

Режими роботи мікропроцесорної системи

[1 Режими роботи мікропроцесорної системи](#)

[2 Програмний обмін інформацією](#)

[3 Обмін за перериваннями](#)

[4 Обмін з використанням прямого доступу до пам'яті](#)

[5 Синхронізація й послідовність дій мікропроцесора i8085](#)

[6 Система переривань i8085](#)

1 Режими роботи мікропроцесорної системи

Мікропроцесорна система забезпечує велику гнучкість роботи, вона здатна налаштуватися на будь-яке завдання. Гнучкість ця обумовлена перш за все тим, що функції, що виконуються системою, визначаються програмою (програмним забезпеченням, software), яку виконує процесор. Апаратура (апаратне забезпечення, hardware) залишається незмінною при будь-якому завданні. Записуючи в пам'ять системи програму, можна змусити мікропроцесорну систему виконувати будь-яке завдання, що підтримується даною апаратурою. До того ж шинна організація зв'язків мікропроцесорної системи дозволяє досить легко замінювати апаратні модулі, наприклад, замінювати пам'ять на нову більшого об'єму або вищої швидкодії, додавати або модернізувати пристрої введення/виведення, нарешті, замінювати процесор потужнішим. Це також дозволяє збільшити гнучкість системи, продовжити її життя при будь-якій зміні вимог до неї.

Але гнучкість мікропроцесорної системи визначається не тільки цим. Налаштуватися на завдання допомагає ще і вибір режиму роботи системи, тобто режиму обміну інформацією по системній магістралі.

Практично будь-яка розвинена мікропроцесорна система (у тому числі і комп'ютер) підтримує три основні режими обміну по магістралі:

- програмний обмін інформацією;
- обмін з використанням переривань (Interrupts);
- обмін з використанням прямого доступу до пам'яті (ПДП, DMA –Direct Memory Access).

2 Програмний обмін інформацією

Програмний обмін інформацією є основним у будь-якій мікропроцесорній системі. Він передбачений завжди, без нього неможливі інші режими обміну. У цьому режимі процесор є одноосібним господарем (Master) системної магістралі. Всі операції (цикли) обміну інформацією в даному випадку ініціюються тільки процесором, усі вони виконуються строго в порядку, згідно виконуваною програмою.

Процесор читає (вибирає) з пам'яті коди команд і виконує їх, читаючи дані з пам'яті або з пристрою введення/виведення, обробляючи їх, записуючи дані в пам'ять або передаючи їх у пристрій введення/виведення. Програма може бути лінійною, циклічною, може містити переходи (стрибки), але вона завжди безперервно і повністю знаходиться під контролем процесора. Ні на які зовнішні події, не пов'язані з програмою, процесор не реагує (рисунок 1). Всі сигнали на магістралі в даному випадку контролюються процесором.

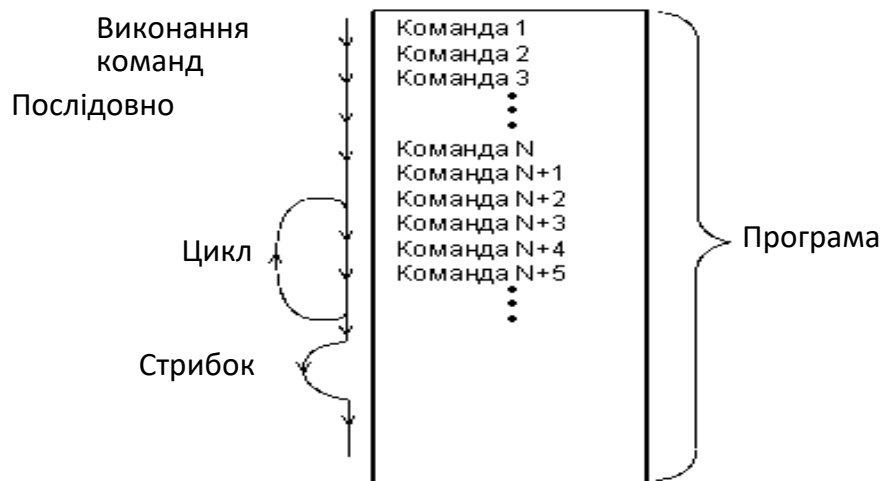


Рисунок 1 – Програмний обмін інформацією

3 Обмін за перериваннями

Обмін за перериваннями використовується тоді, коли необхідна реакція мікропроцесорної системи на якусь зовнішню подію, на появу зовнішнього сигналу. У разі комп'ютера зовнішньою подією може бути, наприклад, натиснення на клавішу клавіатури або надходження по локальній мережі пакету даних. Комп'ютер повинен реагувати на це, відповідно, виведенням символу на екран або ж читанням і обробкою прийнятого по мережі пакету.

