виконується по команді. Наприклад, мнемонічне позначення команди додати (англ. ADDition) вміст регістра В до вмісту акумулятора А має такий вигляд

ADD B,

де ADD - мнемонічне позначення коду операції; В - адреса місцезнаходження операнда (даного); другий операнд знаходиться в акумуляторі А.

Якщо ж команда оперує безпосередніми числовими даними, або адресами областей пам'яті, то в адресній частині команди використовують числа у вісімковій, частіше шістнадцятковій системі.

Наприклад, візьмемо команду, код операції якої має мнемонічне позначення JMP (jump -безумовний перехід). Ця команда вимагає вказування адреси переходу. Повністю запис команди може мати вигляд JMP 5F4EH. Тут адреса переходу вказана шістнадцятковим числом.

Мнемонічна форма запису команд використовується в мові асемблер. Асемблером називається і мова програмування і спеціальна програма-транслятор, що переводить мнемокоди у машинні коди МП (двійкові еквіваленти).

Документація по використанню МП, містить описи відповідних наборів команд з вказівкою кодів операцій, як у вигляді мнемонічного запису, так і в числовій формі.

## Поняття про систему команд мікропроцесора

Кожен МП має свій індивідуальний набір команд або, як ще кажуть, систему команд (СК). Команда, як було сказано вище, представляє собою двійкове слово. Структуру цього слова, тобто будову, прийнято називати форматом команди. Формати різних команд можуть суттєво відрізнятися і складатися з одного, двох, трьох і більше машинних слів. Машинне слово - це двійкове число певної розрядності, з яким оперує МП. У 8-розрядних МП довжина машинного слова дорівнює байту. В 16-розрядного - 2 байти і т. д. Довжина різних команд не однакова, і визначається це способом адресації до операнда.

За форматом набір команд МП можна поділити на дві категорії:

- 1) безадресні команди (без посилання на дані);
- 2) команди зі звертанням до пам'яті або зовнішніх пристроїв.

Безадресні команди містять тільки код операції і складаються з одного машинного слова, наприклад, такі команди як: HLT (HaLT) - ЗУПИН, NOP (No OPeration) - ХОЛОСТА ОПЕРАЦІЯ, RET (RETurn) - ПОВЕРНЕННЯ та інші, які не оперують з даними.

Крім того безадресними вважають команди, в яких дані (операнд) не вказані, але цілком певно визначені. Практично, найчастіше в таких командах мається на увазі слово (дані, операнд), розташоване в одному з внутрішніх регістрів МП.

Прикладами таких команд можуть бути наступні команди: RAR (rotate right through carry) - циклічно зсунути вправо вміст акумулятора і прапорця C; STC (set carry) - встановити прапорець переносу C в одиницю; CMA (complement accumulator) - інвертувати акумулятор, CMC (complement carry) - інвертувати прапорець переносу C та інші.

Формати команд другої групи складаються з кількох машинних слів, щоб задати адресу даних в пам'яті або адресу даних зовнішнього пристрою, інколи може задаватися проміжна адреса або дані.

Наприклад, щоб можна було адресуватися до 256 зовнішніх пристроїв необхідно мати зону адреси у вісім біт, а до пам'яті 64 Кбайт - шістнадцять біт. Відповідно формат команди буде складатися з двох і трьох 8-розрядних машинних слів.

За призначенням всі команди МП прийнято ділити на п'ять функціональних груп:

- 1) переміщення даних;
- 2) перетворення даних (група арифметичних і логічних команд);
- 3) управління програмою;
- 4) вводу-виводу;
- 5) спеціальні.



Команди переміщення даних організують передачу інформації в акумулятор з пам'яті або навпаки, в акумулятор з пристрою вводу-виводу або навпаки, а також обмін інформацією між різними регістрами мікропроцесора. В залежності від команди може здійснюватися переміщення одного або кількох слів даних.

