

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КІРОВОГРАДСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура комп'ютерів

КОНСПЕКТ ЛЕКЦІЙ

Семестр II

Кіровоград

2013

ЗМІСТ

МОДУЛЬ I. Керування пам'яттю	2
Тема 1. Взаємодія мікропроцесора з пам'яттю ПЕОМ. Зберігання даних в пам'яті.....	2
Тема 2. Сегментація пам'яті	4
Тема 3. Дескриптори таблиці. Селектори сегментів	8
МОДУЛЬ II. Відеоадаптери	11
Тема 1. Архітектура відеоадаптерів	11
Тема 2. Відеопам'ять комп'ютера.....	14
МОДУЛЬ III. Основи взаємодії комп'ютера з периферійними пристроями	19
Тема 1. Суть і поняття периферійного пристрою.....	19
Тема 2. Матричні принтери та їх програмування.....	20
Тема 3. Лазерні принтери	27
Тема 4. Струменевий принтер	30
Тема 5. Тривимірний принтер (3D-принтер)	30
Тема 6. Контролер клавіатури.....	33
МОДУЛЬ IV. Шинна архітектура комп'ютера	39
Тема 1. Огляд системних шин комп'ютера	39

МОДУЛЬ I Керування пам'яттю

Тема 1. Взаємодія мікропроцесора з пам'яттю ПЕОМ.

Зберігання даних в пам'яті

Пам'ять в ПЕОМ являє собою восьмибітну величину в байтах (байт – ціле число без знаку, ціле число зі знаком). Кожен байт має власну адресу в масиві байту пам'яті. Таке представлення називають *адресним простором*. Пам'ять має організацію, яку представлено на рисунку 1.

Адресний простір пам'яті утворюється з послідовності байтів, кожен з яких має власну адресу. Мінімальна адреса рівна 0, максимальна – $N-1$, де $N=2^n$, n – кількість адресних ліній.

За одне звернення до пам'яті процесору можуть передаватися дані розміром в байт, слово (два байти, 16 біт), подвійне слово (чотири байти, 32

біти). Процесор Intel для адресування слів і подвійних слів адресує молодшому байту – молодшу адресу, старшому – старшу (рисунок 2).

Рисунок 2 – Адресація байт слів і подвійних слів

Виходячи з рисунку 2 можна сказати, що слово за адресою 04h має значення 2411h, а подвійне слово 04h – 6111h. Будь-які два сусідні байти утворюють слово, яке може починатися з парної або непарної адреси. В першому випадку слово передається за один цикл, в другому – за два цикли.

Алгоритм роботи процесора з пам'яттю, який зображений на рисунку 1, для всіх однаковий. Процесор надсилає адресу на шину адреси за допомогою регістра, фіксує його на час запиту або зчитування і за допомогою дешифрування адреси вибирає відповідну комірку пам'яті. Далі процесор формує запит на зчитування або запис, який надходить на шину даних або записується у вибрану комірку шини даних. Процесор, який має n ліній на шині, може звертатись до $N=2^n$ комірки пам'яті, де N – кількість ліній в адресній шині. Якщо адреса починається з 0, то адресний простір дорівнює $N-1$. Така модель пам'яті називається плоскою (flat) або лінійною (безперервною) та була притаманна першим комп'ютерам.

Збільшення адресного простору стримувалось апаратними можливостями (12-розрядний регістр). Наявність шістнадцяти адресних ліній призвело до рішення, яке полягало в тому, що завдяки **вільним 4-адресним 4 адресним лініям був збільшений в 16 разів.**

В кожному блоці здійснюється лінійна адресація, яка поступає на всі блоки. Але активним є блок, який вибирається за допомогою регістра та

дешифратора, формуючи потрібний символ chipselect. Це привело до того, що були створені модулі пам'яті і адреса комірки визначалась логічно. Вона складалась із початкової адреси модуля (базова адреса), до якої додавалось зміщення (offset). Починаючи з процесора 8086 це направлення отримало розвиток сегментованої моделі пам'яті, в якому для програміста адресний простір поділений на блоки сумісних адрес, які називали *сегментами*. Програміст може звертатись до змісту комірок пам'яті в сегменті. В сегменті лінійна адресація – це загальна адреса комірки пам'яті, що визначаються як сума базової адреси та зміщення в межах сегменту. Така адресація має назву *логічна (віртуальна)*. Вона дозволяє спрощування в мультифункціональних задачах і збільшувати швидкодію процесора за рахунок поділення на сегмент даних та сегмент коду.

Тема 2. Сегментація пам'яті

Сегмент – це блок сумісних комірок пам'яті. У процесорі 8086 максимальний розмір сегменту становить 64 Кб і початковою (базовою) адресою, яка знаходиться на межі комірки пам'яті, кратній 16 байтів (*параграф*). **Таке розташування адреси, кратній 16, завжди закінчується на 0, який можна не писати, а при визначенні адреси зсув адреси вліво на 4.** Поточна адреса міститься у регістрах DS (базова адреса) та SI (зміщення). Загальна адреса по комірці MOV AX, [SI] становить зміст регістру DS*16+SI.

Проста сегментація пам'яті в процесорі 8086 має такі особливості:

1. Сегменти, які визначаються тільки сегментними регістрами, мають всього два атрибути – *початкова адреса* і *максимальний розмір сегменту* 64 Кб. Ніяких апаратних засобів контролю правильності використання сегментів немає.

2. Розміщення сегментів у пам'яті (початкова адреса) довільне: сегменти можуть частково або повністю перекриватись або не мати спільних частин у просторі пам'яті комп'ютера.

3. Програми можуть звертатись до будь-якого сегменту як для читання, так і для запису. В загальному випадку програма може звертатись за будь-якою фізичною адресою, а для визначення області пам'яті потрібні зовнішні схеми.

4. Відсутні жодні перешкоди для звертання до фізично неіснуючої пам'яті (початкова адреса знаходиться на верхній межі адресного простору). Коли програма видає адреси неіснуючої пам'яті, результати роботи непередбачувані.

З урахуванням перерахованих недоліків сегментація пам'яті в процесорі x86 була суттєво вдосконалена: всю інформацію, яку визначає сегмент, неможливо представити в одному 16-бітному регістрі, тому кожен сегмент почав визначатись *дескриптором* сегмента.

В опис сегмента входить:

- базова (початкова) адреса;
- кінцева адреса сегмента (межа, границя);
- тип сегмента;
- рівень привілеїв;
- інформація про стан.

Кількість дескрипторів у системі не обмежується. Якщо не описати область адресного простору пам'яті дескриптора, відповідний діапазон адрес стає недоступним і процесор відмовляється звертатись до нього. Функціональна схема адресного простору за допомогою дескрипторів використовується у захищеному режимі. На рисунку відображена логічна схема утворення адресного простору у захищеному режимі.

Рисунок – Логічна схема утворення адресного простору у захищеному режимі

Формат дескриптора сегмента

Базова адреса – 32-бітне поле базової адреси, займають байти 2, 3, 4, 7 дескриптора. Визначає початкову адресу сегмента у лінійному адресному просторі 4 Гб. Саме цю адресу формує процесор при заданні базової адреси.

Кінцева адреса – 20-бітне поле, яке ще зветься *межею сегмента*, займає байти 0 та 1, а також молодші 4 біти байту дескриптора.

Розмір сегмента задає спеціальний біт G (глянулярності); якщо $G = 0$, то відсутня сторінка і розмір рівний 1Мб (2^{20}), якщо $G=1$ – сторінкова гранулярність, розмір – 4Гб.

В процесорі x286 байтна гранулярність. Для отримання його кінцевої адреси (в байтах) необхідно приписати з правої сторони 3FF. Формат байтів, які реалізують вимоги для роботи з сегментом, міститься в байті 5 дескриптора сегмента – прав доступа AR.