У загальному випадку організувати реакцію на зовнішню подію можна трьома різними шляхами:

- за допомогою постійного програмного контролю факту появи події (так званий метод опитування прапора або polling);
- за допомогою переривання, тобто примусового переведення процесора з виконання поточної програми на виконання екстрено необхідної програми;
- за допомогою прямого доступу до пам'яті, тобто без участі процесора при його відключенні від системної магістралі.

Проілюструвати ці три способи можна наступним простим прикладом.

Допустимо, ви готуєте собі сніданок, поставивши на плиту кип'ятитися молоко. Природно, на закипання молока треба реагувати, причому терміново. Як це організувати? Перший шлях – постійно стежити за молоком, але тоді ви нічого іншого не зможете робити. Правильніше регулярно поглядатиме на молоко, роблячи одночасно щось інше. Це програмний режим з опитуванням прапору. Другий шлях – встановити на каструлю з молоком датчик, який подасть звуковий сигнал при закипанні молока, і спокійно займатися іншими справами. Почувши сигнал, ви вимкнете молоко. Правда, можливо, вам доведеться спочатку закінчити те, що ви почали робити, так що ваша реакція буде повільніша, ніж в першому випадку. Нарешті, третій шлях полягає в тому, щоб з'єднати датчик на каструлі з управлінням плитою так, щоб при закипанні молока пальник був вимкнений без вашої участі (правда, аналогія з ПДП тут не дуже точна, оскільки в даному випадку на момент виконання дії вас не відволікають від роботи).

Перший випадок з опитуванням прапору реалізується в мікропроцесорній системі постійним читанням інформації процесором з пристрою введення/виведення, пов'язаного з тим зовнішнім пристроєм, на поведінку якого необхідно терміново реагувати.

У другому випадку в режимі переривання процесор, отримавши запит переривання від зовнішнього пристрою (IRQ – Interrupt ReQuest), закінчує виконання поточної команди і переходить до програми обробки переривання. Закінчивши виконання програми обробки переривання, він повертається до перерваної програми з того місця, де його перервали (рисунк 2).

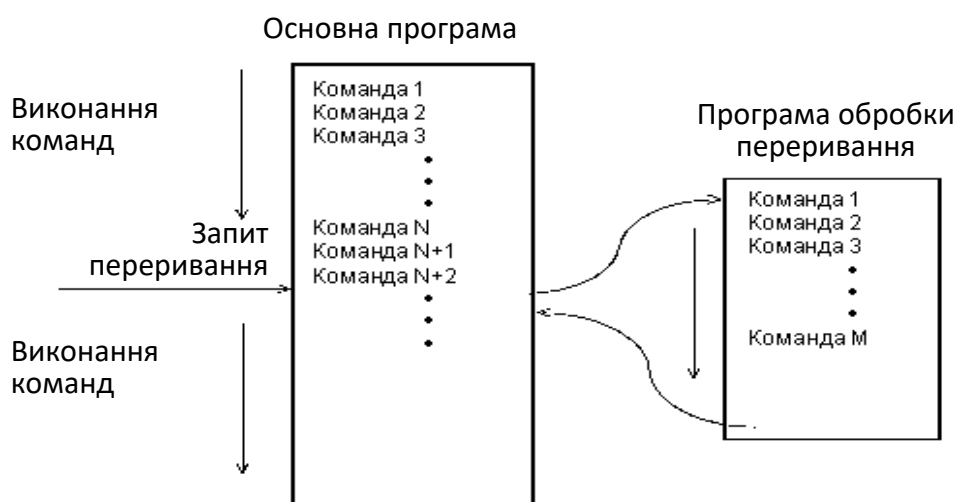


Рисунок 2 – Обслуговування переривання

Тут важливе те, що вся робота, як і у разі програмного режиму, здійснюється самим процесором, зовнішня подія просто тимчасово відволікає його. Реакція на зовнішню подію за перериванням в загальному

випадку повільніше, ніж при програмному режимі. Як і у разі програмного обміну, тут всі сигнали на магістралі виставляються процесором, тобто він повністю контролює магістраль. Для обслуговування переривань в систему іноді вводиться спеціальний модуль контролера переривань, але він в обміні інформацією не бере участь. Його завдання полягає в тому, щоб спростити роботу процесора із зовнішніми запитами переривань. Цей контролер програмно управляється процесором по системній магістралі.