Прикладами команд переміщення є команди: MOV A, M (move accumulator from memory) -переслати байт даних з комірки пам'яті, адреса якої знаходиться у відповідній регістровій парі МП, в акумулятор; MOV B, C (move C to B) - переслати байт даних з регістра С в регістр В; IN 45 - перемістити дані з порту ПРВВ, адреса якого дорівнює 45, в акумулятор МП.

Команди перетворення даних використовуються для перетворення інформації, тобто для виконання арифметичних, наприклад ADD, SUB, логічних, наприклад, AND, OR, дій. Дії виконуються над даними (здебільшого це два слова), які знаходяться в регістрах МП. Завжди одним з регістрів МП, що використовується при цьому, є акумулятор А. Інколи дані знаходяться в акумуляторі МП та комірці пам'яті. Після виконання операції результат завжди залишається в акумуляторі А.

Команди управління змінюють вміст внутрішнього регістра МП, який називається програмним лічильником (Program Counter), або скорочено (PC), вміст якого є адресою наступної команди. Якби не було команд управління, то алгоритм програми складався б з ряду операцій, які виконувалися б тільки послідовно в порядку занесення команд в пам'ять.

Команди управління можуть бути умовними і безумовними.

По команді безумовного переходу вміст програмного лічильника РС, або інколи кажуть лічильника команд, приймає значення, яке вказане в команді безумовного переходу. Тому наступною після команди безумовного переходу буде виконуватися команда, записана за адресою, яка вказана в команді безумовного переходу. Прикладом такої команди може бути команда JMP 8092. Після неї буде виконуватися команда, що записана за адресою 8092.

Якщо команда умовна, то вона супроводжується вказівкою на стан, який повинен бути попередньо перевіреною, наприклад, додатне значення результату або від'ємне, переповнення регістру або відсутність його, наявність нульового результату і т.д. Якщо умова виконується, то має місце перехід на нову адресу, що вказана в команді умовного переходу. В протилежному разі команда умовного переходу пропускається.

Можливість програмувати процедуру перевірки і приймати рішення дозволяє писати програми, що мають кілька гілок, кожна з яких може виконатися тільки після виконання заданої умови. Прикладом такої команди може бути команда RZ (return zero) - ПОВЕРНЕННЯ (з підпрограми) ЯКЩО НУЛЬ.

Команди вводу-виводу організують обмін інформацією з зовнішніми пристроями. Прикладами таких команд є IN 80 або OUT 40. Перша з них означає, що треба ввести інформацію з зовнішнього пристрою, адреса якого дорівнює 80, в акумулятор МП, а друга -вивести інформацію з акумулятора МП в зовнішній пристрій, адреса якого дорівнює 40.

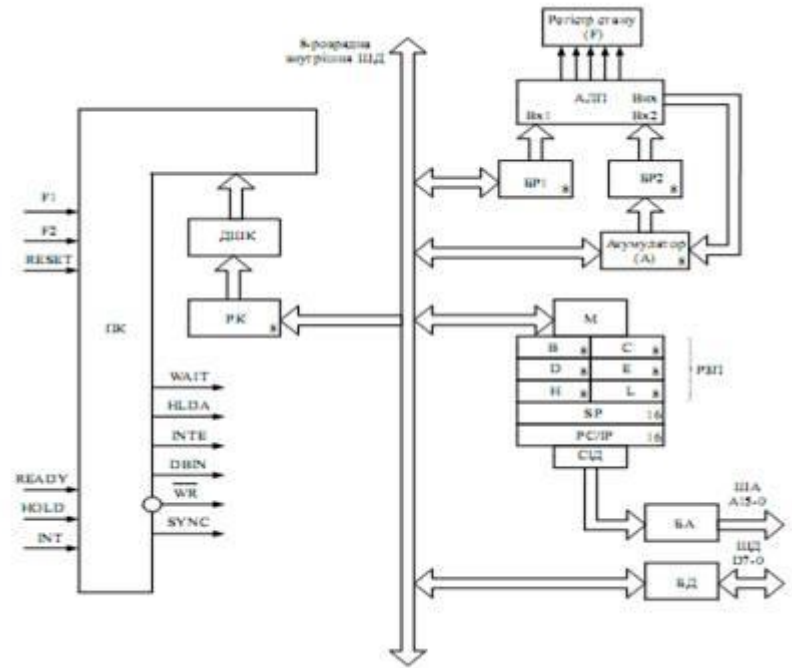
Спеціальні команди призначені для виконання деяких специфічних функцій. Прикладами таких команд можуть бути команди EI (Enable Interrupt) - дозволити переривання програми; DI (Disenable Interrupt) - заборонити переривання програми.

