

Тема 1. Особливості побудови спеціалізованих та промислових МПС

- 1.1 Визначення, загальна структура та принципи організації МПС.
- 1.2 Визначення розподіленої МПС. Розподіленість структури як властивість сучасних МПС.
- 1.3 Типова структура розподіленої МПС.
- 1.4 Класифікація каналів зв'язку розподіленої МПС.

Список літератури

1. Мікропроцесорні та мікроконтролерні системи: підручник. У 2 ч. Мікропроцесорні системи / А.О. Новацький. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. – 367 с.
2. Грищук Ю.С. Мікроконтролери: архітектура, програмування та застосування в електромеханіці: навч. посіб. / Ю.С. Грищук. – Харків: НТУ «ХПІ», 2019. – 384 с.
3. Цирульник С.М. Проектування мікропроцесорних систем: навч. посібник / С.М. Цирульник, Г.Л. Лисенко. – Вінниця: ВНТУ, 2010 – 201 с.
4. Совін Я.Р. Мікропроцесори в системах технічного захисту інформації: навч. посібник / Я.Р. Совін, Ю.М. Наконечний. – Львів: Видавництво Львівської політехніки, 2013. – 308 с.
5. Програмування мікроконтролерів систем автоматики: конспект лекцій для студентів базового напрямку 050201 «Системна інженерія» / Укл.: А.Г. Павельчак, В.В. Самотий, Ю.В. Яцук – Львів: Львівська політехніка. – 2012. – 143 с.
6. Мікропроцесорна техніка: Підручник / Ю. І. Якименко, Т.О. Терещенко, Є.І. Сокол, В.Я. Жуйков, Ю.С. Петергеря. – К.: ШЦ «Видавництво «Політехніка»; «Кондор», 2004. – 440 с.
7. Geoffrey Brown. Discovering the STM32 Microcontroller. Creative Commons Attribution-NonCommercialShareAlike 3.0 Unported. 2016. 244 p.

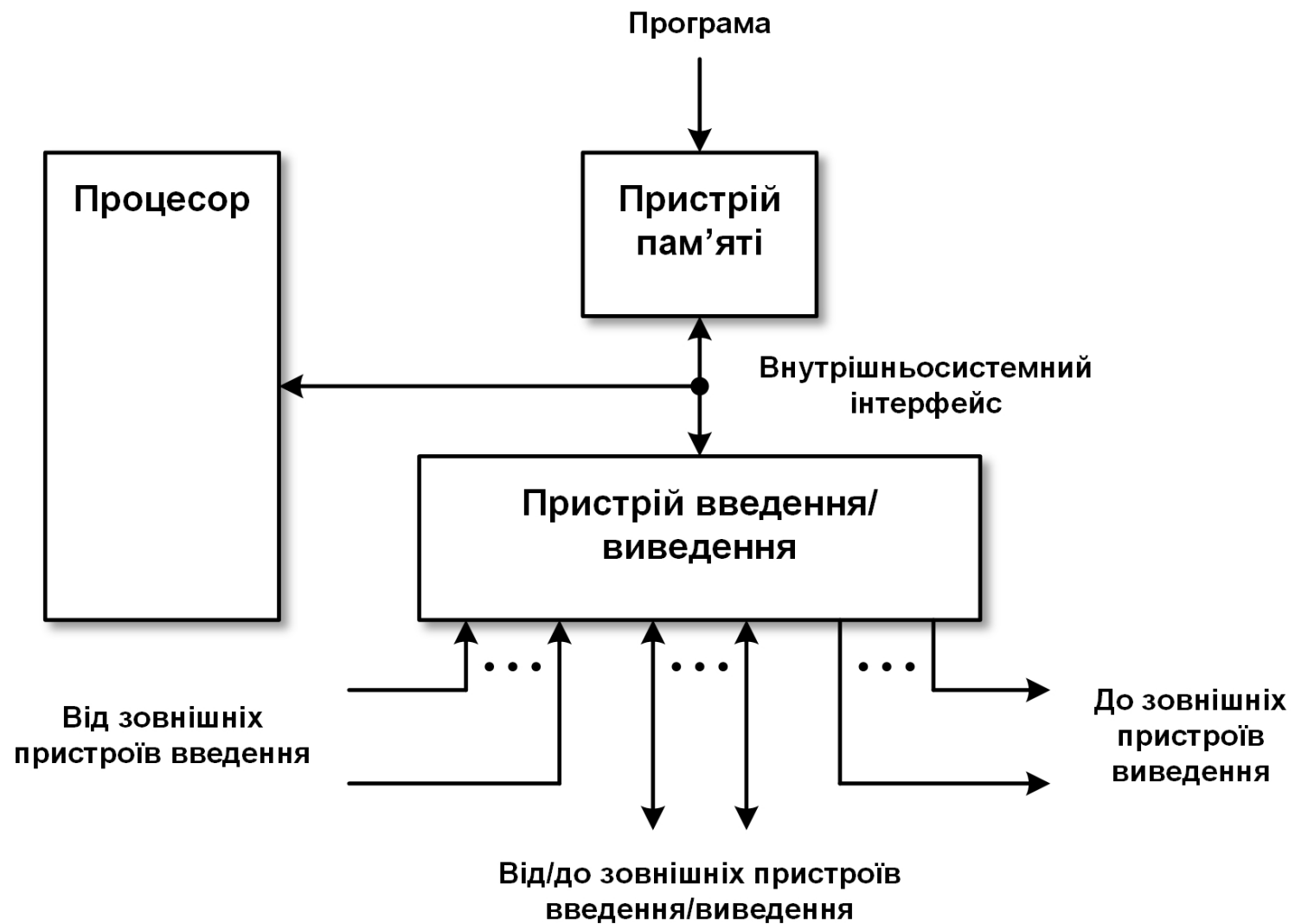
1.1 Визначення та загальна структура МПС

МПС – цифрові системи обробки інформації і управління, функціональні можливості яких визначаються програмним забезпеченням, а взаємозв'язок із зовнішнім середовищем забезпечується зовнішніми пристроями.

Додаткові властивості: 1) практична реалізація на базі однієї або декількох ВІС або НВІС та 2) впровадження принципу програмного управління.

МПС будуються по структурі цифрової обчислювальної машини, тому обов'язково містять процесори, запам'ятовувальні пристрої, пристрої введення/виведення.

Загальна структура МПС



Загальна структура МПС

Процесор виконує основні функції управління і синхронізації роботи, як внутрішніми блоками та вузлами, так і зовнішніми модулями та пристроями, а також виконує обчислювальну обробку інформації (арифметичні та логічні операції на кодовими послідовностями).

Пристрій пам'яті призначений для зберігання коду програми, за яким здійснюється функціонування МПС, а також зберігання оброблюваних даних, проміжних та кінцевих результатів обчислень.

Пристрої введення/виведення забезпечують взаємозв'язок із зовнішніми пристроями шляхом побудові каналів введення/виведення, обмін за якими організується згідно стандартних або нестандартних інтерфейсів, за правилами та процедурами стандартних або нестандартних протоколів.

Внутрішньо системний інтерфейс слугує для об'єднання пристроїв МПС між собою та забезпечення сумісної роботи у переважній кількості випадків під управлінням процесора, іноді під управлінням програмованих контролерів.

Принципи організації МПС

1) **функціональна закінченість модулів та блоків МПС:** намагаються, що б виконання зовнішньої або внутрішньої функції МПС забезпечувалося тільки переважно одним блоком або модулем;

Наслідок – простота налагоджування, модернізації й розширення МПС

2) **регулярність структури:** для побудови МПС переважно застосовують ІС регулярної структури – ІС пам'яті, контролери інтерфейсів, системні контролери тощо, у випадку застосованих нестандартних модулів – ПЛІС;

Наслідок – простота проектування й налагоджування, швидкість розробки МПС

3) **шинна організація зв'язку:** блоки, модулі МПС з'єднуються між собою за допомогою шин даних, адреси й управління, що призводить до необхідності застосування адрес (імен) пристроїв МПС (адресний принцип).

Наслідок – зниження апаратних витрат й масогабаритних показників, простота розширення МПС

Поняття розподіленої МПС

Характерна риса сучасних спеціалізованих і промислових МПС:

розподіленість

у підходах до вирішення інформаційних задач збору даних, їх обчислювальної обробки, приймання та формування керуючих рішень, що призводить до розподіленості структури МПС.