Значення полів байту AR наступні.

Bit P (present) встановлений у 1, коли описаний сегмент присутній у пам'яті. Коли система скидає сегмент на диск, то вона встановлює $P=0$. Для того, щоб скопіювати зміст диска у пам'ять, необхідно встановити $P=1$ і здійснити restart. Цей процес зветься свопінгом або підвантаженням.

DPL (Descriptor privilege level) є двухбітним полем привілеїв дескриптора, яке визначає рівень привілеїв тієї комірки пам'яті, яку описує сегмент. Використовуються *три рівня*: 0, 1, 2. При цьому рівень 0 є найвищим.

S (System) є системним бітом, який встановлений в 1, якщо конкретне призначення сегмента описується полем типу. Якщо $S=0$, дескриптор описує системний об'єкт, який може бути сегментом пам'яті.

Трьохбітне поле *Type* (тип) визначає цільове використання сегменту і задає допустимі у сегменті операції:

- 000B – сегмент даних, дозволяється тільки зчитування;
- 001B – сегмент даних, дозволяється зчитування і запис;
- 010B – сегмент стека, дозволяється тільки читання;
- 011B – сегмент стека, дозволяється читання і запис;
- 100B – сегмент коду, дозволяється тільки виконання;
- 101B – сегмент коду, дозволяється виконання і зчитування;
- 110B – підлеглий сегмент коду, дозволяється тільки виконання;
- 111B – підлеглий сегмент коду, дозволяється виконання і зчитування.

Поле типу *Type* визначає правила доступу до сегментів. Наприклад, в регістр CS неможливо завантажити сектори сегментів з типом 000B, 001B, 010B. Жодна програма не може модифікувати сегмент, для якого дозволяється тільки зчитування. Сегменти коду, яким дозволено тільки виконуватися, неможливо зчитувати (використовується для несанкціонованого копіювання). Спроба порушити будь-яке призначення викликає особистий спосіб захисту.

Біт доступу *A* автоматично встановлений в 1, коли процесор викликає сегмент пам'яті.

Тема 3. Дескриптори таблиці. Селектори сегментів

Зміст сегменту регістру зветься *селектором*. Його довжина – 16 біт. Якщо в процесорі 8086 вміст CS вказує на базову адресу сегмента, то, починаючи з $x286$, визначає дескрипторну таблицю, за якою визначається відповідний дескриптор.

Формат селектора

Якщо $T_i=0$, то це вказує на локальну дескрипторну таблицю; якщо $T_i=1$, то на глобальну дескрипторну таблицю. RPL – рівень пріоритетів. Кількість дескрипторів сегментів практично не обмежено.

Є три види дескрипторних таблиць.

Глобальна дескрипторна таблиця (GDT) визначає загальні сегменти для багатьох задач. Для локалізації розміщення таблиці GDT призначений спеціальний регістр GDTR, в якому знаходиться 32-бітне лінійне поле базової адреси і 16-бітне поле з байтною грануляцією.

Рисунок – Формат регістра GDTR

Локальна дискрипторна таблиця (LDT) використовується при реалізації мультизадачних програм. Вона визначає сегменти, які доступні для рішення конкретних задач. Для визначення таблиці LDT є спеціальний регістр LDTR.

Сторінкова організація пам'яті

Сторінкова організація пам'яті в захищеному (віртуальному) режимі підтримує сторінкове перетворення адреси. Воно пов'язане з необхідністю збільшення основної пам'яті за рахунок віртуальної пам'яті.

Базовим об'єктом є 4Кб, який зветься *сторінкою*. При сторінковій організації пам'яті процес відбувається наступним чином: молодші 12 біт лінійної адреси 011 визначають байт в сторінці (зміщення). Наступні 20 біт адреси 12-32 сторінок по 4 Кб. Таким чином, адресуємий простір рівний $1\text{ Мб} \cdot 4\text{Кб} = 4\text{Гб}$. При такому визначенні адресного простору необхідно використовувати 1Гб пам'яті для адреси сторінок.

Лінійна адреса

Основою сторінкового перетворення виступає системний регістр керування CR2, який зветься регістром базової адреси каталога томів сторінок. Таблиця томів сторінок першого рівня містить 1Кб томів і зветься PDE. Кожен том містить в собі 1Кб 32-бітних дескрипторів, які зветься сегментами таблиці сторінок PTE і кожен сегмент PTE адресує сторінковий кадр у фізичну пам'ять, в якому за допомогою зміщення визначається потрібний байт.

Вимоги до програмування задач

Програмування при сегментації пам'яті, поділеної на сегментацію, без сторінкового перетворення потребує тільки GDT. Для фіксованої кількості прикладних програм можливо заздалегідь фіксувати кількість сегментів і таблиць LDT (промислові контролери).

На відміну від сегментного, сторінковим перетворенням керує біт режиму PG у системному регістрі CR0, Якщо PG=0, сторінкове перетворення не діє. Для встановлення PG в 1 необхідно задовільнити наступні вимоги:

1. Створити як мінімум дві таблиці: каталог таблиці сторінок і таблицю сторінок (PDE і PTE).
2. В базовий регістр каталога сторінок завантажити фізичну адресу каталога сторінок.
3. Біт PG і PE (захищений режим) необхідно встановити в 1 одночасно.

Структура кеш-пам'яті

Кеш-пам'ять представляє собою 4-направлену асоціативну по множині пам'ять. Коли до асоціативної пам'яті поступає адреса, вона зрівнюється з додатковими для кожної одиниці ознаками (тегами). Коли один із тегів (21 біт) зрівнюється з адресою, подальше визначення одиниці пам'яті відбувається за допомогою дій, які асоційовані з даним тегом. Це – поле індексу і вибір байту.

Кеш-пам'ять складається з трьох блоків: *блока даних*, *блока тегів*, *блока достовірності і LPU*.

Блок даних містить до 8 Кб даних і команд, поділений на 4 масива (0, 1, 2, 3), кожен з яких утворюються із 128-ми рядків. Рядок містить дані із 16-ти послідовних адрес пам'яті, починаючи з адреси, кратній 16. Індекс масиву блока даних складається із семи біт, відповідає чотирьом рядкам кеш-пам'яті, які називаються *множиною*.

Блок тегів містить 4 масива по 128 тегів. Тег складається із 21-го біта фізичної адреси даних, які знаходяться в асоціативному рядку кеш-пам'яті.

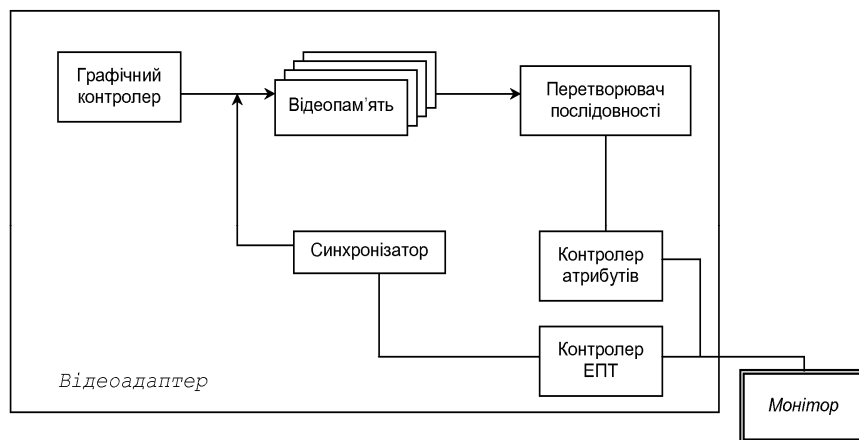
МОДУЛЬ II

Відеоадаптери

Тема 1. Архітектура відеоадаптерів

Відеоадаптери умовно поділяють на шість логічних блоків:

- графічний контролер;
- відеопам'ять;
- синхронізатор;
- контролер атрибутів;
- контролер променевої трубки;
- перетворювач послідовності (знакогенератор).