Типова структура 8-розрядного однокристального МП Intel 8080 представлена на рис.

Як видно зі схеми МП складається з таких основних чотирьох частин: 1) арифметико-логічного пристрою (АЛП); 2) множини регістрів; 3) пристрою керування; 4) 8-розрядної внутрішньої шини даних (ВШД).

АЛП призначений для виконання однієї з головних функцій МП - обробки даних. АЛП являє собою комбінаційну схему, яка виконує арифметичні чи логічні операції - додавання, віднімання, І, АБО, зсув і т.д. Він має два входи та один вихід, через які вводяться і виводяться дані.

Перед входами АЛП розташовані буферні регістри (БР), які можуть зберігати для АЛП по одному слову даних. Буферні регістри приймають дані з ВШД, крім того один буферний регістр ще має можливість приймати дані із спеціального регістра, що зветься акумулятором (А). Після виконання операції дані (результат) пересилаються з АЛП через його вихід в акумулятор А.



В залежності від виду операції АЛП оперує з одним або двома словами. При маніпуляціях з одним словом це слово вибирається з акумулятора А.

Регістри МП приймають участь в реалізації основних функцій. Їх можна розділити на регістри загального (РЗП) та спеціального призначення. РЗП можуть використовуватися програмістом за його розсудом. До них відносяться 8-розрядні регістри В, С, D, E, H, L. До регістрів спеціального призначення належать - А, F, РК, БА, БД, ІР/РС, SP.

Акумулятор (А) - це регістр, призначений для зберігання даних, що надходять з АЛП або з ВШД. Він є основним регістром МП при різних маніпуляціях з даними, бо для виконання будь-якої операції над ними їх перш за все необхідно помістити в акумулятор. Розрядність акумулятора відповідає довжині слова МП. Інколи розрядність акумулятора може бути подвійної довжини, або МП може мати кілька акумуляторів. Напр. важливих регістрів, у якому зберігається адреса команди, що виконується.

Щоб пристрій міг виконувати свої функції, команди в МП повинні поступати з пам'яті в строго визначеному порядку. Цей порядок задає лічильник команд. Розрядність лічильника команд РС часто більша ніж розрядність самого МП. Так у 8-розрядних МП, що можуть адресуватися до пам'яті 64 Кбайт, число розрядів РС дорівнює 16.

Лічильник команд інколи ще називають вказівником команд. Він з'єднаний через мультиплексор з ВШД МП. Тому в принципі він може одержувати адреси команд з будь-якого вузла МП, під'єданого до ВШД.

Коли МП тільки починає працювати, то по сигналу початкової установки RESET в лічильник команд РС завантажується заздалегідь визначена адреса (звичайно це адреса 0000H), і виконання програми починається з цієї адреси.

Після читання першої команди програми за вказаною адресою МП автоматично дає приріст лічильнику команд РС. Цей приріст РС одержує в той момент, коли МП приступив до виконання команди. Тобто з цього моменту РС вказує адресу наступної команди. Лічильник команд може відступати від такої послідовності, коли зустрічається команда безумовного переходу JMP ADDR або коли є необхідність виконати підпрограму (команда CALL ADDR) або такі ж команди умовного переходу чи виклику підпрограми та в результаті виконання деяких інших команд. При цьому адреса переходу вказується в самій команді і з неї завантажується в лічильник команд.