Природно, ніякого прискорення роботи системи переривання не дає. Його застосування дозволяє тільки відмовитися від постійного опитування прапора зовнішньої події і тимчасово, до настання зовнішньої події, зайняти процесор виконанням якихось інших завдань.

Переривання в мікропроцесорних системах бувають двох основних типів:

- векторні переривання, які вимагають проведення циклу читання по магістралі;
- радіальні переривання, які не вимагають ніякого циклу обміну по магістралі.

Переривань в мікропроцесорній системі зазвичай буває багато. Тому процесору необхідна інформація про номер (адреса вектора) конкретного переривання. Ця інформація може бути передана процесору двома шляхами.

При векторному перериванні код номера переривання передається процесору тим пристроєм введення/виведення, який дане переривання запитало. Для цього процесор проводить цикл читання по магістралі, і по шині даних отримує код номера переривання. Шина адреси в даному циклі не використовується, оскільки пристрій, що зробив запит переривання, і так знає, що процесор звертатиметься саме до нього. В цьому випадку в магістралі достатньо всього однієї лінії запиту переривання для всіх пристроїв введення/виведення. Такі переривання організовані, наприклад, в магістралі Q-bus (рисунок 3).

Запит переривання здійснюється негативним сигналом -VIRQ, який може формуватися кожним з пристроїв, що роблять запит переривання. Тип вихідного каскаду для цього сигналу — відкритий колектор, щоб уникнути конфліктів між пристроями, що роблять запит переривання. Отримавши сигнал – VIRQ, процесор надає переривання (закінчивши заздалегідь виконання поточної команди). Для цього він виставляє сигнал читання даних – DIN і сигнал надання переривання IAKO. Цей сигнал IAKO послідовно проходить через всі пристрої, які можуть запитувати переривання. Якщо пристрій зробив запит переривання, то він не пропускає через себе цей сигнал. Якщо переривання одночасно запитали два або більше пристроїв, то сигнал надання переривання отримає тільки один пристрій, а саме той, який

ближче до процесора. Такий механізм вирішення конфліктів називається іноді географічним пріоритетом (або ланковим пріоритетом, Daisy Chain). Отримавши сигнал IAKO, пристрій, що запитав переривання, повинний зняти свій сигнал –VIRQ.

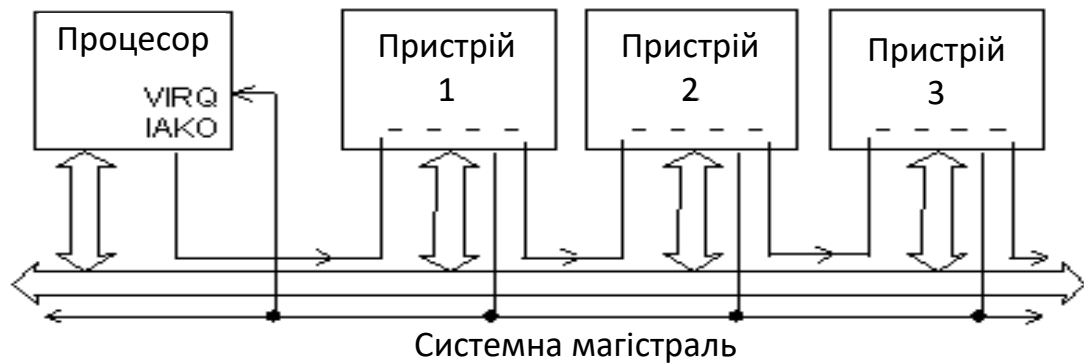


Рисунок 3 – Сигнали запиту і надання переривання в магістралі Q-bus

Далі процесор проводить цикл безадресного читання номера переривання. У відповідь на отримані сигнали – DIN і IAKO пристрій, якому надано переривання, повинен видати на шину адреси/даних код номера переривання (адресу вектора переривання) і виставити сигнал підтвердження – RPLY. Процесор читає код номера переривання і закінчує цикл безадресного читання зняттям сигналів – DIN і IAKO.

При радіальному перериванні в магістралі є стільки ліній запиту переривання, скільки всього може бути різних переривань. Тобто кожен пристрій введення/виведення, для організації обміну за переривання, подає сигнал запиту переривання по своїй окремій лінії. Процесор дізнається про номер переривання за номером лінії, по якій прийшов сигнал запиту переривання. Ніяких циклів обміну по магістралі при цьому не потрібні. У разі радіальних переривань в систему включається додаткова мікросхема контролера переривань, яка обробляє сигнали запитів переривань. Саме так організовані переривання, наприклад, у магістралі ISA.