У *відеопам'яті* розміщуються дані, які відображаються на екрані дисплея. Її розмір – 256 Кб. Для відеоадаптерів SVGA, XVGА об'єм збільшений до кількох Мб. Відеопам'ять знаходиться в адресному просторі процесора і програми можуть безпосередньо проводити з нею обмін даними.

Графічний контролер служить для здійснення обміну між процесором і відеопам'яттю комп'ютера. Апаратура графічного контролера дозволяє здійснювати над даними пам'яті логічні операції.

Послідовний перетворювач забезпечує вибірку з відеопам'яті одного або декілька байтів, перетворення їх у послідовний потік бітів і передачу контролеру атрибутів.

Контролер атрибутів перетворює інформацію про колір з формату відеопам'яті у формат ЕПТ. Перетворення кольорів здійснюється відповідно до таблиці кольорів палітри. *Атрибути* визначають колір символу і колір фону. Завдяки такому режиму досягається значна економія пам'яті в порівнянні з графічним режимом.

Контролер ЕПТ виробляє часові синхросигнали, які керують ЕПТ.

Знакогенератор. При відображенні символа на екрані відбувається перетворення з ASCII в вдомірний масив пікселів. Для цього перетворення використовується таблиця трансляції символа (*знакогенератор*). У відеоадаптерах MDA (монохромний), CGA і "Геркулес" таблиці знакогенератора знаходяться в ПЗУ. Програміст не має можливості змінити та прочитати ці таблиці без спеціальних команд. На інших відеоадаптерах, крім вказаних, таблиці знакогенератора зчитуються з ПЗУ і записуються в ОЗУ, що дає можливість програмісту використовувати власні таблиці знакогенератора і відображати на екрані монітора будь-які символи.

Типи дисплеїв

Спочатку комп'ютери IBM PC комплектувалися монохромними дисплеями MD. Цей адаптер мав невелику роздільну здатність. Через деякий час компанія «Геркулес» випустила монохромний адаптер, який мав можливість виведення графіки та велику роздільну здатність.

Першим кольоровим відеоадаптером став CGET. Він забезпечував можливість відображати текстову і графічну інформацію, але мав низьку

дозволяючи здатність порівняно з «Геркулесом». Пізніше фірма IBM випустила відеоадаптери VGA і EGA, які мали значно кращі роздільні здібності, ніж CGA.

В останній час різні фірми випустили адаптери SVGA, які поки не мають єдиного стандарту.

Фірма IBM почала випуск нового відеоадаптера Extended (XGA), який стане новим стандартом. Він містить графічний відеоадаптер, який збільшує його швидкість роботи та можливості. При обміні даними сам адаптер замість ЦП реалізує керування шиною даних, що дозволяє швидко передавати зображення на екран. В залежності від використаного інтерфейсу дисплеї поділяються на:

- *компаративний* (кольоровий монохромний). Має одну аналогову вхідну лінію стандарту NTSC, який використовується у телебаченні. Композитний дисплей використовують з адаптером CGA.

- *цифровий дисплей*. Має від одного до шести ліній. Може відображати до 2^n різних кольорів, де n – кількість вхідних ліній. Може використовуватись з V-2, VGA, CGA.

- *аналоговий RGB-дисплей*. Має три аналогові вхідні лінії, які керують червоним, зеленим та синім кольорами. Рівень напруги на кожній лінії відповідає інтенсивності кольору на екрані. Кількість кольорів, які саме відображає аналоговий дисплей, обмежена тільки можливостями відеоадаптера аналогового дисплея. Використовується з VGA, SVGA, XGA.

- *РК-монітори*. Австрійський ботанік Рейницер в кінці XIX століття відкрив залежність органічних речовин від температури. Пізніше фізик Легман звернув увагу на здібність рідких кристалів змінювати свої відображаючі властивості в залежності від температури (хамелеони). Пізніше в 70-х роках це відкриття знайшло практичне впровадження – винайдено рідкокристалічні монітори.

Тема 2. Відеопам'ять комп'ютера

У відеопам'яті комп'ютера розміщені дані, які відображаються адаптером на екрані дисплея. Відеопам'ять знаходиться у адресному просторі процесора і програми можуть безпосередньо проводити з нею обмін даними. Фізично відеопам'ять поділена на *чотири банки* (по байту в кожному шарі). Якщо ці шари використовуються для читання або запису визначені програмістом за допомогою встановлення деяких регістрів. Ці шари розміщені в одному адресному просторі таким чином, що за кожною адресою розміщується по чотири байти, по байту в кожному шарі. Деякі операції заповнення екрану здійснюються дуже швидко. Є можливість записати тільки в один шар за допомогою встановлення регістра дозволу запису кольорового слоя. Для операцій читання у кожний момент часу може бути дозволений тільки один слой за допомогою регістра.

Більшість режимів відеопам'яті також розділені на декілька сторінок, з яких одна є *активною* і відображається на екрані. Програміст може перемикає активні сторінки відеопам'яті.

Текстовий режим

У текстовому режимі на екрані відображаються тільки символи, а також символи псевдографіки. Стандартні текстові режими роботи відеоадаптера дозволяють видавати на екран 25 рядків по 40 або 80 символів у кожному. Якщо перепрограмувати регістри відеопам'яті, то можна збільшити кількість рядків, які відображатимуться на екрані.

Для кодування кожного знакомісця екрану (символа) використовують два байти: *перший байт* містить ASCII-код відображеного символу, *другий байт* – атрибути символу (колір символу, колір фону).

Символ	Атрибут	Символ	Атрибут
Знакомісце 0		Знакомісце 1	

При відображенні символу на екрані відбувається цого перетворення його з ASCII в вдомірний масив пікселів (таблиця трансляції символів є таблицею

знакогенератора, яка зберігається у другому шарі відеопам'яті). Байти сивола знаходяться у парних адресах, а байти атрибутів – у непарних.

При встановленні текстового режиму роботи відеоадаптера завантажується знакогенератор в другий шар, а потім символи відображуються на екрані. Завдяки цьому можлива заміна стандартної таблиці знакогенератора на власну. Ця можливість використовується для завантаження потрібного алфавіту або нестандартних символів.

Відеоадаптер забезпечує можливість одночасно завантажувати до восьми таблиць знакогенератора. Кожна таблиця містить 256 символів. Одночасно активними можуть бути одна або дві таблиці знакогенератора, що дає можливість одночасного відображення на екрані до 512 символів. При цьому перший біт байту атрибутів вказує, яка із завантажених таблиць використовується при відображенні символу. Номера активних таблиць знакогенератора визначає регістр знакогенератора.

EGA підтримує наступні розміри матриці символа: 8x8 (стандартний), 8x14 (поліпшений), VGA – 9x16. Опис кожного символу містить 8 байт.

Програмування контролера дисплея 6845

Контролер 6845 має вісімнадцять керуючих регістрів (0 - 17). Перші десять регістрів фіксують горизонтальні і вертикальні параметри дисплея. Вони встановлюються BIOS під час зміни режиму екрана. Регістрові порти 10-11, 14-15 встановлюють форму і місце розташування курсора. Регістрова пара 12-16 керує сторінками дисплея, 16-17 – визначає позицію світлового пера. Доступ до регістрів здійснюється за допомогою двох портів: 3B4 і 3B5 (монохромний), 3D4 і 3D5 (кольоровий) адаптери.

Методика програмування складається з наступних дій.

Для запису інформації необхідно записати у порт 3B4 (3D4) номер регістра відеоадаптера, а потім через порт 3B5 (3D5) – необхідний байт. Аналогічний алгоритм для зчитування інформації.