Розподілена МПС (РМПС) – це МПС, яка отримала специфіку територіально розосередженої системи

Розподілена МПС – це сукупність незалежних мікропроцесорних пристроїв (МПП), які взаємодіють один з одним з метою рішення задач, які не вирішуються одним МПП індивідуально

Розподілена МПС – це сукупність незалежних МПП, яка представляється користувачу як єдина система

Виходячи з цих визначень, невід'ємними частинами РМПС є
цільові завдання, МПП та канали зв'язку

Поняття розподіленої МПС

Цільові завдання (ЦЗ) – це завдання, для вирішення яких призначена дана РМПС і які вирішуються усією системою в цілому, але не її компонентами відокремлене

МПП – це мікропроцесорні пристрої, кожний з яких взаємодіє з іншими МПП, що входять до РМПС, з метою вирішення певної задачі, яка не вирішується одним МПП індивідуально

Канал зв'язку (КЗ) – це канали, які забезпечують обмін інформацією між МПП, які входять до РМПС

Побудова РМПС для важкодоступних і рухомих об'єктів має особливість:

ієрархічний принцип структури,

тобто мають на кожному рівні ієрархії явно виражений «центр» і підлеглих йому абонентів.

Обмін даними між довільними абонентами, крім центру, є можливим, але не обов'язковим.

Приклад 1. Система автоматичного збору метеорологічної інформації

Завданням системи є доставка інформації від датчиків в центр збору даних. На рівні цільової завдання визначається:

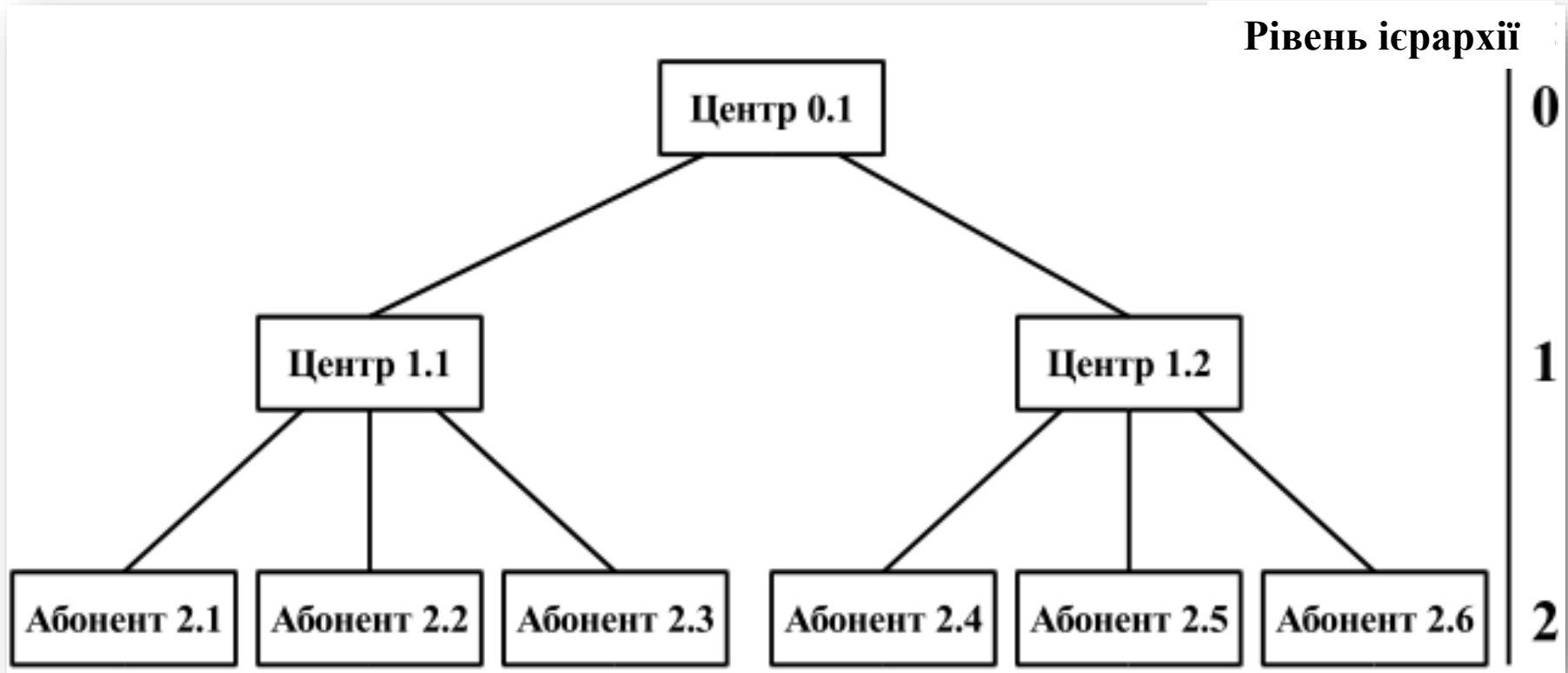
- максимальна кількість датчиків;
- тип і кількість інформації для кожного датчика;
- терміни збору інформації (наприклад: безперервний збір, період збору та ін.) для кожного датчика і типу інформації;
- спосіб передачі інформації для кожного датчика - по ініціативи датчика або центру;
- додаткові функції системи, що не належать безпосереднім але до вирішення основного завдання збору метеорологічної інформації;
- інші характеристики, важливі для функціонування системи в цілому.

Приклад 2. Система оповіщення населення про надзвичайні ситуації

Завданням системи є доведення сигналів оповіщення до об'єктів оповіщення. На рівні цільової завдання визначається:

- максимальна кількість сигналів оповіщення;
- максимальна кількість об'єктів оповіщення;
- спосіб доведення сигналів - звуковий (сирена, мова і т.п.), візуальний («мигалка», дисплей, рядок, що біжить і т.п.);
- терміни доведення сигналу від джерела до кожного об'єкта;
- способи підтвердження об'єктом отримання сигналу оповіщення;
- додаткові функції системи, що не відносяться безпосередньо до вирішення основного завдання оповіщення;
- інші характеристики, важливі для функціонування системи в цілому.