Спрощена структура зв'язків між пристроями, що беруть участь у обміні за перериваннями, на магістралі ISA показана на рисунку 4. Процесор «спілкується» з контролером переривань як по магістралі (щоб задати йому режими роботи), так і поза магістраллю (при обробці запитів на переривання). Сигнали запитів переривань IRQ розподіляються між усіма пристроями магістралі. На кожен ліній IRQ доводиться один пристрій. Запитом переривання є передній, позитивний фронт сигналу IRQ. При одночасному надходженні сигналів IRQ від декількох пристроїв порядок їх обслуговування визначається контролером переривань.

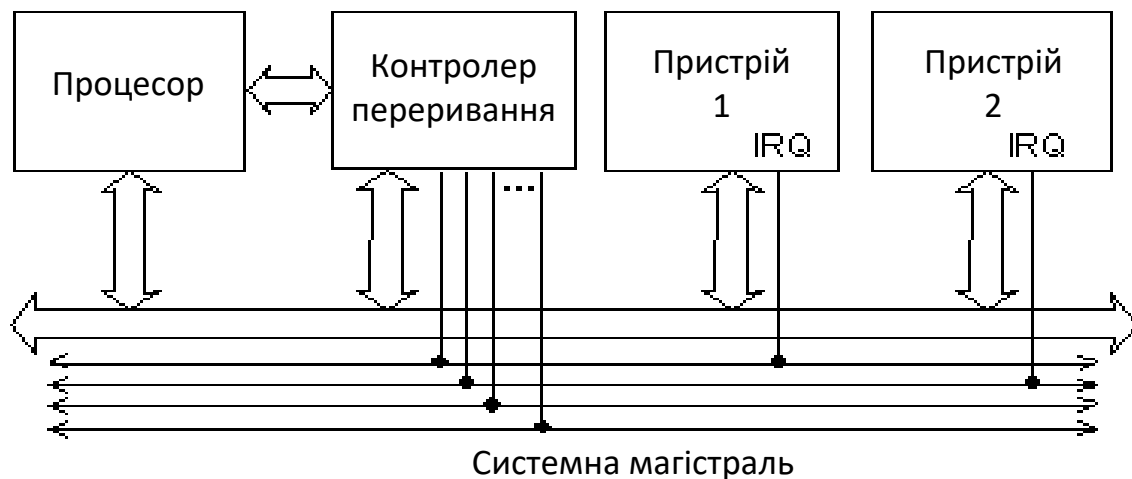


Рисунок 4 – Структура зв'язків для організації радіальних переривань на магістралі ISA

Векторні переривання забезпечують системі велику гнучкість, в системі їх може бути дуже багато. Однак вони вимагають додаткових апаратних вузлів у всіх пристроях, що роблять запит на переривання, для обслуговування циклів безадресного читання.

Радіальних переривань в системі не дуже багато (від 1 до 16). При цьому типі переривань, як правило, потрібне введення в систему спеціального контролера переривань. Кожне радіальне переривання вимагає введення додаткової лінії в шину управління системної магістралі. Але працювати з радіальними перериваннями простіше, оскільки все зводиться тільки до формування єдиного сигналу IRQ, і ніяких циклів обміну по магістралі не потрібні.

4 Обмін з використанням прямого доступу до пам'яті

Прямий доступ до пам'яті (ПДП, DMA) – це режим, що принципово відрізняється від двох раніше розглянутих режимів тим, що обмін по системній шині йде без участі процесора. Зовнішній пристрій, що вимагає обслуговування, сигналізує процесору, що режим ПДП необхідний, у відповідь на це процесор закінчує виконання поточної команди і відключається від усіх шин, повідомляючи пристрою, що запитав, що обмін у режимі ПДП можна починати.

Операція ПДП зводиться до пересилки інформації з пристрою введення/виведення в пам'ять або ж з пам'яті в пристрій введення/виведення. Коли пересилка інформації буде закінчена, процесор знов повертається до перерваної програми, продовжуючи її з того місця, де його перервали

(рисунок 5). Це схоже на режим обслуговування переривань, але в даному випадку процесор не бере участь в обміні. Як і у разі переривань, реакція на зовнішню подію при ПДП істотно повільніше, ніж при програмному режимі.