Оскільки регістри використовуються попарно, необхідно спочатку визначити номер регістру для молодшого байту і записати молодший байт, потім знову вказати номер регістру для старшого байту і записати старший байт.

Приклад. Записати в 10-11 слово, яке записане в CX (3515h).

```
mov dx, 3d4h
mov al, 10h
out dx, al
inc dx
mov al, cl
out dx, al
dec dx
mov al, 11h
out ax, al
inc dx
mov al, ch
out dx, al
```

Засоби BIOS для роботи з відеоадаптером

BIOS здійснює роботу з відеоадаптером через переривання *int 10h*. Це переривання має функції, які вирішують різні задачі для програмування відеоадаптерів.

Функції int 10h

- 00h – вибір режиму роботи;
- 01h – задання форми курсора;
- 02h – задання позиції курсора;
- 03h – визначає колір і розмір курсора;
- 04h – використання світового пера;
- 05h – задання активної сторінки;

06h – прокрутка вікна вгору;
 07h – прокрутка вікна вниз;
 08h – читання символів і його атрибутів у поточній позиції курсора;
 09h – запис символу з атрибутом, який був у позиції курсора;
 0Ah – запис символу з тим атрибутом, який буз у позиції курсора;
 0Bh – встановлення кольорової палітри, режими 4, 5, 6;
 0Ch – вивід пікселя на екран;
 0Dh – читання пікселя;
 0Eh – вивід символу в режим тілтайпа;
 0Fh – визначення поточного режиму відеоадаптеру;
 10h – керування регістрами пам'яті має підфункції:
 11h – завантаження таблиці знакогенератора, має 11 підфункцій;
 12h – визначає конфігурацію і вибір програми друку екрану;
 13h – вивід текстового рядка;
 1Ah – читання/запис конфігурації відеоадаптера;
 1Bh – отримання даних про стан VGA;

Режими дисплея

№	Режим
0	40 x 25 (350 - 200) – алфавітно-цифровий монохромний
1	40 x 25 (350 - 200) – алфавітно-цифровий кольоровий
2	80 x 25 (640 - 200) – алфавітно-цифровий монохромний
3	80 x 25 (640 - 200) – алфавітно-цифровий кольоровий
4	320 x 200 – чотирьох кольорова графіка
5	320 x 200 – алфавітно – цифровий і графіка
6	640 x 200 – кольорова графіка
7	720 x 350 - алфавітно – цифровий
8	160 x 200 – 16-кольорова графіка
9	320 x 200 – 16-кольорова графіка
A	640 x 200 - чотирьох кольорова графіка

Функції BIOS

00h – вибір режиму роботи. Ця функція дозволяє змінити режим роботи відеоадаптера на вказаний в регістрі AL.

01h – задання форми курсора, Дозволяє змінити вертикальні розміри курсора, шляхом задання верхніх і нижніх границь курсора.

02h – задання позиції курсора. Якщо відеопам'ять поділена на декілька сторінок, кожна сторінка має власний курсор, координати якого можна встановити окремо.

Текстова пам'ять містить *вісім відеосторінок* та займає в адресному просторі 32 Кбайти, починаючи з сегментної адреси B800h. Починається вона з відео сторінки 0, адреса якої співпадає з адресою всієї текстової відеопам'яті. Кожна відеосторінка займає 4 Кбайти. Таким чином, відеосторінка 1 починається з сегментної адреси B900h, відеосторінка 2 – BA00h і т.д.

Важливо: під час увімкнення комп'ютера активною (видимою) відеосторінкою стає відеосторінка 0. Зміна відеосторінок здійснюється за допомогою функції 5 переривання 10h BIOS.

Функції BIOS для завантаження символів

Функція 11h, підфункція 10h – завантаження шрифту користувача. Для виклику в регістр AX записуємо 1110h, в регістр BH – символ, BL – блок генератора (0), CX – кількість символів.

МОДУЛЬ III

Основи взаємодії комп'ютера з периферійними пристроями

Тема 1. Суть і поняття периферійного пристрою

Сукупність апаратних і програмних засобів, які забезпечують обмін інформацією між ЦП та периферійним пристроєм зветься інтерфейсом.

Периферійний пристрій – це пристрій, який підключається до комп'ютера за допомогою портів телекомунікації (COM, USB), а також різноманітних модемів.

З точки зору програміста слід враховувати, що як і ПК, так і периферійний пристрій являють собою самостійні системи, які працюють незалежно. В задачу програміста входить забезпечення роботи системи в режимі обміну інформацією. Слід врахувати, що кожний пристрій одночасно не може приймати та передавати інформацію, а також слід врахувати час, необхідний для обробки інформації.

Всі ці задачі вирішує програміст за допомогою програмно доступних регістрів (регістр керування, стану і режиму). В якості прикладу наступним етапом є взаємодія ПК з принтером.

Тема 2. Матричні принтери та їх програмування

Розглянемо часову діаграму роботи матричного принтера та комп'ютера.

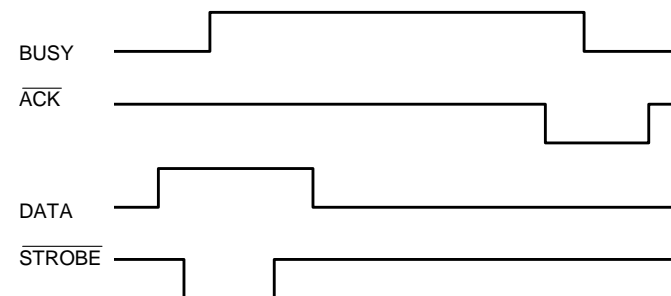


Рисунок – Часова діаграма принтерного порта

За системою команд існують два види принтерів: IBM та Epson. Найбільше розповсюдження отримали Epson FX, які мають печатні голівки з 9 або 24 голками.



Рисунок – Матричні принтери Epson FX

Підключення принтера до ПК здійснюється через COM-порт, через LPT або через послідовний адаптер USB.

ПК обмінюється даними з принтером за допомогою трьох портів: 378h (278h), 37Ah, 379h.

Рисунок – Функціональна схема підключення принтера

З наведеного рисунку видно, що для того, щоб вивести символ на принтер програма спочатку повинна перевірити BUSY і рівень ACK. Після цього необхідно встановити код символу на лініях DATA 378h. Далі через 0,5 мілісекунд лінію STROBE перевести в стан 0. При цьому символ запишеться у внутрішній буфер символу. Потім необхідно лінію STROBE 0,5 мсек. утримати в стані нуля і після закінчення перевести в стан логічної 1. Після встановлення STROBE в стан 0 BUSY через 5 мсек. слід встановити в стан логічної 1.

Коли принтер обробить лінію ACK, через 5 мсек. ACK перейде в стан 0, а ще через 5мсек. – в логічну 1. Після цього принтер знову готовий прийняти символ.

Порти принтера

378h – порт призначений для запису байту даних, який виводиться на принтер. Можливе тільки читання записаного байту.

D0 – подається на другий контакт роз'єма PC;

D7 – на 9 контакт.

37Ah – порт керування принтером. Доступний як для читання, так і для запису.

379h – порт стану принтера. Доступний тільки для читання.

D2 - сигнал помилки, активний рівень - 0, контакт роз'єму - 15, ERROR;

D4 - принтер обраний, контакт 13, SLCT;

D5 - кінець паперу, контакт 12, PE;

D6 - готовність принтера, активний рівень - 0, контакт роз'єму - 10, ACK;

D7 - 0 - принтер зайнятий, знаходиться в стані offline або сталася помилка, контакт 11, BUSY.

Засоби BIOS для роботи з принтером

BIOS використовує для роботи з принтером функції 0, 1, 2 переривання INT 17h.

Функція 00h призначена для друку одного символу.