Типова структура РМПС для важкодоступних і рухомих об'єктів

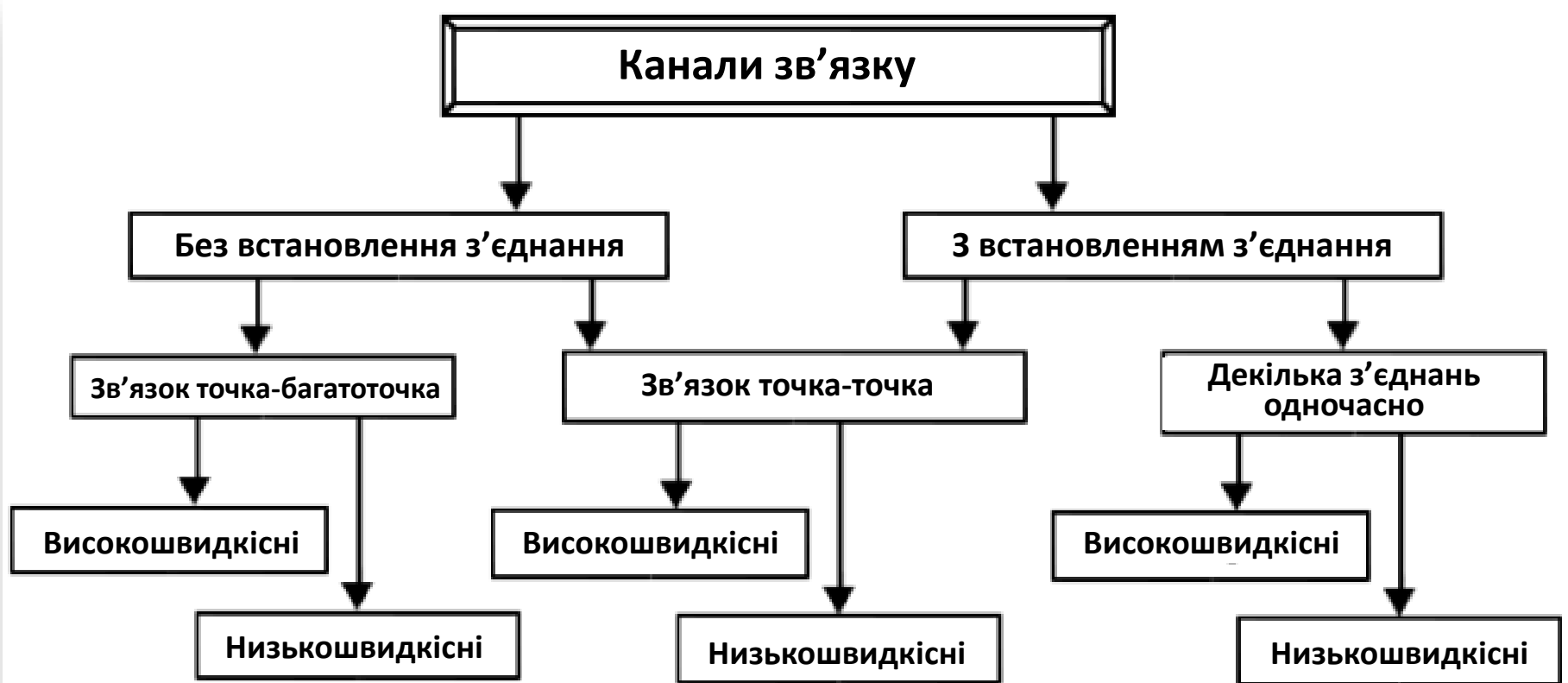


Завдання РМПС можуть сильно розрізнятися, але структура цих систем однакова і являє ієрархією, що включає кілька рівнів. Найбільш ефективна ієрархічна структура, коли весь обмін даними проводиться між центром і безпосередньо підлеглими йому абонентами (центрами).

Класифікація каналів зв'язку

Вибір каналів зв'язку і оцінка придатності їх для вирішення завдань, поставлених перед РМПС – важливий етап проектування РМПС і МПП

Класифікація каналів за кількома параметрами – швидкісним характеристикам, способу встановлення з'єднання і способу обміну даними.



Класифікація каналів зв'язку

Високошвидкісні КЗ – це КЗ, що дозволяють реалізувати повнофункціональний режим роботи РМПС.

Низькошвидкісні КЗ – ті КЗ, які дозволяють реалізувати функції РМПС в режимі обмеженого функціоналу.

Поняття «низькошвидкісний» і «високошвидкісний» канал зв'язку залежать від конкретної ЦЗ та вимог до конкретної РМПС.

За способом установки з'єднання КЗ класифікуються:

– **КЗ з постійним з'єднанням.** Це одноразово встановлюється і постійно контрольоване з'єднання. При виявленні розриву провадиться відновлення з'єднання. Постійне з'єднання характерне для КЗ, що використовують протокол TCP/IP;

– **КЗ із з'єднанням на час сеансу зв'язку.** В цьому випадку при передачі повідомлення проводиться процедура встановлення з'єднання, потім обмін даними, після чого з'єднання розривається. З'єднання на час сеансу характерно для прямих модемних з'єднань (PSTN, GSM);

– **КЗ без встановлення з'єднання.** Обмін даними проводиться відразу, без процедури встановлення з'єднання. Такий спосіб характерний, наприклад, для радіоканалу.

Тема 2. Програмна модель розподіленої мульти-мікропроцесорної системи

2.1 Програмна модель DOD:

- рівні моделі DOD;
- призначення рівнів моделі DOD.

2.2 Складові компоненти програмного забезпечення МПП у складі розподіленої МПС:

- програмування без використання ОС;
- спеціалізовані ОС та їх структура;
- операційні системи загального призначення.

2.1 Програмна модель DOD

Програмні моделі, які дозволяють розглядати взаємодію МПП у складі РМПС на різних рівнях уявлення:

- ❑ **сьомирівнева модель OSI** (Open System Interconnection);
- ❑ **чотирьохрівнева модель TCP/IP (DOD, Department of Defence).**

Програмна модель DOD створена для реалізації стека протоколів TCP/IP, є по суті спрощеної моделлю OSI.

Модель DOD складається з чотирьох рівнів:

прикладний рівень або рівень додатків, транспортний рівень, міжмережевий рівень, рівень мережевого доступу

Програмні моделі взаємодії між МПП – OSI і DOD – досить універсальні, але в той же час не описують деяких обмежень, що виникають при проектуванні реальних РМПС, наприклад швидкісних обмежень фізичних каналів зв'язку, обмеженої швидкодії МПП.

2.1 Програмна модель DOD

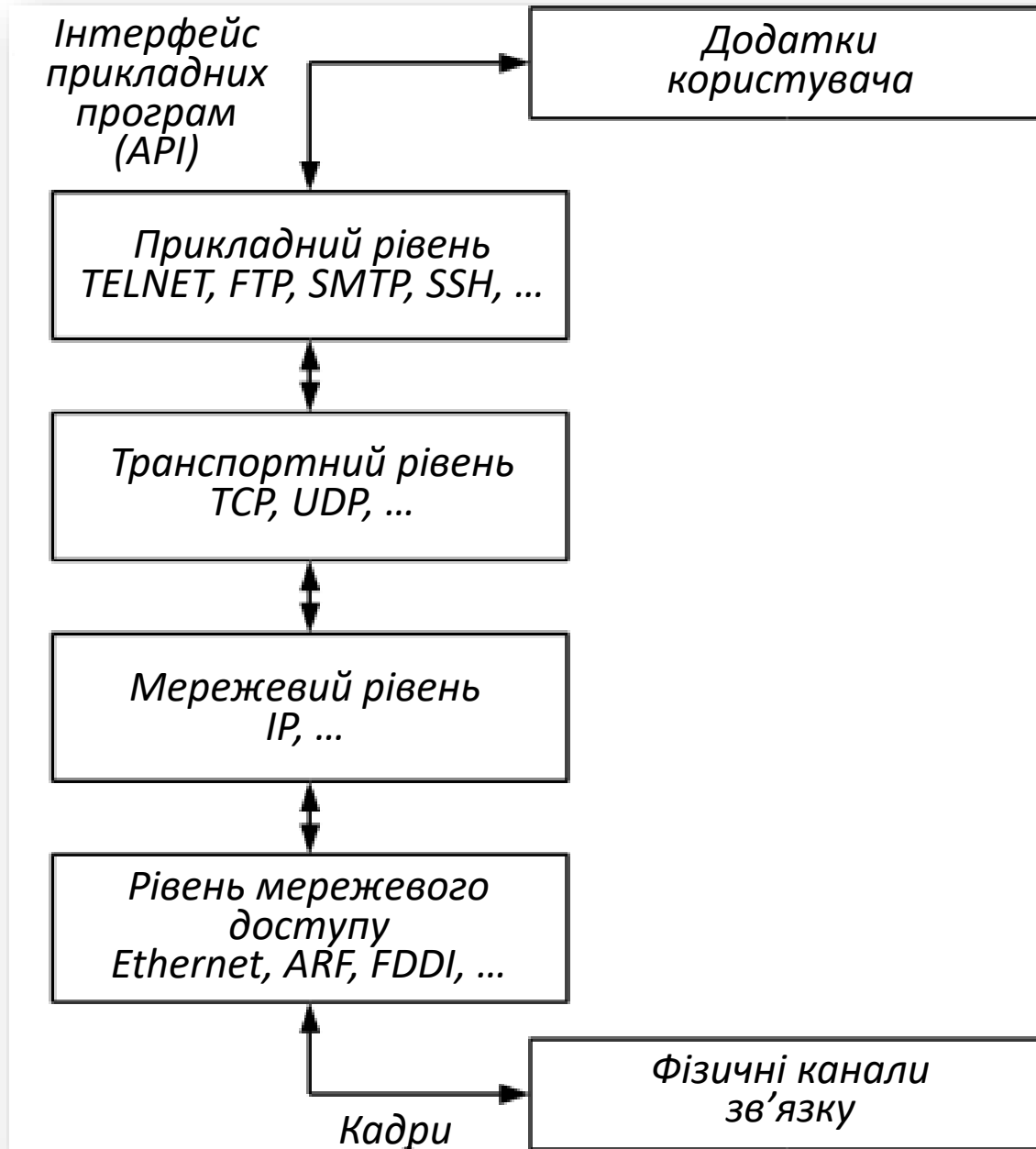
Прикладний рівень або рівень додатків (Process/Application), що забезпечує програмний інтерфейс для взаємодії з додатками користувача. Покладаються задачі перетворення протоколів, шифрування/дешифрування даних, підтримка сеансу зв'язку, дозволяючи додаткам взаємодіяти між собою тривалий час, створення і завершення сеансу зв'язку.