Зрозуміло, що в цьому випадку потрібне введення в систему додаткового пристрою (контролера ПДП), який здійснюватиме повноцінний обмін по системній магістралі без жодної участі процесора. Причому процесор заздалегідь повинен повідомити контролеру ПДП, звідки йому слід брати інформацію і/або куди її слід поміщати. Контролер ПДП може вважатися спеціалізованим процесором, який відрізняється тим, що сам не бере участь в обміні, не приймає в себе інформацію і не видає її (рисунок 6).

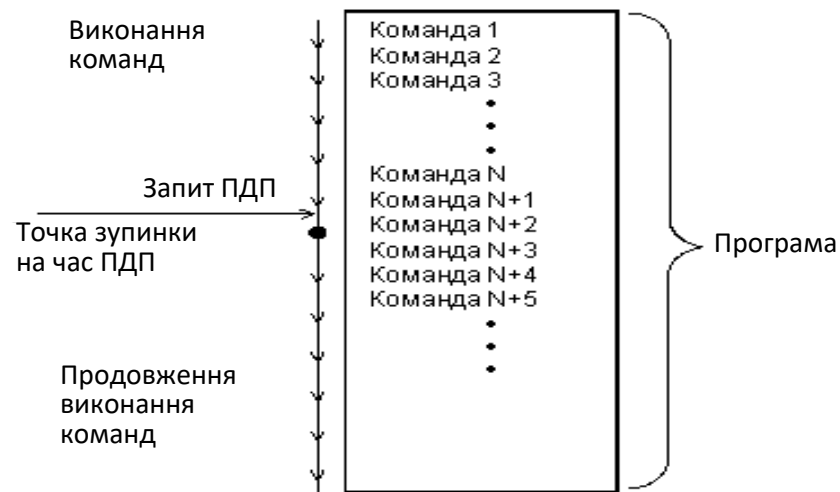


Рисунок 5 – Обслуговування ПДП

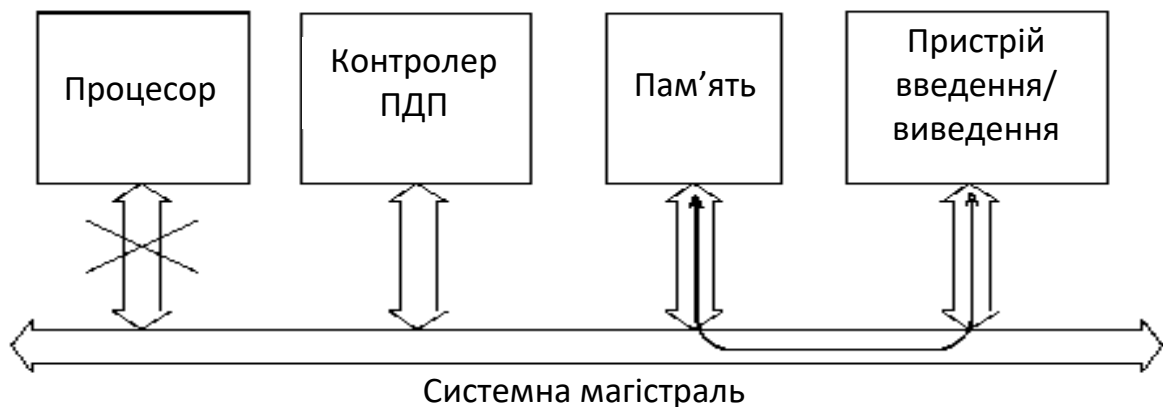


Рисунок 6 – Інформаційні потоки в режимі ПДП

Контролер ПДП може входити до складу пристрою введення/виведення, якому необхідний режим ПДП або навіть до складу декількох пристроїв введення/виведення. Теоретично обмін за допомогою прямого доступу до пам'яті може забезпечити вищу швидкість передачі інформації, ніж

програмний обмін, оскільки процесор передає дані повільніше, ніж спеціалізований контролер ПДП. Проте на практиці ця перевага реалізується далеко не завжди. Швидкість обміну в режимі ПДП обмежена можливостями магістралі. До того ж необхідність програмного завдання режимів контролера ПДП може звести нанівець вигоду від вищої швидкості пересилки даних у режимі ПДП. Тому режим ПДП застосовується рідко.

Якщо в системі вже є самостійний контролер ПДП, то це може у ряді випадків істотно спростити апаратну реалізацію пристроїв введення/виведення, що працюють у режимі ПДП. У цьому полягає єдина безперечна перевага режиму ПДП.