На вході: AH = 00h;

AL = ASCII -код символу для друку;

DX = номер принтера: 0, 1 або 2.

На виході: AH = слово стану принтера.

Ця функція виводить на принтер один символ, заданий в регістрі AL. У регістр DX необхідно записати номер принтера, який використовується (для LPT1 – 0, для LPT2 - 1). Після виконання переривання регістр AH буде містити *слово стану*, що має наступний формат:

D0 - таймаут, занадто велика затримка при виконанні операції друку, можливо, оо принтер несправний;

D1-D2 - не використовуються;
D3 - помилка введення / виводу;
D4 - 1 – принтер online; 0- принтер в стані offline;
D5 - закінчення паперу;
D6 – підтвердження Ask;
D7 - 1 - принтер готовий, 0 - принтер зайнятий.

Викликавши функцію 0 переривання INT 17h, програма повинна перевірити окремі біти слова стану і переконатися в тому, що виведення байту відбулось без помилок. Якщо принтер несправний, програма повинна надати оператору можливість скасувати друк.

Область даних BIOS за адресою 0000h:0478h містить чотири байти, які використовуються як лічильників часу при очікуванні готовності принтера.

Переривання INT 17h має ще дві функції:

Функція 01h ініціалізує принтер.

На вході: AH = 01h;

DX = номер принтера: 0, 1 або 2.

На виході: AH = слово стану принтера.

Ця функція виконує апаратне скидання принтера.

Слово стану принтера може бути отримано за допомогою *функції 02h*.

На вході: AH = 02h;

DX = номер принтера: 0, 1 або 2.

На виході: AH = слово стану принтера.

Програмування символів і режимів принтера

Програмування символів принтера і зміни його режимів роботи здійснюються за допомогою командної послідовності, ознакою якої є байт ESC-послідовність з ознакою 1Bh. Слідом за цим байтом програма посилає в принтер саму командну послідовність. Довжина послідовності залежить від виконуваної команди. Деяким командам байт ESC не потрібен. Наприклад, 07h

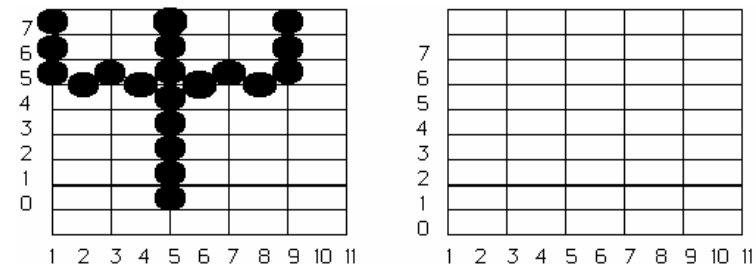
– генерація звукового сигналу, 0Dh – повернення каретки, 0Ah – переведення рядка, 0Ch – перевод сторінки.

Для задання якості друку потрібно вивести на принтер три байти:

ESC " x " n – вибір якості друку: 0 - низька якість; 1 – висока якість.

Друкування власних символів

Для друкування власних символів використовується сітка. У 9-голчастих принтерах ця сітка має 11 стовпців і дев'ять рядків:



З дев'яти рядків може використовуватися тільки вісім верхніх або вісім нижніх.

Для *перевизначення символу* 9-голчастий принтер Epson використовує команду *ESC "&"*, яка має наступний формат:

ESC "&" "0" n1 n2 a1 d1 d2... dn

Параметри n1 і n2 задають діапазон кодів ASCII символів. Якщо визначається один символ, ці два параметри повинні бути однаковими. Параметр a1 визначає ширину символу в точках і його положення в сітці (верхні вісім ліній, або нижні вісім ліній). Ширина визначеного символу потрібно для друку. Старший біт параметра a1 задає розміщення символу в сітці. Якщо цей біт дорівнює 1, використовуються вісім верхніх ліній сітки, якщо 0 – вісім нижніх.

Молодші сім бітів задають ширину символу і є число, яке визначається наступним чином:

- ✓ для ширини символу береться число 8;
- ✓ для кожного порожнього стовпця в сітці з правого боку символу треба відняти з початкового значення одиницю;
- ✓ для кожного порожнього стовпця в сітці з лівого боку символу треба додати до початкового значення число 16.

Приклад.

$a1 = 8$ (початкове значення) – 2 (два порожніх стовпця праворуч) + 32 (два порожніх стовпця ліворуч) + 128 (старший біт дорівнює 1) = 166.

Параметри d1... dn – зразки стовпців точок визначення символу. Їх має бути завжди одинадцять, навіть якщо символ містить порожні стовпці.

Для увімкнення визначеного програмою символу *ESC "% " 0 "*, для використання із внутрішнього ПЗУ принтера – *ESC "% " 1 "*.

Формат команди для визначення символів у принтері Epson LQ-2550:

ESC "&" "0" n1 n2 d0 d1 d2 data

Параметри n1 і n2 задають діапазон кодів ASCII символів, d0 задає кількість вільних стовпців зліва від символу, d2 визначає кількість вільних стовпців праворуч від символу. Параметр d1 визначає ширину символу.

Друкування у графічному режимі

Для переводу принтера у графічний режим використовується команда:

ESC "" m n1 n2 data*

m – режим друку;

n1, n2 – визначають довжину графічного рядка у точках.

При визначенні довжини графічного рядка необхідно враховувати, що в режимі одинарної щільності на рядку довжиною 8” можна розмістити 480 точок, в режимі чотирьохнавної щільності – близько 2000.

Ділимо довжину рядка на 256, цілу частину результату ділення використовуємо як параметр n2, залишок від ділення – як n1.

Приклад.

Треба роздрукувати рядок з 1234 точок. Тоді параметр n2 буде дорівнювати $1234/256 = 4$; залишок від ділення складе $1234 - 256 * 4 = 210$. Це і є параметр n1.

Команда повинна завжди містити два параметри, навіть якщо параметр n2 вийшов рівним нулю. Для представлення одного восьмиточкового стовпця графічного рядка використовується один байт даних, причому верхній точці в стовпці відповідає старший розряд байта, а нижньої – молодший:

7	10011011b або 9Bh
o	6
o	5
*	4
*	3
o	2
*	1
*	0

Засоби ОС для роботи з принтером

Для друку символу принтері використовують *функцію 05h* переривання INT 21h.

На вході:

AH = 05h;

DL = ASCII-код символу для друку.

На виході:

АН = слово стану принтера.

Типова послідовність для вивода одного символу друку:

1. Вивести передаваний байт у регістр даних;
2. В циклі перевірити стан принтера до установки біта D₇ регістра стану (який використовує timeout);
3. Перевірити біти 3-5 регістра стану на наявність помилки;
4. Встановити і одразу ж скинути нульовий біт регістра керування. Для цього використовують наступну програму:

```
mov dx, 37Ah
mov dl, 00001101b
out dx, al
xor al, 1
out dx, al
```

Після цього можна прочитати регістр стану і перейти до друку іншого символу.

Сучасні принтери являють собою складні пристрої, які мають свої МП, ОЗУ, ПЗУ. МП використовують для обробки керуючих послідовностей і керування друком. ПЗУ містить опис символів в залежності від їх коду. ОЗУ використовується для тимчасового зберігання введених даних і запрограмованих користувачем символів. Команди починаються з символу ESC і мають довжину від 2-х байт.

Тема 2. Лазерні принтери

Лазерний принтер є одним з видів принтерів, що дозволяє швидко виготовляти високоякісні відбитки тексту й графіки на папері. Формування

зображення відбувається шляхом безпосередньої експозиції (освітлення) лазерним променем фоточутливих елементів принтера.

Функціональну схему лазерного принтера наведено на рисунку 1.