Транспортний рівень (Transport), призначений для забезпечення надійної передачі даних від відправника до одержувача.

Міжмережевий рівень (Internet) відповідає за трансляцію логічних адрес і імен у фізичні, визначення найкоротших маршрутів, комутацію і маршрутизацію, відстеження неполадок в мережі.

Рівень мережевого доступу (Network Access), призначений для забезпечення взаємодії мереж на фізичному рівні і контролю за помилками. Дані упаковуються в кадри і здійснюється передача-прийом кадрів по фізичних каналах зв'язку, а також контроль цілісності кадрів.

2.1 Програмна модель DOD



2.2 Складові компоненти програмного забезпечення МПП у складі розподіленої МПС

Спільні програмні компоненти, присутні в будь-якому ПЗ, яке виконується на МПП в складі РМПС:

- 1) ПЗ, що забезпечує обмін даними між МПП, функції даного ПЗ - реалізація алгоритмів обміну даними по кожному з КЗ, підключеному до МПП;
- 2) інша складова ПЗ – це програмні модулі, що реалізують цільову функцію.

При розробці ПЗ для РМПС слід визначити наступне:

- 1) загальний алгоритм роботи всієї системи в цілому і кожного МПП окремо (алгоритм складається з постановки завдання на проектування);
- 2) функції і алгоритм роботи МПП в складі РМПС (функції МПП обґрунтовуються виходячи із завдання на проектування, алгоритм роботи кожного МПП описується виходячи з реалізованих їм функцій і алгоритму роботи системи в цілому);
- 3) функції і алгоритм роботи кожного програмного модуля;
- 4) спосіб реалізації програмних модулів;
- 5) механізм взаємодії між різними програмними модулями.

2.2 Складові компоненти програмного забезпечення МПП у складі розподіленої МПС

Залежно від призначення і можливостей апаратного забезпечення МПП застосовуються основні методи проектування і реалізації програмного забезпечення:

- 1) програмування без використання операційної системи (ОС);
- 2) спеціалізовані ОС; 3) ОС загального призначення.

1. Програмування без використання ОС.

Застосовується при розробці ПЗ для МПП, які не мають будь-якого системного ПЗ.

Переваги – повний доступ програміста до апаратного забезпечення процесора (МК), що дозволяє оптимальним чином конфігурувати периферійні пристрої та добитися, наприклад, максимально можливої продуктивності.

Недолік – трудомісткість, обумовлена відсутністю стандартних засобів і необхідністю писати «з нуля» підтримку всіх необхідних функцій роботи з апаратурою, протоколами та ін.

Мікроконтролери з низькою продуктивністю сімейства AVR або PIC, а також молодші представники сімейств ARM, виробники МК додають набори програмних бібліотек для роботи з апаратурою.

2.2 Складові компоненти програмного забезпечення МПП у складі розподіленої МПС

2. Спеціалізовані ОС.

Типові задачі МПП при реалізації ПЗ – переключення процесів, організація буферів введення/виведення, обробка переривань тощо.

Спеціалізовані ОС – eCos, ChibiOS/RT, FreeRTOS, Trace Mode та ін. (зазвичай спеціалізовані ОС є системи реального часу).

Спеціалізовані ОС мають кілька базових компонентів:

- **планувальник завдань;**
- **рівень абстракції обладнання;**
- **інтерфейс користувача ПЗ.**

До складу спеціалізованої ОС можуть включатися додаткові компоненти, що розширюють функціональність ОС: математичні бібліотеки, підтримка багатопотоковості, налагоджувальні функції та ін.

Планувальник завдань дозволяє організувати в ПЗ кілька незалежних процесів (завдань), що виконуються паралельно. Така необхідність виникає при організації обміну даними по декількох КЗ одночасно, що актуально для МПП в складі РМПС. Часто спеціалізована ОС містить кілька планувальників задач, один з яких можна вибрати при її конфігурації.

2.2 Складові компоненти програмного забезпечення МПП у складі розподіленої МПС

Рівень абстракції обладнання являє собою набір драйверів пристроїв. Кожен драйвер має стандартний для даної ОС програмний інтерфейс, що дозволяє поліпшити переносимість ПЗ з одного пристрою на інший.

Інтерфейс користувача ПЗ являє собою набір системних функцій, за допомогою яких ПЗ користувача звертається до різних компонентів ОС - планувальника завдань, драйверів введення-виведення (рівню абстракції обладнання) і здійснює міжпроцесну і міжпотоківу взаємодію.

Дані ОС спеціалізовані в тому сенсі, що містять обмежений набір функцій у порівнянні з ОС загального призначення. Як правило, набір функцій спеціалізованої ОС не містить функцій файлового введення-виведення, не дає можливості розділяти права доступу до ресурсів. Оскільки дані ОС найчастіше застосовуються в МК і знаходяться фізично в ПЗП разом із ПЗ користувача, то вони не містять функцій, що дозволяють завантажувати і запускати програми з зовнішніх носіїв.

Переваги – відносна простота організації спеціалізованих ОС дозволяє, в разі необхідності, легко додавати потрібні модулі, а також позбавляти програміста від рутинних операцій по переключенню процесів, організації функцій термінального введення-виведення та інших подібних речей.

2.2 Складові компоненти програмного забезпечення МПП у складі розподіленої МПС

Недолік – дещо підвищені вимоги до продуктивності МК, для яких планується застосування спеціалізованої ОС.

При правильній побудові ПЗ уся взаємодія між спеціалізованою ОС і ПЗ користувача відбувається тільки через інтерфейс призначеного для користувача ПЗ.

3. Операційні системи загального призначення.

З ростом продуктивності і зниженням ціни МП (МК) в МПТ все частіше застосовуються ОС загального призначення, такі як Linux, QNX, WinCE і т.п.

Переваги – створення ПЗ для таких систем найбільш комфортно з точки зору проектувальників і програмістів, оскільки, по-перше, для даних ОС існує дуже широкий інструментарій для написання і налагодження програм, по-друге, тим, що можна створювати і налагоджувати ПЗ на персональному комп'ютері з тієї ж ОС, що і на цільовому МПП, а потім шляхом крос-компіляції перенести вже налагоджене ПЗ на необхідну МПП.

Недолік – підвищені вимоги до продуктивності МП (МК), для яких планується застосування спеціалізованої ОС, та підвищена вартість ОС загального призначення.

Тема 3. Поширені архітектури спеціалізованих та промислових МПС

3.1 Системи з централізованим, децентралізованим і комбінованим керуванням.

3.2 Системи зі змінною структурою.

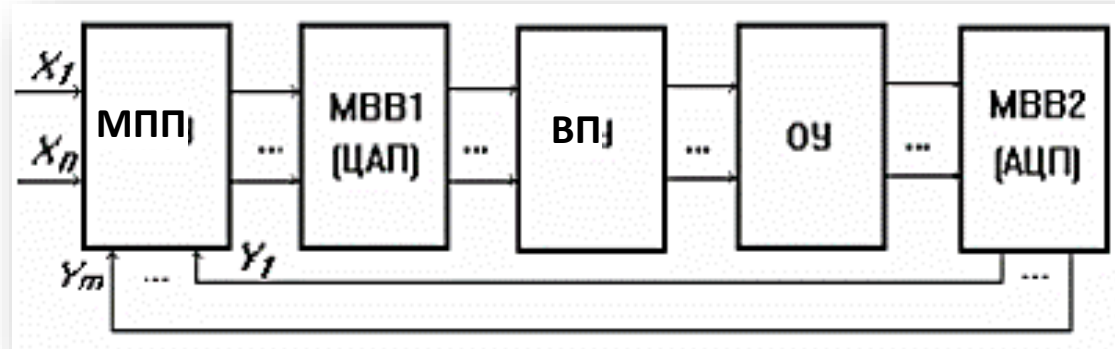
3.3 Системи з резервуванням:

- ✓ мажоритарна обробка сигналів;
- ✓ обробка з перемиканням каналів.

3.4 Ієрархічні системи.

3.1 Системи з централізованим, децентралізованим і комбінованим керуванням

1) МПС з централізованим керуванням



Задача обробки вхідних сигналів X_1, \dots, X_n з метою формування керуючих впливів Y_1, \dots, Y_m вирішується МПП, що включає МП і елементи пам'яті, які з'єднані каналами зв'язку через цифро-аналогові перетворювачі (ЦАП), виконавчі пристрої (ВП) з об'єктом (або об'єктами) управління (ОУ).