На магістралі Q-bus запит і надання ПДП організуються подібно до запиту і надання переривання. Спрощена структура зв'язків пристроїв, що беруть участь у ПДП, показана на рисунку 7.

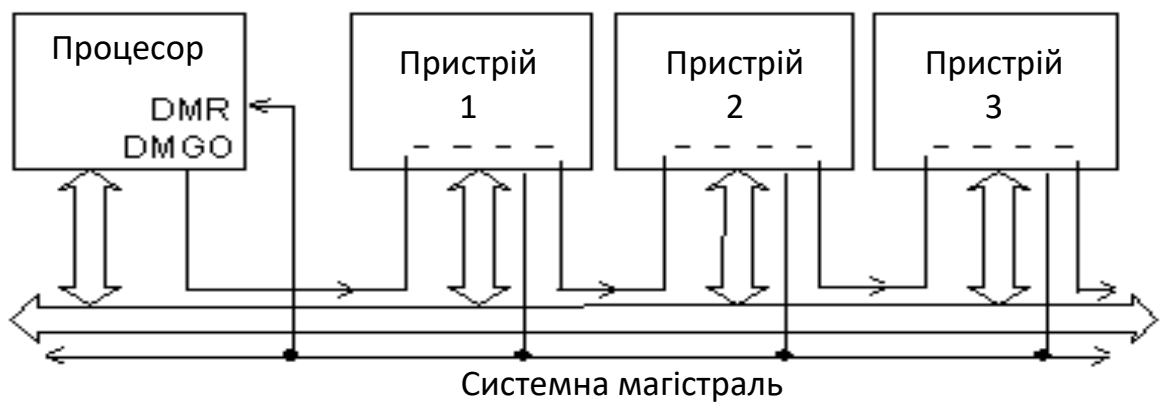


Рисунок 7 – Структура зв'язків запиту/надання ПДП на магістралі Q-bus

Сигнал запиту ПДП (DMR) передається всіма пристроями, що потребують ПДП, по одній лінії магістралі. Процесор, отримавши сигнал – DMR, видає сигнал надання ПДП – DMGO, аналогічний сигналу ІАКО. Цей сигнал також проходить через всі пристрої послідовно, внаслідок чого ПДП отримує тільки той пристрій, який знаходиться ближче до процесора (географічний пріоритет). А потім пристрій, що отримав ПДП, проводить цикли обміну по магістралі, аналогічно циклам програмного обміну. У циклах ПДП інформація читається з пам'яті і записується в пристрій введення/виведення, або навпаки — читається з пристрою введення/виведення і передається до пам'яті.

На магістралі ISA запит/надання ПДП дуже нагадує організацію радіальних переривань (рисунку 8). Так само в системі існує контролер ПДП, до якого сходяться сигнали запиту ПДП, які мають назву DRQ, і від якого розходяться сигнали надання ПДП – DACK. До кожного каналу ПДП (пара сигналів DRQ і –DACK) підключається тільки один пристрій, що запитує

ПДП. Пристрій, що потребує ПДП, посилає сигнал запиту DRQ і отримує у відповідь сигнал надання –DACK. Після цього контролер ПДП проводить цикли обміну по магістралі між пристроєм введення/виведення і пам'яттю.

На магістралі ISA використовуються окремі строби запису до пам'яті (MEMW) і запису до пристроїв введення/виведення (IOW), а також окремі строби читання з пам'яті (MEMR) і читання з пристроїв введення/виведення (IOR). Це дозволяє за один цикл обміну ПДП читати інформацію з пам'яті і записувати її до пристроїв введення/виведення або ж читати інформацію з пристрою введення/виведення і записувати її до пам'яті. При цьому на шині адреси виставляється адреса пам'яті, а адреса пристрою введення/виведення замінюється єдиним сигналом AEN. У циклі обміну в режимі ПДП бере участь тільки той пристрій введення/виведення, який заздалегідь запитав ПДП і якому ПДП було надано. Тому ніяких конфліктів між пристроями введення/виведення із-за такої спрощеної адресації не виникає.