Буферний регістр адресу (БА). При кожному звертанні до пам'яті або порту вводу-виводу БА вказує адресу, яка буде використана МП. Вихід регістра адреси називають шиною адреси. Найчастіше БА завантажується від РС. Але інколи може завантажуватися і від інших елементів, наприклад, від регістрових пар, які можуть використовуватися як вказівники адрес комірок пам'яті. Протягом часу виймання команди з пам'яті БА і РС мають однаковий вміст, тобто БА вказує місце знаходження команди, що виймається з пам'яті. Після декодування команди РС одержує приріст, а вміст БА не змінюється.

Деякі команди надають можливість змінювати вміст БА шляхом виконання обчислень. Оскільки БА підключений до ВШД МП, то він може завантажуватися від різних джерел. Більшість МП мають команди, що дозволяють завантажувати БА вмістом РС, РЗП або якої-небудь області пам'яті.

Звичайно в МП розрядність РС і БА однакова.

Регістр команд (РК) призначений для зберігання коду команди, яка виконується. РК з'єднаний з ВШД і може лише приймати дані. З виходу РК код команди подається на дешифратор команд (ДШК), який команду дешифрує і повідомляє МП, що йому робити для реалізації команди.

Розрядність РК співпадає з розрядністю МП.

Мультиплексом (М) - пристрій, що з'єднує один з РЗП з ВШД.

Регістр стану або прапорців F призначений для зберігання деяких перевірок, що здійснюються в процесі виконання програми. Тобто при виконанні операції в АЛП розряди регістра F можуть приймати те чи інше значення, яке і є результатом здійснюваних перевірок. Зберігання результатів перевірок в регістрі F дозволяє реалізувати програми, що містять умовні переходи (тобто порушувати задану послідовність виконання команд).

Наприклад, якщо при додаванні двох 8-бітних чисел одержується результат більший ніж

1111 1111,

то з'являється біт перенесення, який в свою чергу встановлює в 1 однойменний розряд регістра F. Якщо перенесення нема, то цей розряд буде встановлений в 0.

Або якщо після якої-небудь операції всі розряди акумулятора приймають значення 0, то в регістрі F відповідний біт нульового стану встановлюється в 1.

Формат регістра стану F 8-розрядного однокристального МП i8080 подано на рис.

7	6	5	4	3	2	1	0
S	Z	x	AC	x	P	x	C

Рис. Формат регістра стану F МП I8080

Коротко охарактеризуємо розряди регістра F.

1. Розряд перенесення C. 1 в даному розряді показує, що остання операція, яка була виконана, супроводжувалася перенесенням (від'ємним переносом). Нагадаємо, що при додаванні маємо перенесення з старшого розряду, а при відніманні більшого числа від меншого маємо переміщення у старший розряд.
2. Розряд нульового результату Z. 1 в цьому розряді означає, що після виконання операції у всіх розрядах акумулятора (або іншого регістра) є нулі.
3. Знаковий розряд S встановлюється в 1, якщо старший розряд результату операції дорівнює 1, тобто результат від'ємний.
4. Розряд парності P. 1 в цьому розряді означає, що число одиниць в результаті операції парне.
5. Розряд додаткового переносу AC встановлюється в 1 при появі переносу з 3-ого розряду в 4-ий при операціях додавання та віднімання. Ознака використовується при обробці двійково-десяткових чисел.

Як бачимо, не всі розряди регістра F використовуються МП. Розглянемо приклади.

MVI A, 26 ; Занести в регістр A число  $26_{10} = 00011010_2$

MVI D, 42 ; Занести в регістр D число  $42_{10} = 00101010_2$

ADD D ; Додати до акумулятора A регістр D

+ 00011010 00101010

0100 0100 =  $68_{10}$  Прапорці: Z = 0, C = 0, P = 1, S = 0, AC = 1.

---

MVI A, 11 ; Занести в регістр A число  $11_{10} = 00001011_2$

MVI D, -5 ; Занести в регістр D число  $-5_{10} = 11111011_2$

ADD D ; Додати до акумулятора A регістр D

+ 00001011 1 1 11 1011

00000110 =  $06_{10}$  Прапорці: Z = 0, C = 1, P = 1, S = 0, AC = 1.