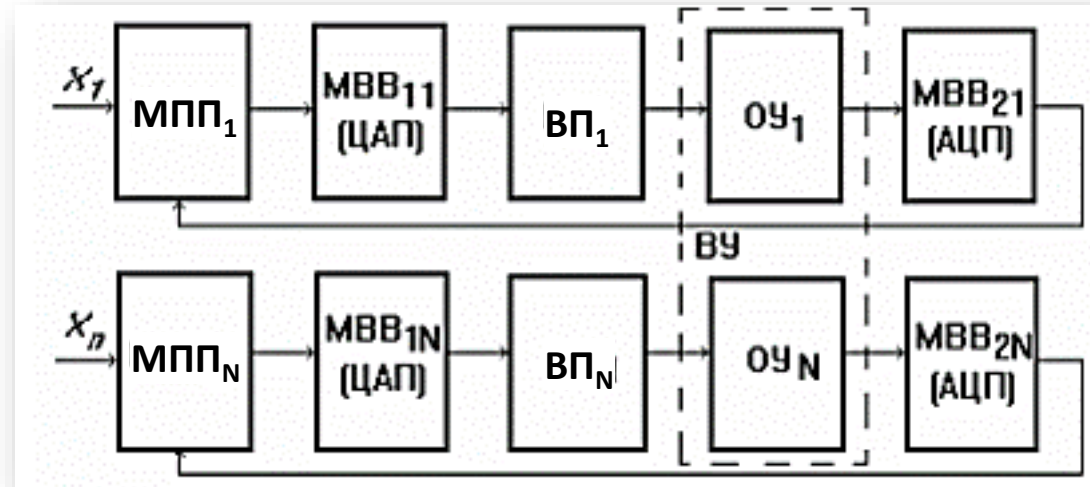
Зворотній зв'язок про стан ОУ забезпечується сигналами від ОУ, які надходять на МПП через аналого-цифрові перетворювачі (АЦП).

Якщо здійснюється управління одним, але складним багатовимірним об'єктом (роботом, прокатним станом, доменною піччю, літаком, космічним літальним апаратом і т. п.), то така система є **зв'язаною**.

Якщо ж вирішується завдання управління сукупністю незалежних по керованим параметрам одновимірних об'єктів, то система є **незв'язаною**.

3.1 Системи з централізованим, децентралізованим і комбінованим керуванням

2) МПС з децентралізованим керуванням



В кожен контур управління включається автономний МПП

МПП розміщені в безпосередній близькості від об'єкта управління $ОУ$ або вбудовані в нього і функціонально орієнтовані на рішення конкретних завдань. Як МПП широко застосовується програмовані регулюючі МК.

У децентралізованих системах МПП може вводитися для передачі йому функцій диспетчера. У цьому випадку реалізується **комбіноване управління**. В комбінованих системах використовується обидві перераховані структури управління.

3.1 Системи з централізованим, децентралізованим і комбінованим керуванням

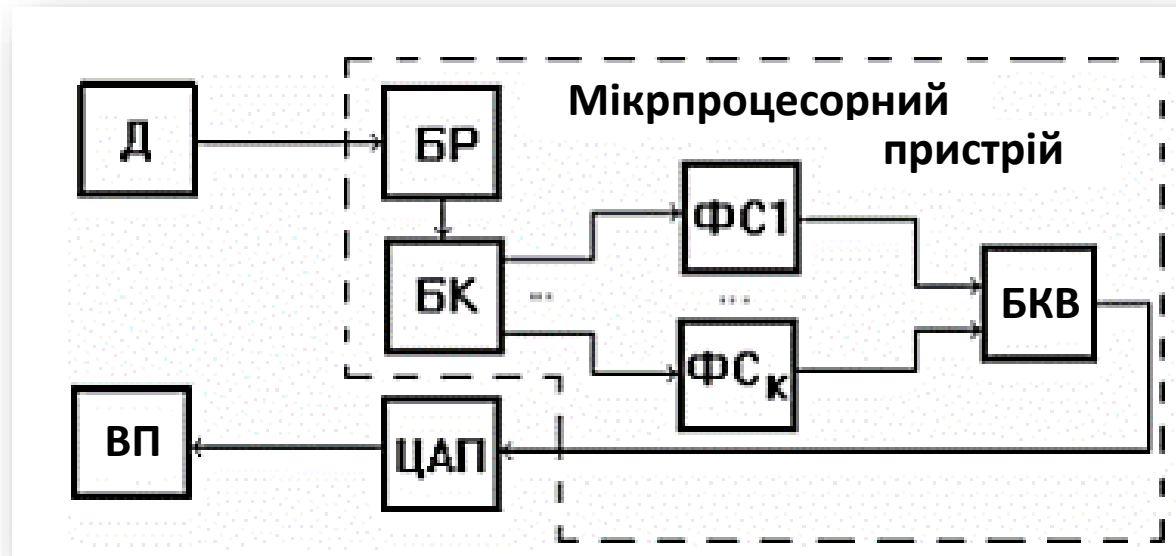
Вибір структури управління в МПС, побудованих на базі МПК ВІС, залежить від багатьох взаємопов'язаних факторів:

вартість і надійність систем, їх живучість, гнучкість, здатність працювати в масштабі реального часу.

Застосування принципу децентралізованого (розподіленого) управління при побудові МПС в технічному і економічному плані мають переваги в порівнянні з іншими структурами МПС.

3.2 Система зі змінною структурою

МПС зі структурою, яка перебудовується:



Склад системи - цифровий датчик візуальної інформації Д, інформація з якого надходить в блок розпізнавання зображення БР, який формує електричний логотип зображення. Блок комутації алгоритмів управління БК виробляє вибір і обробку програми обслуговування сформованого логотипу. За допомогою сигналів формувачів керуючих сигналів $\Phi C_1, \dots, \Phi C_k$ блок комутації виходу БКВ видає на вихід МПП виконавчий сигнал, який через ЦАП подається на виконавчий пристрій ВП.

3.2 Система зі змінною структурою

Завдання, які вирішуються такими МПС, *залежать від характеру вхідних впливів, що надходять в систему* (наприклад, управління роботом може здійснюватися за різними алгоритмами в залежності від результату розв'язання задачі розпізнавання представленого роботу об'єкта).

В цьому і подібних випадках структура МПС виявляється змінної. У МПС вона **перебудовується програмно.**

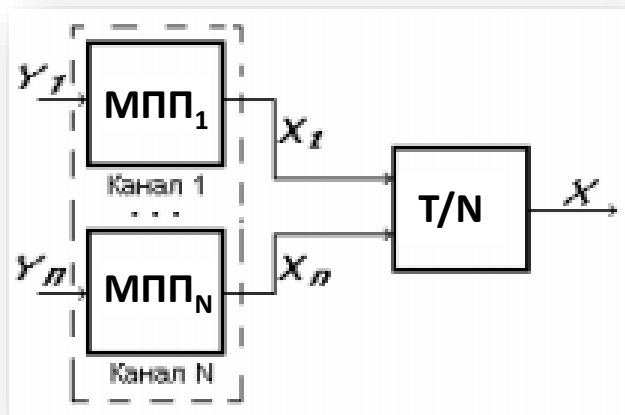
Здійсненність перебудови МПС, виконуваної в реальному масштабі часу на програмному рівні, є **наслідком застосування в автоматичних системах високопродуктивних МП**, на які покладаються завдання обробки великих потоків інформації, пов'язаної зі статистичними експрес-аналізом випадкових сигналів, їх ідентифікацією, класифікацією, розпізнаванням зображень і т.п. Це в кінцевому рахунку істотно покращує показники якості управління системою.

3.3 Системи з резервуванням

Одним із шляхів збільшення відмовостійкості МПС є резервування. Резервування підрозділяється на:

апаратне, програмне та інформаційне.

Поширеними методами апаратного резервування є методи, засновані на мажоритарній обробці і обробці з перемиканням каналів.