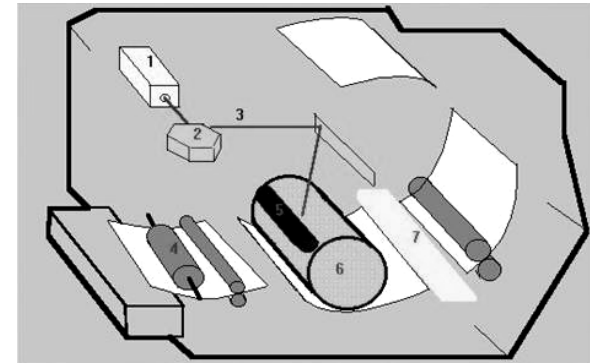


Рисунок 1 – Функціональна схема лазерного принтера:

- 1 – генератор лазера,
- 2 – дзеркало, що обертається,
- 3 – лазерний промінь,
- 4 – валики подачі паперу,
- 5 – валик подачі тонера,
- 6 – фотопровідний циліндр,
- 7 – вузол фіксації зображення.

Головним елементом лазерного принтера є *фотопровідний циліндр* (фотобарабан) – металевий циліндр, вкритий тонкою плівкою напівпровідника (як правило, оксид цинку). Його поверхні можна надати позитивний або негативний заряд, який зберігається на поверхні, поки барабан не освітлено.

Малогабаритний лазер генерує тонкий світловий промінь, який відбивається від 6-гранного дзеркала на фотобарабан та розряджає таким чином його позитивно заряджену поверхню. Щоб отримати зображення лазер вмикається/вимикається керуючим мікроконтролером, а дзеркало, що обертається, віддзеркалює промінь в рядок на поверхні барабана. Таким чином формується рядок зображення із заряджених/розряджених ділянок барабану.

Барабан прокручується повз валик з тонером. Тонер, заряджений позитивно, прилипає виключно до нейтральних (розряджених) ділянок поверхні фотобарабану і, відповідно, відштовхується від позитивно заряджених.

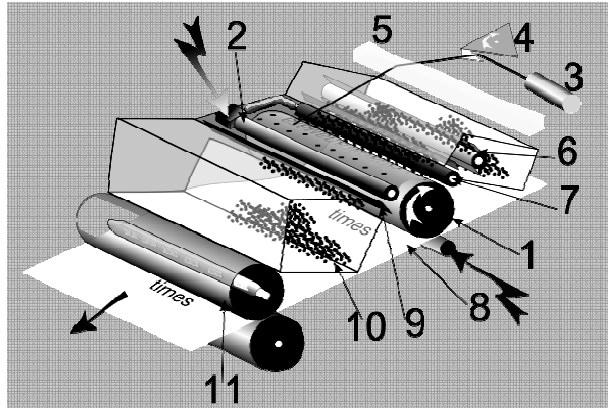


Рисунок 2 – Принцип роботи лазерного принтера:

- 1 – фотобарабан,
- 2 – зарядний валик,
- 3 – промінь лазера,
- 4 – обертове дзеркало,
- 5 – розподільна лінза,
- 6 – катридж з тонером,
- 7 – магнітний вал,
- 8 – папір,
- 9 – рапель,
- 10 – бункер з відпрацьованим тонером,
- 11 – нагрівальний вузол фіксації зображення, який припаює тонер до паперу.

За допомогою коронувального дроту паперові надається статичний заряд. Проходячи повз фотобарабан заряди різної полярності, накопичені на поверхні

паперу і фотобарабану, спричиняють притягування тонеру на поверхню паперу. Після цього сформоване зображення з тонеру фіксується за допомогою нагрівального елемента. Тонер як правило, є полімером або смолою, тому він легко плавиться.

Тема 4. Струменевий принтер

Тема 5. Тривимірний принтер (3D-принтер)

Тривимірний або 3D-принтер – пристрій, що використовує метод пошарового створення фізичного об'єкта за цифровою 3D-моделлю (рисунок 1). Тобто це пристрій виведення тривимірних даних, результатом якого є деякий фізичний об'єкт. Він використовує метод пошарового створення фізичного об'єкта на основі віртуальної 3D-моделі.

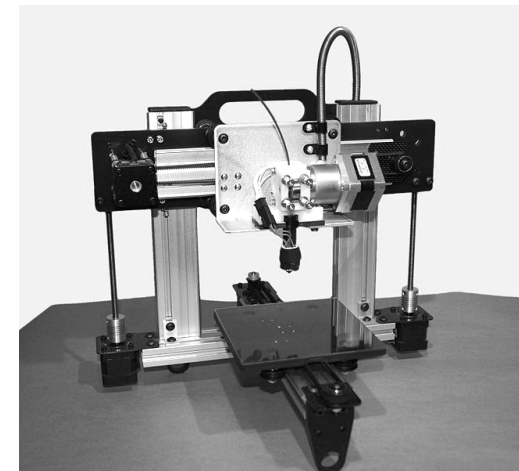


Рисунок 1 – Тривимірний (3D) принтер

Спочатку на комп'ютері створюється 3D-модель об'єкта, яка надсилається на 3D-друк. В результаті створюється фізичний об'єкт за моделлю в її натуральному вигляді.

Принцип роботи 3D-принтерів схожий з роботою струменевих принтерів. Основна відмінність полягає у тому, що замість нанесення чорнила з друкувальної голівки на аркуш паперу, в 3D-принтері сполучна речовина через друкувальну голівку наноситься на тонкий (близько 0,1 мм) шар порошку, створюючи один перетин об'єкта. У тих місцях, де нанесено сполучну речовину, порошок твердне. Наступний перетин “склеюється” з попереднім і так далі, поки не буде сформовано повністю твердий об'єкт. Після закінчення роботи 3D-принтера виріб витягується з маси порошку.

На рисунку 2 наведено приклад друку об'єктів на 3D-принтері.



Рисунок 2 – Приклади друку об'єктів за допомогою 3D-принтера

У той же час розрізняють лазерну та струменеву технології 3D-друку. До лазерних належать наступні.

Лазерний друк полягає у тому, що ультрафіолетовий лазер поступово, піксель за пікселем, засвічує рідкий фотополімер, або фотополімер засвічується ультрафіолетовою лампою через фотошаблон. Після цього він твердне та перетворюється в твердий пластик.

Лазерне запікання полягає у тому, що лазер випікає у порошок з легкоплавного пластику шар за шаром контур майбутньої моделі. Після цього зайвий порошок струшується з готової моделі.

Ламінування полягає у тому, що деталь створюється з великої кількості шарів робочого матеріалу, які поступово накладаються один на один і склеюються. При цьому лазер вирізає в кожному контурі перерізу майбутньої деталі.

До струменевих технологій належать наступні.

Застигання матеріалу під час охолодження – роздавальна голівка видавлює на охолоджувальну платформу краплі розігрітого термопластика. Краплі швидко застигають і злипаються один з одним, формуючи шари майбутнього об'єкта.

Полімеризація фотополімерного пластику під дією ультрафіолетової лампи – спосіб схожий на попередній, але пластик твердне під дією ультрафіолету (рисунк 3).

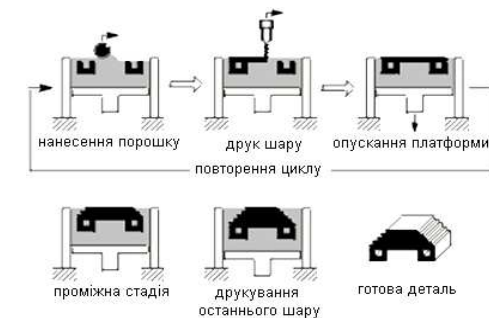


Рисунок 3 – Принцип струменевої полімеризації

Склеювання (спікання) порошкоподібного матеріалу – те ж, що і лазерне спікання, лише порошок склеюється клеєм, який надходить із спеціальної струменевої голівки. При цьому можливо відтворити забарвлення деталі, використовуючи сполучні речовини різних кольорів.