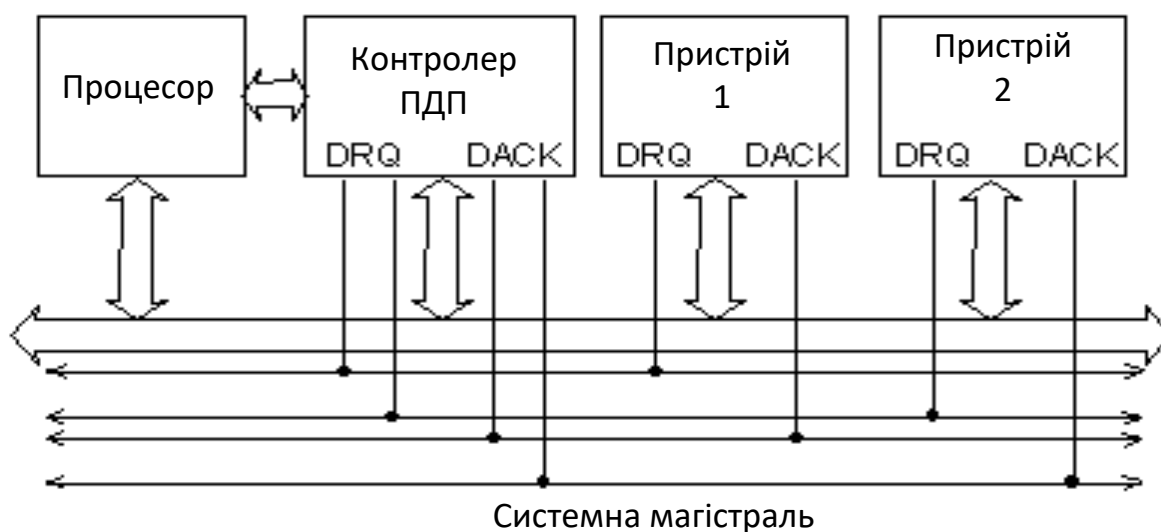


Рисунок 8 – Структура зв'язків запиту/надання ПДП на магістралі ISA.

5 Синхронізація й послідовність дій мікропроцесора i8085

Командний цикл КЦ (рисунок 9, а) починається з вибірки команди (Opcode Fetch, OF). Перший машинний цикл М1 завжди OF, у ньому МП одержує перший байт команди. Після цього можуть бути ще один або два машинних цикли типу MR (Memory Read), оскільки команда може бути однобайтова, двохбайтова або трьохбайтова.

Якщо команда трьохбайтова, то вона зберігається у пам'яті так, як зображено на рисунку 9, б. Перший байт містить код операції КОП,

інформацію про спосіб адресації, а якщо команда однобайтова, то й адресу операнда. Наявність адреси можливо для операцій типу "регістр-регістр" з короткими адресами. Для адресації 8 регістрів загального призначення достатньо трьох розрядної адреси, а для адресації регістрових пар навіть двох розрядної. Другий байт містить молодші напівадреси операнда, якщо команда трьохбайтова, або безпосередній операнд або адресу ЗП, якщо команда двохбайтова. Третій байт містить старші напівадреса операнда або байт безпосередніх даних при завантаженні пари регістрів. Адреси регістрів і регістрових пар наведені у таблиці 1.

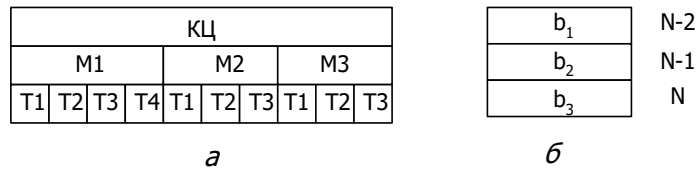


Рисунок 9 – Цикли й такти мікропроцесора K1821VM85A (а) і приклад розміщення команди в пам'яті мікропроцесорної системи (б)

Таблиця 1 – Адреса регістрів та регістрових пар

Регістри							Пари регістрів			
B	C	D	E	H	L	A	B	D	H	SP
000	001	010	011	100	101	111	00	01	10	11

Після вибірки й декодування команди можуть знадобитися додаткові машинні цикли для її виконання. Усього в командному циклі може бути від одного до п'яти машинних циклів.

Машинний цикл (МЦ) складається з тактів, у яких виконуються типові дії. Число тактів у різних машинних циклах – 3...6. Більшість машинних циклів містять три такти.

У командному циклі може бути від 4 до 18 тактів.

Сигнали, що реалізують той або інший МЦ, генеруються блоком керування МП на підставі інформації, яка міститься у першому байті команди.