Y_1	Y_2	Y_3	X
-	-	-	-
-	-	+	-
-	+	-	-
-	+	+	+
+	-	-	-
+	-	+	+
+	+	-	+
+	+	+	+

T/N – критерій мажоритарності

Мажоритарна обробка сигналів

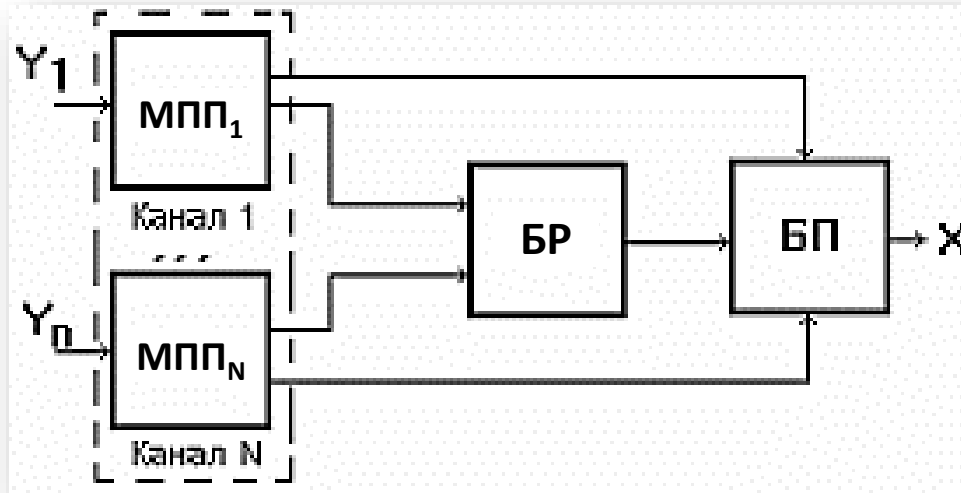
Складається з N незалежних каналів обробки інформації і залишається в робочому стані до тих пір, поки зберігається працездатність T з N каналів.

В системі «2 з 3» працездатний стан каналу визначається з таблиці, значками «+» і «-» визначено відповідно працездатний і непрацездатний стан каналів.

3.2 Системи з резервуванням

В МПС з перемиканням каналів надлишкові (резервні) канали обробки інформації включаються в роботу тільки після виходу з ладу основного або раніше заміненого каналу.

У таких системах є додаткові блоки розпізнавання (БР) несправних каналів і їх перемикання (БП).



Обробка з перемиканням каналів

3.4 Ієрархічні системи

Складні об'єкти управління (літаки, космічні апарати, прокатні стани і т. п.) являють собою сукупність взаємопов'язаних багаторежимних керованих систем, об'єднаних єдиною системою управління.

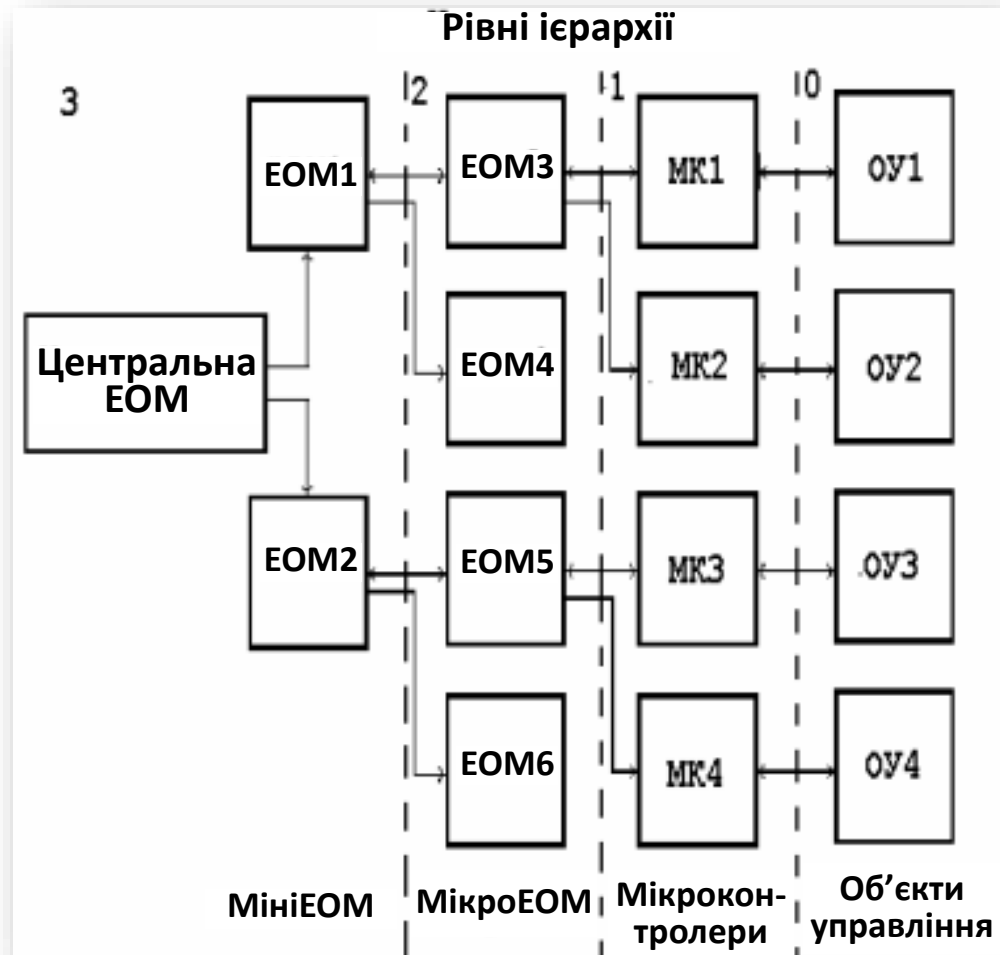
Основоположні принципи, які

визначають структуру МПС управління подібними об'єктами:

- ✓ **ієрархічність;**
- ✓ **незалежність управління за рівнями ієрархії;**
- ✓ **інформаційна замкнутість.**

Особливість МК на нульовому рівні ієрархії – на його виході не використовується мультиплексування (число ЦАП дорівнює числу вихідних ланцюгів контролера).

Така побудова МПС пов'язана з необхідністю запам'ятовування кожного значення керуючого сигналу після зупинки обчислювального процесу.



3.4 Ієрархічні системи

Перетворювачі аналогових сигналів в код дозволяють підключати МК до безперервних і дискретних датчиків, з виконавчими механізмами, а також з різними пристроями дискретного і логічного управління.

Всі алгоритми мікроконтролера досить універсальні і в функціональному відношенні еквівалентні типовим ланкам МПС управління об'єктом або типовий «зв'язці» таких ланок.

За оцінками фахівців, існує обмежене число (орієнтовно 20 -25) алгоритмів, комбінація яких дозволяє автоматизувати управління процесами і об'єктами практично будь-якого ступеня складності. Ці алгоритми, оформлені у вигляді бібліотеки програм, зберігаються в постійній пам'яті і можуть бути використані в будь-якому заданому поєднанні.

На першому рівні ієрархії знаходяться серійні мікроЕОМ, які забезпечують управління групою функціонально пов'язаних об'єктів. На цьому рівні, відповідному локальному управлінню, застосовують серійні мікроЕОМ багатofункціонального призначення.

Другий рівень включає керуючі вузли, реалізовані на базі мініЕОМ, які координують роботу групи локальних систем.

На третьому рівні розташовується центральна управляюча ЕОМ, яка є вищим координуючим органом в даній структурі.

Тема 4. Паралельна та розподілена обробка даних

4.1 Поняття паралельних обчислень:

- ✓ визначення та основна мета паралельних обчислень;
- ✓ актуальність застосування паралельних систем;
- ✓ стримуючі фактори впровадження багатопроцесорних паралельних систем.

4.2 Рівні паралелізму:

- ✓ рівні розпаралелювання операцій;
- ✓ паралелізм за ступенем гранулярності.

4.3 Закони Амдала та Густафсона.

4.1 Поняття паралельних обчислень

Паралельні обчислення (паралельна обробка) – це використання декількох або багатьох обчислювальних пристроїв для одночасного виконання різних частин однієї програми (одного проекту).

Паралельні обчислення – такий спосіб організації комп'ютерних обчислень, при якому програми розробляються як набір взаємодіючих обчислювальних процесів, що працюють паралельно.

Паралельні обчислення – обчислення, які можна реалізувати на багатопроцесорних системах з використанням можливості одночасного виконання багатьох дій, що породжуються процесом вирішення однієї або багатьох завдань (одного проекту).