---

MVI A, 7 ; Занести в регістр A число  $7_{10} = 00000111_2$

MVI D, 15 ; Занести в регістр D число  $15_{10} = 00001111_2$

SUB D ; Відняти від акумулятора A регістр D

000001 1 1 000011 1 1

- 1 1 11 10 00 =  $-8_{10}$  Прапорці: Z = 0, C = 1, P = 0, S = 1, AC = 1.



16-розрядний вказівник стека SP (Stack Pointer) призначений для зберігання адреси комірки пам'яті, яка є початком області пам'яті, відведеної під стек. Вміст SP зменшується на 1 перед кожним занесенням слова в стек, або збільшується на 1 після кожного витягування слова зі стека.

Регістри загального призначення (РЗП). Ці регістри всі є програмно-доступними і мають багатоцільове призначення, через що і одержали назву регістрів загального призначення. Вони можуть служити як комірки надоперативного запам'ятовувального пристрою (НОЗП). Функціональні можливості деяких з них не поступаються можливостям А. Це досягається тоді, коли АЛЛІ може розташовувати в них дані.

Звичайно, 8-розрядний МП має шість 8-розрядних програмно-доступних регістрів В, С, D, E, H, L, які можуть використовуватися цілком окремо і незалежно. Крім цього, ці регістри можуть використовуватися як три 16-розрядні програмно-доступні пари ВС, DE, HL, які також можуть позначатися тільки першими буквами, тобто В, D, H.

Регістрові пари В, D, H звичайно використовуються в ролі регістрів-вказівників при непрямій адресації, коли виконуються арифметичні, логічні операції та операції. При цьому основною регістровою парою є пара H.

При додаванні 16-розрядних слів регістрові пари В, D, H використовуються для їх зберігання. При цьому вміст будь-якої пари побайтно може сумуватися в АЛЛІ з вмістом регістрової пари H і результат заноситься в регістрову пару H. Тобто при додаванні 2-байтових чисел регістрова пара H виконує роль акумулятора.

Схема інкремента/декремента (СІД) - пристрій, який дозволяє без участі АЛЛ збільшити або зменшити на одиницю вміст одного з РЗЛ, РС або БР.

Пристрій керування (ПК) відповідно до дешифрованих кодів команд та зовнішніх керувальних сигналів генерує керувальні сигнали для всіх блоків структурної схеми.

Для ЛК дуже важливими є вхідні лінії мікропроцесора, що з'єднують його з генератором тактових імпульсів, які синхронізують роботу МП. Вхідні тактові сигнали можуть бути однофазними або двофазними.

ЛК генерує керуючі сигнали у відповідності з початковою фазою синхросигналів Б1 чи і формує в ці моменти часу сигнали для зовнішніх по відношенню до МП компонентів, таких як, наприклад, пам'ять або пристрої вводу-виводу інформації.

В ролі генератора тактових імпульсів звичайно використовується генератор, стабілізований кварцовим резонатором. Сам генератор може бути зовнішнім по відношенню до МП, як, наприклад, це є у мікропроцесора і8080, або внутрішнім - як це є у мікропроцесора і8085, хоча сам кварцовий резонатор і тут є зовнішнім.

Внутрішня шина даних МП - це 8-розрядна шина, що з'єднує між собою АЛЛ і внутрішні регістри та забезпечує передачу даних всередині МП.

Кожен функціональний компонент МП завжди підключений до ВШД, однак скористатися нею може лише після одержання відповідного сигналу від ЛК.

Більшість компонентів мають з ВШД двосторонній зв'язок, тобто вони можуть і посилати дані на шину і приймати їх з шини. Зрозуміло, ВШД - це лінія двостороннього зв'язку, по якій передаються слова даних, а не окремі біти.

Програмною моделлю МП називається сукупність програмно доступних реєстрів, тобто тих реєстрів, вміст яких можна зчитати або змінити за допомогою команд.

Програмну модель МП складають акумулятор, РЗЛ, реєстр прапорців, вказівник стеку та вказівник команд.