Тема 6. Контролер клавіатури

Контролер клавіатури призначений для управління роботою комп'ютера і функціонально складається безпосередньо із самої клавіатури, яка містить в собі мікропроцесор, який виконує функції контролера клавіатури по визначенню скан-коду (номера натиснутої клавіші) і подальшої передачі скан-коду в ЕОМ, яка також містить мікропроцесор (системний контролер клавіатури).

Контролер 8042 клавіатури з'єднаний 4-проводним заекранованим кабелем (+5V, земля, лінія даних, лінія синхронізації).

Контролер клавіатури забезпечує синхросигнал для передачі даних у кожному напрямі. Контролер клавіатури передає і приймає дані в 11-розрядному форматі.

Контролер клавіатури автоматично визначає тип клавіатури за форматом даних: перший розряд є стартовим, потім вісім інформаційних розрядів, розряд паритету і стоповий. Посилка даних синхронізується клавіатурою. Контролер клавіатури блокує інтерфейс, поки система не отримує байт. Якщо у байті наявна помилка, то у контролера клавіатури автоматично відправляє RESEND (протокол обміну).

Управління клавіатурою можливо за допомогою портів 60h, 61h, 64h.

61h - можна дозволити або заборонити роботу з клавіатурою. Якщо D₇=1 порту 61h – то клавіатуру заблоковано, якщо D₇=0, то клавіатура розблокована. Оскільки порт 61h використовується для роботи з іншими пристроями, то після

закінчення роботи з клавіатурою необхідно відновити статус цього порту. Таким чином при роботі з клавіатурою необхідно дотримуватись наступних правил.

```
in al, 61h
mov ah, al
or al, 80h
out 61h, al
xchg ah, al
out al, 61h
mov al, 20h
out 20h
```

Порт 60h призначений для читання даних з клавіатури. Порт 64h дозволяє читати дані з клавіатури, визначення статусу, програмувати і здійснювати налаштування клавіатури.

Робота клавіатури здійснюється за допомогою регістра керування і регістра стану контролера.

Формат регістра стану контролера системного блоку клавіатури:

D₀ – 0 у буфері відсутні дані; - 1- є дані (порт 60h або 64h);

D₁ – 0 вхідний буфер пустий; - 1- вхідний буфер буфер;

D₂ – 0 - скид по включенню живлення; 1 - програмний скид;

D₃ – 0 у вихідному буфері дані; - 1- у вихідному буфері команда;

D₄ – 0 клавіатура заблокована; - 1- розблокована;

D₅ – тайм аут передачі: 0 – передача даних незавершена; - 1- завершена;

D₆ – тайм аут прийому: 0 – прийом не закінчився за 2 мілісек; - 1- закінчився;

D₇ – помилка паритету: 0 – останній байт даних отриманий від клавіатури з непарним паритетом; - 1 – з парним.

Крім того використовується комбінації згідно таблиці:

D7	D6	D5	Дія
0	0	1	Немає синхроімпульсів
0	1	1	Є синхроімпульси, немає відгуку
1	0	1	Помилка паритету

Формат команди управління:

D₀ – дозвіл переривання по заповненню вихідного буфера: 0 – забороняє;

- 1 – дозволяє;

D₁ – повинен дорівнювати 0;

D₂ – системний флажок, значення записане у цей розряд записується у відповідний регістр стану;

D₃ – відміна блокування клавіатури: 0 – захисний замок діє, 1 – захисний замок ігнорується;

D₄ – блокування клавіатури: 0 – дозволити інтерфейс клавіатури, 1 – заборонити;

D₅ – режим IBM PC: 0 – використовувати 11-розрядні коди клавіатури, 1 – використовувати коди клавіатури PC/XT;

D₆ – режим сумісності PC AT та PC XT двохбайтовий код віджиму перетворюється в однобайтний: 0 – не перетворювати код, - 1 перетворювати код;

Для зв'язку контролера 8042 використовуються наступні порти:

8-розрядний вихідний порт для управління системними функціями і обміну з клавіатурою (порт 2);

8-розрядний вхідний порт читання конфігурації системи BIOS (порт 1);

2-розрядний контрольний порт читання даних (Test 1, Test 2).

Буфер клавіатури має довжину 32 байти і розміщується за адресою 0000:041Eh.

Буфер клавіатури FIFO має довжину 32 байти і розміщується за адресою 0000:041Eh для IBM PC/XT. У IBM AT ці комірки пам'яті містять значення 00Eh і 003Eh. Так як зміщення задається відносно адреси 0040h то звичайно розміщення буфера AT і PC 2 відповідають його розміщення у PC/XT.

Буфер клавіатури організовано циклічно. При його організуванні самі старі значення будуть загублені. Дві комірки пам'яті в області BIOS 0000:04Ah

і 0000:041Ch містять початок і кінець буфера. Якщо значення ці дорівнюють одне одному, то буфер пустий.

Комп'ютери типу IBM PC AT дозволяють керувати швидкісними характеристиками клавіатури, а також запалювати або гасити діоди на панелі клавіатури. Для розширеного керування клавіатурою використовують порт 60h у режимі запису. Для керування мікропроцесором клавіатури використовують порт 64h.

При використанні порту 60h на запис програма додатково отримує наступні можливості:

- о встановлення часу очікування перед переходом клавіатури в режим автоповтору;
- о встановлення періоду генерації скан-коду в режимі автоповтора;
- о керування світлодіодами, розташованими на лицьовій панелі клавіатури - Scroll Lock, Num Lock, Caps Lock.

Для посилки команди процесору 8042 спочатку необхідно переконатися в тому, що його внутрішня черга команд порожня. Це можна зробити, прочитавши слово стану 8042 з порту з адресою 64h біт D1=0. Після того, як програма дочекається готовності мікропроцесора, вона може послати йому команду, записавши її у порт.

Формати команд процесора 8042: команда затримки періоду автоповтору, команда керування світлодіодами.

Для встановлення автоповтору 0F3h, байт, який визначає характеристики режиму.

Біти	Значення	Примітка
0-4	Період автоповтора 0 - 30.0; 0Ah - 10.0; 1 - 26.7; 0Dh - 9.2; 2 - 24.0; 10h - 7.5; 4 - 20.0;	визначає кількість посилок скан-коду, що генеруються процесором клавіатури в одну секунду.

	14h - 5.0; 8 - 15.0; 1Fh - 2.0.	
5-6	Затримка включення режиму автоповтора 00 - 250 мс; 01 - 500 мс; 10 - 750 мс; 11 - 1000 мс.	
7	Зарезервовано, має дорівнювати 0	

Для управління світлодіодами, розташованими на лицьовій панелі клавіатури, використовується команда 0EDh.

Біти	Значення
0	1 - увімкнути діод Scroll Lock;
1	1 - увімкнути діод Num Lock;
2	1 - увімкнути світлодіод Caps Lock;
3-7	не використовуються (0)

Апаратне переривання клавіатури IRQ1

Цій лінії відповідає переривання. Обслуговується модулями BIOS INT 16h. Драйвера клавіатури і резидентні програми можуть використовувати INT 09h додатково. Обробник виконує наступні дії:

- ✓ читає з порту 60h скан-код натиснутої клавіші;
- ✓ записує обчислене за скан-коду значення ASCII-коду натиснутої клавіші в спеціальний буфер клавіатури, розташований в області даних BIOS;
- ✓ встановлює в 1 біт 7 порту 61h, дозволяючи подальшу роботу клавіатури;
- ✓ повертає цей біт в початковий стан;
- ✓ записує в порт 20h значення 20h для правильного завершення обробки апаратного переривання.