Основна мета паралельних обчислень – зменшення часу рішення задачі

Багато необхідних для потреб практики завдань потрібно вирішувати в реальному часі або для їх вирішення потрібно дуже великий обсяг обчислень.

Збільшення числа процесорів не обов'язково призводить до зменшення часу виконання завдання.

4.1 Поняття паралельних обчислень

При цьому:

- ✓ паралельні обчислення не зводяться до використання тільки багатопроцесорних обчислювальних систем ;
- ✓ операції, які одночасні виконуються, повинні бути спрямовані на вирішення загального завдання;
- ✓ паралельні обчислення слід відрізнити від багатозадачних (багатопрограмних) режимів роботи послідовних ЕОМ.

Застосування паралельних обчислювальних систем є стратегічним напрямком розвитку обчислювальної техніки

Ця обставина викликано наступними причинами:

1. Випередження потреби обчислень швидкодії існуючих комп'ютерних систем:

моделювання клімату, гена інженерія, проектування інтегральних схем, аналіз забруднення навколишнього середовища, створення лікарських препаратів і ін.

Оцінка необхідної продуктивності - 10^{12} операцій за секунду (1 Tflops).

2. Теоретична обмеженість зростання продуктивності послідовних комп'ютерів.

3. Різке зниження вартості багатопроцесорних (паралельних) обчислювальних систем: 1 Cray T90 processor – 1.8 GFlops (\$2 500 000), 8 Node IBM SP2 using R6000 - 2.1 GFlops (\$500 000).

4.1 Поняття паралельних обчислень

Стримуючі фактори:

– **висока вартість паралельних систем.** Відповідно до закону Гроша (Grosch), продуктивність комп'ютера зростає пропорційно квадрату його вартості і, як результат, набагато вигідніше отримати необхідну обчислювальну потужність придбанням одного продуктивного процесора, ніж використання декількох менш швидкодіючих процесорів. Але:

- зростання швидкодії послідовних ЕОМ не може тривати нескінченно, крім того, комп'ютери схильні до швидкого морального старіння;
- сучасна тенденція побудови паралельних комплексів з типових конструктивних елементів (МП, мікросхем пам'яті, комунікаційних пристроїв) знизилла вплив цього фактору і зараз практично кожен споживач може мати в своєму розпорядженні багатопроцесорні обчислювальні системи досить високою продуктивності.

– **втрати продуктивності для організації паралелізму.**

Відповідно до гіпотези Мінського (Minsky), прискорення, що досягається при використанні паралельної системи, пропорційне двійковому логарифму від числа процесорів (тобто при 1000 процесорах можливе прискорення 10).

Але існує велика кількість завдань, при паралельному вирішенні яких досягається 100% використання всіх наявних процесорів паралельної обчислювальної системи.

4.1 Поняття паралельних обчислень

– **постійне вдосконалення послідовних комп'ютерів.** Відповідно до закону Мура (Moore) *потужність послідовних процесорів зростає практично в 2 рази кожні 18-24 місяців* і, як результат, необхідна продуктивність може бути досягнута і на "звичайних" послідовних комп'ютерах. Але

- аналогічний розвиток властивий паралельним системам;
- застосування паралелізму дозволяє отримувати бажане прискорення обчислень без будь-якого очікування нових більш швидкодіючих процесорів.

– **існування послідовних обчислень.** Відповідно до закону Амдала (Amdahl) *прискорення процесу обчислень при використанні процесорів обмежується* (алгоритмів без частки послідовних команд практично не існує). Але

- часто частка послідовних дій характеризує не можливість паралельного вирішення завдань, а послідовні властивості застосовуваних алгоритмів;
- частка послідовних обчислень може бути істотно знижена при виборі найбільш підхожих для розпаралелювання алгоритмів.

– **існуюче ПЗ орієнтоване в основному на послідовні ЕОМ.** Але

- якщо послідовні програми не дозволяють отримувати вирішення завдань за прийнятний час або ж виникає необхідність вирішення нових завдань, то необхідною стає розробка нового ПЗ і ці програми можуть реалізовуватися в паралельному виконанні;
- чисельні методи в разі багатопроцесорних систем повинні проектуватися як системи паралельних і взаємодіючих між собою процесів, що допускають виконання на незалежних процесорах.

4.2 Рівні паралелізму

– залежність ефективності паралелізму від характерних властивостей паралельних систем (відсутність мобільності для паралельних програм). На відміну від класичної схеми послідовних ЕОМ паралельні системи відрізняються різноманітністю архітектурних принципів побудови і максимальний ефект від паралелізму може бути отриманий тільки при повному використанні всіх особливостей апаратури – як результат, перенесення паралельних програм між різними типами систем стає скрутним (якщо взагалі можливий).

- при всій різноманітності архітектури паралельних систем, проте, існують і певні "усталені" способи забезпечення паралелізму (конвеєрні обчислення, багатопроцесорні системи і т.п.). Крім того, інваріантність створюваного ПЗ може бути забезпечена і за допомогою використання типових програмних засобів підтримки паралельних обчислень (типу програмних бібліотек MPI, PVM і ін.).

Паралелізм - основа високопродуктивної роботи всіх підсистем обчислювальних машин.

(організація пам'яті будь-якого рівня ієрархії, організація системного введення/виведення, організація мультиплексування шин і т.д. базуються на принципах паралельної обробки запитів. Сучасні ОС є багатозадачними і розрахованими на багато користувачів, імітуючи паралельне виконання програм за допомогою механізму переривань)

4.2 Рівні паралелізму

Розвиток процесорів також орієнтований на розпаралелювання операцій, тобто на виконання процесором більшого числа операцій за такт.

Ключові досягнення – гіперконвейеризація, суперскалярність, неупорядкована модель обробки, векторне процесування.

Потужні сервера є мультипроцесорні системи, а в процесорах активно використовується паралелізм рівня потоків.

Розпаралелювання операцій – підвищення продуктивності обчислень

1 рівень - розпаралелювання виконання операцій процесора (також зовнішніх пристроїв) на мікрорівні - конвеєризація обчислень.

Використання мікрорівневого паралелізму дозволяло лише зменшувати CPI (Cycles Per Instruction).

2 рівень - паралелізму рівня команд (інструкцій) ILP - розміщення в процесорі відразу декількох конвеєрів.

Суперскалярні процесори дозволяли досягати $CPI < 1$. Паралелізм рівня інструкцій (ILP) породив неупорядковану модель обробки, динамічне планування, станції резервації інше. Від CPI перейшли до IPC (Instructions Per Clock). ILP обмежений алгоритмом програми. Сучасні методи підвищення ILP на процесорах класу SIMD – векторне процесування, матричні процесори, архітектура VLIW.

3 та 4 рівні - паралелізм рівня потоків і рівня завдань – процесори класу SIMD (багатопотокові процесори та багатоядерні процесори).

Кращі сучасні обчислювачі - це мультікомп'ютерні і мультипроцесорні системи.

4.2 Рівні паралелізму



Іноді класифікують паралелізм за ступенем гранулярності як відношення обсягу обчислень до обсягу комунікацій.

Розрізняють **дрібнозернистий, середньозернистий і крупнозернистий паралелізм**.

Дрібнозернистий паралелізм забезпечує сам процесор, але компілятор може і повинен йому допомогти для забезпечення більшого IPC. **Середньозернистий паралелізм** – прерогатива програміста, якому необхідно розробляти багатопотокові алгоритми. Тут роль компілятора полягає у виборі оптимальної послідовності інструкцій (з великим IPC) за допомогою різних методик (наприклад, символічне розгортання циклів). **Крупнозернистий паралелізм** забезпечує ОС.

5.1 Закони Амдала та Густафсона

Метрики паралелізму:

1) ступінь паралелізму програми $D(t)$ – число процесорів, які беруть участь у виконанні програми в момент часу t ;

при старті програми $D(0) = 1$, далі програма може створювати незалежні потоки і передавати їм частину свого навантаження, при поведінки алгоритму на необмеженій машині $D(t)$ - профіль паралелізму програми

2) $T(n)$ - час виконання програми на n процесорах;

$T(n) < T(1)$, якщо паралельна версія алгоритму ефективна, $T(n) > T(1)$, якщо накладні витрати (витрати) реалізації паралельної версії алгоритму надмірно великі, за $T(1)$ взято не час виконання багатопотокової програми на одному процесорі, а час виконання однопотокової програми

3) $S(n) = T(1)/T(n)$ – прискорення за рахунок паралельного виконання;

4) ефективність системи з n процесорів $E(n) = S(n)/n$ – прискорення в перерахунку на один процесор

4.3 Закони Амдала та Густафсона

Будь-яка паралельна програма містить послідовну частину: введення/виведення, менеджмент потоків, точки синхронізації і т.п. Позначимо частку послідовної частини за f . Тоді частка паралельної частини буде $1-f$.