Розглянемо виконання команди STA b3b2 (Store Accumulator Direct), яка передає вміст акумулятора до комірки пам'яті при прямій адресації. Команда трьохбайтова, для її передачі у МП потрібні три машинних цикли, у першому з яких байт b1 передається до регістру команд IR, у наступні байти b2 і b3 передаються до регістрів тимчасового зберігання W і Z. Після одержання всієї команди МП виконує її, передаючи байт із акумулятора до комірки пам'яті, адреса якої надійшла до МП. Таким чином,

цикл команди складається із чотирьох машинних циклів у наступному порядку OF-MR-MR-MW.

Типові дії, що виконуються у тактах машинного циклу:

- T₁ Адреса пам'яті або ЗП, яка виставляється на AD₇₋₀ і A₁₅₋₈, генерується сигнал ALE для фіксації біт AD₇₋₀. На лініях IO/M, S₁ і S₀ виставляється інформація, яка визначає тип циклу. Перевіряється прапорець HALT.
- T₂ Перевіряються входи Ready і Hold. Програмний лічильник інкрементується, якщо даний машинний цикл є частина вибірки команди. У всіх машинних циклах крім циклу BI (звільнення шин) один з керуючих стробів \overline{RD} , \overline{WR} або \overline{INTA} переходить із одиничного стану у активне нульове.
- T_w З'являється при неготовності пам'яті або ЗП до обміну (на лінії READY низький рівень напруги). Стан ліній адрес, даних і керування залишаються тими ж, що й наприкінці такту. Сигнал READY перевіряється у кожному такті очікування.
- T₃ Байт команди або даних передається до мікропроцесор або з нього. Рівень активного керуючого стробу змінюється з нульового на одиничний.
- T₄ Декодується вміст регістра команд.
- T_{5,6} Використовується при необхідності для завершення деяких команд. Системні шини не використовуються.

Машинний цикл завжди містить такти T₁...T₃, іноді має більше число тактів, але для читання або запису потрібно тільки три такти.

6 Система переривань i8085

Мікропроцесор i8085 має п'ять входів переривання й один вихід керування їм – \overline{INTA} . Переривання повинне ввести в дію команду CALL, відповідно до якого стан програмного лічильника PC передається в стек, а в PC завантажується адреса підпрограми, що підлягає виконанню. Ініціатива введення команди CALL належить апаратним засобам мікропроцесорної системи. Якщо переривання дозволені, то вони здійснюються мікропроцесором наприкінці виконання поточної команди.

Входи МП, пов'язані з перериваннями, називаються TRAP; RST 5,5; RST 6,5; RST 7,5; INTR. При організації переривань розв'язуються завдання маскуванню запитів і визначаються їхні рівні пріоритету при конфліктах через одночасне надходження декількох запитів.

Маскування полягає у забороні дії відповідного входу. Входи запитів переривання можуть маскуватись або не маскуватись. Вхід TRAP є не маскується і має найвищий пріоритет. Він не може бути заборонений командами програми. До цього входу підключають сигнали, що сповіщають про найбільш важливі події у мікропроцесорній системі,

поява яких вимагає негайної реакції (наприклад, сигнал, що сповіщає про аварії живлення).

Початкова адреса підпрограми обслуговування переривання TRAP розташована у фіксованій комірці пам'яті з адресою 24H. Таким чином, поява запиту переривання на вході TRAP викликає відповідне переривання після завершення виконання поточної команди.

Переривання по входах RST_n (n = 5,5; 6,5; 7,5) можуть бути дозволені або заборонені командами EI (Enable Interrupt) і DI (Disable Interrupt), що діють на всі три входи одночасно. Початкове скидання мікропроцесора забороняє обслуговування цих запитів, для їхнього наступного дозволу варто подати команду EI. Пріоритети входів RST_n фіксовані, вони знижуються у порядку RST 7,5; RST 6,5; RST 5,5. Початкові адреси підпрограм обслуговування переривань типу RST_n відомі. Команди RST_n закінчуються завантаженням у програмний лічильник числа 8n. Цифри 5,5; 6,5 і 7,5 визначають початкові адреси 002CH, 0034H и 003CH. Для цих входів вектори переривання визначаються автоматично і їх не потрібно передавати до МП із зовнішніх пристроїв.

При надходженні запиту на вхід INTR (Interrupt) вектор переривання повинен бути переданий у МП ззовні. До цього входу, зокрема, підключають контролер переривань – блок, що сприймає кілька запитів від зовнішніх пристроїв, розв'язує завдання пріоритетності й маскуванню і виробляє для МП єдиний сигнал INTR, з пересиланням у МП відповідного вектора переривання. У цьому випадку також виконується команда RST_n, але n залежить від джерела переривань.