Засоби BIOS для роботи з клавіатурою

Набір функцій для роботи з клавіатурою з переривання INT 16h містить наступні функції:

- 00h – читання символу з очікуванням;
- 01h – перевірка буфера на наявність у ньому символу;
- 02h – отримання стану переключаємих станів;
- 03h – встановлення часових характеристик клавіатури;
- 05h – запис символів у буфер клавіатури;
- 10h – читання символу для розширеної клавіатури;
- 11h – перевірка буфера на наявність у ньому символу для розширеної клавіатури;
- 12h – отримання стану переключаємих клавіш розширеної клавіатури.

Функція 00h виконує читання символу з буфера, якщо він там є. Якщо буфер клавіатури порожній, програма переходить в стан чекання до тих пір, поки не буде натиснута наступна клавіша.

Для виклику: AH=00h, AL=ASCII, AH=ASCII-розширена клавіатура.

Команди контролера клавіатури:

Код	Функція
EDh	Встановити індикатори стану
EEh	Ехо-діагностика
F0h	Вибрати альтернативний режим
F2h	Ідентифікатор розширеної клавіатури
F3h	Встановлення частоти повторення клавіш
F4h	Дозвіл роботи клавіатури
F5h	Заборона сканування по замовченню
F6h	Встановити умови по замовчуванню
F7h-FAh	Встановити автоповтор всіх клавіш
FBh-FDh	Встановити автоповорот окремих клавіш
FEh	Послати повторну наступну передачу
FFh	Скид клавіатури

МОДУЛЬ IV
Шинна архітектура комп'ютера

Тема 1. Огляд системних шин комп'ютера

В перших ПК шина була основним каналом за допомогою, якої здійснювався обмін інформацією між вузлами комп'ютера. В якості системної шини використовувалась шина Multibus. Вона має дві модифікації:

- PC / XT bus (personal Computer eXtended Technology) ;
- PC / AT bus (PC Advachnology).

Шина PC / XT bus – 8-розрядна шина даних і 20-розрядна шина адреси, розрахована на тактову частоту 4,77 МГц; має 3 лінії для адаптерних переривань і 3 канали для прямого доступу в пам'ять (канали DMA - Direct Memory Access). Шина адреси обмежувала адресний простір мікропроцесора величиною 1 Мб. Використовується з МП 8086, 8088.

Шина PC / At bus – 16-розрядна шина даних і 24-розрядна шина адреси, робоча тактова частота до 8 МГц, але може використовуватися і МП з тактовою частотою 16 МГц, тому що контролер шини може ділити частоту навпіл ; має 7 ліній для адаптерних переривань і 4 канали DMA. Використовується з МП 80286 ;

Шина ISA (Industry Standard Architecture - архітектура промислового стандарту) – 16-розрядна шина даних і 24-розрядна шина адреси, робоча тактова частота 16 МГц, але може використовуватися і МП з тактовою частотою 50 МГц; в порівнянні з шинами PC / XT і PC / AT збільшено кількість ліній апаратних переривань з 7 до 15 і каналів прямого доступу до пам'яті DMA з 7 до 11. Завдяки 24 - розрядній шині адреси адресний простір збільшилася з 1 до 16 Мбайт. Теоретична пропускна здатність шини даних дорівнює 16 Мбайт / с, але реально вона нижча, близько 3 - 5 Мбайт / с, зважаючи на низку

особливостей її використання. З появою 32 - розрядних високошвидкісних МП шина ISA стала суттєвою перешкодою збільшення швидкодії ПК ;

Шина EISA (Extended ISA) - 32-розрядна шина даних і 32-розрядна шина адреси, створена в 1989 році. Адресний простір шини 4 Гбайта, пропускна здатність 33 Мбайт / с, причому швидкість обміну по каналу МП - КЕШ - ОП визначається параметрами мікросхем пам'яті, збільшено число роз'ємів розширень. Покращена система переривань, шина EISA забезпечує автоматичне конфігурування системи і керування DMA ; повністю сумісна з шиною ISA. Шина EISA дорога і застосовується у швидкісних ПК, мережних серверах і робочих станціях;

Шина MCA (Micro Channel Architecture) – 32-розрядна шина, розроблена фірмою IBM в 1987 році для машин PC / 2, пропускна здатність 76 Мбайт / с, робоча частота 10 - 20МГц. За своїми іншими характеристикам близька до шини EISA, але не сумісна ні з ISA, ні з EISA.

В сучасних ПК маєтья ряд швидкодіючих локальних шин:

Local Bus – локальна шина не заміняла ISA, EISA, а доповнювала їх за допомогою слотів, підключена до системної шини. Реальна швидкість передачі даних по 80 Мбайт/с (теоретично досяжна - 132 Мбайт/с).

Недоліки шини:

- розрахована на роботу МП 80386,80486, не адаптована для процесорів Pentium, Pentium Pro, Power PC;
- жорстка залежність від тактової частоти МП (кожна шина VLB розрахована тільки на конкретну частоту);
- малу кількість підключаються - до шини VLB можуть підключатися тільки чотири пристрої;
- відсутня арбітраж шини - можуть бути конфлікти між підключаються пристроями.

Шина PCI (Peripheral Component Interconnect – з'єднання зовнішніх пристроїв) розроблена в 1993 р. фірмою Intel, наявність чипсету (контролер для

входу в системну шину), робить шина PCI незалежною і дозволяє працювати з різними пристроями, а також здійснювати обмін даними, які знаходяться у зовнішній пам'яті.

Основні можливості шини PCI:

- синхронний обмін (32 і 64 розрядний) даними, використовується одні й ті самі лінії шини для передачі адрес і даних;
 - підтримка режиму пакетних передач, який дозволяє не витратити час шини на установку адреси кожного елемента даних адрес, адрес автоматично модифікується чіпсетом, що автоматично збільшує швидкість обміну;
 - максимальні пропускні здібності при частотах;
 - підтримка зовнішнього кеша із зворотнім і сквозним записом;
 - автоматичне конфігурування карт при включенні живлення;
 - підтримка режиму MultiplyBusMaster, при якому на шині можуть працювати декілька пристроїв;
 - встановлення запитів переривання здійснюється по рівню, що дозволяє використовувати для декількох пристроїв;
- Специфікація шини дозволяє використовувати одну лінію для декількох периферійних пристроїв.

Чіпсет PCI є основним зв'язковим елементом між усіма частинами плати. Його можна поділити на дві функціональні частини: 1 – забезпечення зв'язку шини PCI з локальною шиною, цю частину прийнято називати, головним мостом.

Чіпсет шини PCI не тільки зв'язує різноманітні шини ПК, а є зв'язним елементом між всіма пристроями системної плати. У загальному випадку чіпсет можливо поділити на 2 функціональні частини: одна забезпечує зв'язок шини PCI з локальною шиною процесора і пам'яттю, яку називають головним мостом; іншу – взаємодію з ISA та EISA, яку називають мостом PIIX.

Функції головного моста:

- ✓ обслуговування керуючих і конфігуруючих сигналів процесора;

✓ мультиплексування адреси та формування керуючих сигналів динамічної пам'яті;

- ✓ зв'язок шини даних пам'яті з локальною шиною;
- ✓ формування сигналів зовнішньої кеш-пам'яті;
- ✓ забезпечення когерентності даних в обох рівнях кеш-пам'яті та оперативної пам'яті;
- ✓ зв'язок мультиплексової шини адреси і даних PCI з сигналами процесора та ОП;
- ✓ підтримка магістрального інтерфейса AGP, який призначений для підключення потужних графічних пристроїв.

Функції PIIX-моста:

- ✓ організація зв'язку між PCI та іншими шинами для синхронізації частот цих шин;
- ✓ комутація ліній запитів переривань шин PCI та ISA;
- ✓ комутація каналів DMA;
- ✓ синхронізація моста з внутрішньою шиною X-bus;
- ✓ реалізація контролера магістралі USB.