Амдал (1967) розглянув прискорення програми на n процесорах, виходячи з припущення лінійного прискорення паралельної частини:

$$S(n) = \frac{T(1)}{T(n)} = \frac{T(1)}{f \times T(1) + \frac{(1-f) \times T(1)}{n}} = \frac{n}{1 + (n-1) \times f}$$

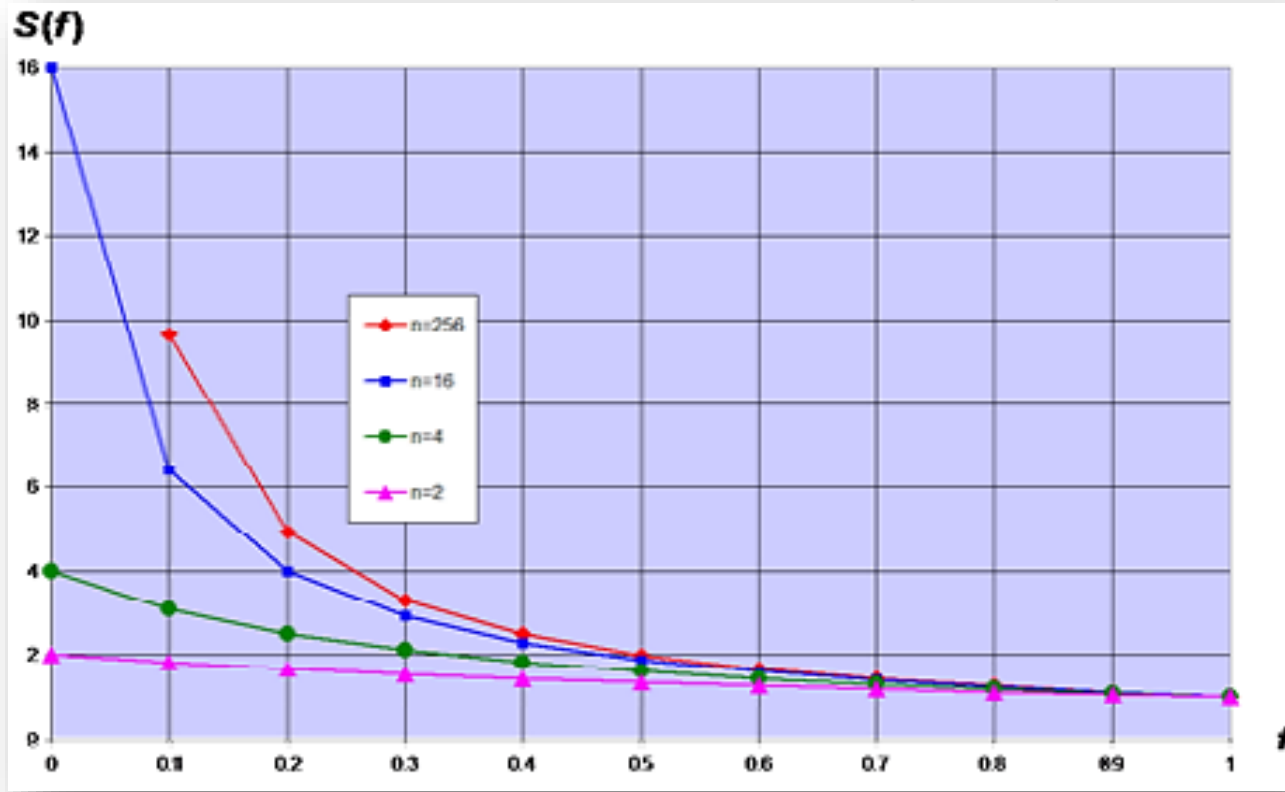
При необмеженій кількості процесорів прискорення складає всього лише $1/f$.

$$\lim_{n \rightarrow \infty} S(n) = \frac{1}{f}$$

Наприклад, якщо частка послідовної частини 20%, то теоретично неможливо отримати прискорення обчислень більш ніж в п'ять разів

При $f > 50\%$ ні про яке істотне прискоренні говорити не доводиться. Тільки якщо частка f мала, багаторазове збільшення числа процесорів стає економічно доцільним.

4.3 Закони Амдала та Густафсона



Більш того, на практиці не уникнути накладних витрат на підтримку багатопотокових обчислень, що складаються з алгоритмічних витрат (менеджмент потоків і т.п.), комунікаційних витрат на передачу інформації між потоками і витрат в вигляді дисбалансу завантаження процесорів.

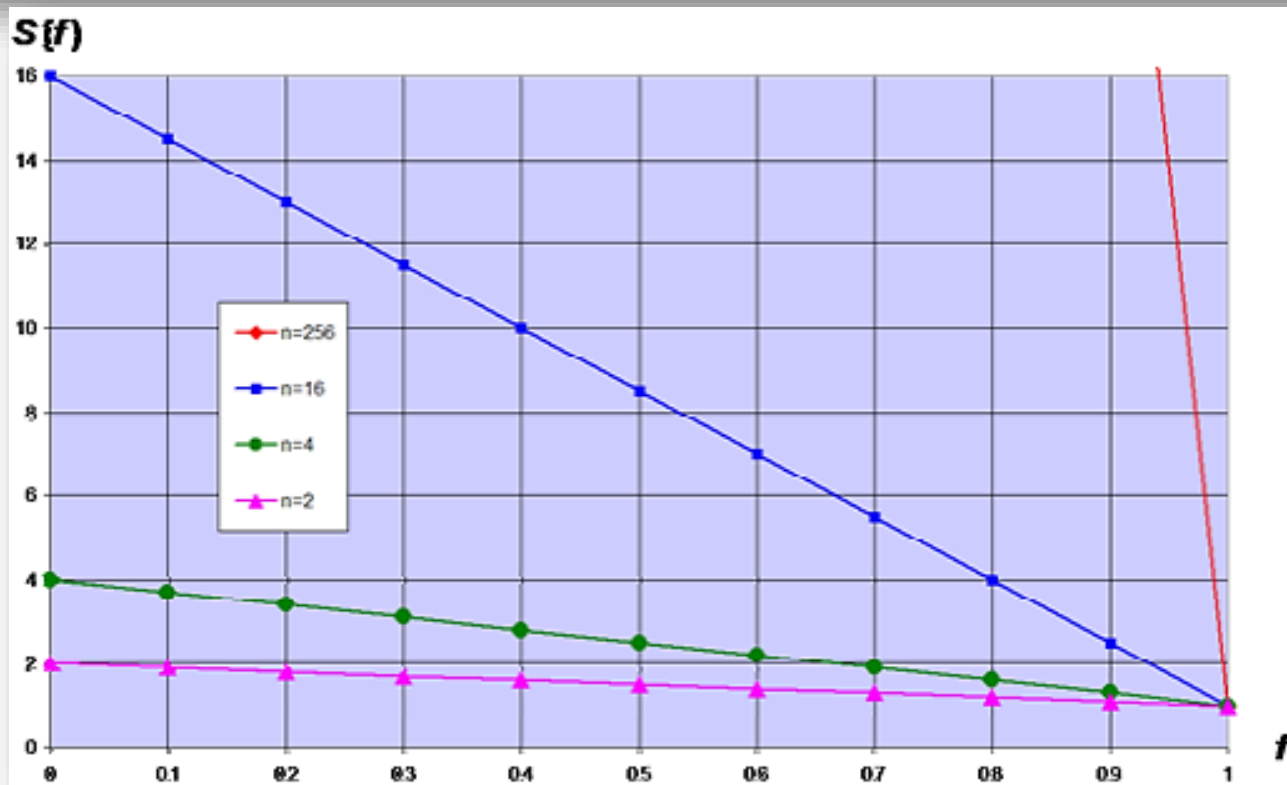
Дисбаланс завантаження процесорів виникає, навіть якщо вдалося розбити вихідну завдання на рівні по складності підзадачі, так як час їх виконання може істотно відрізнятися за різних причин (від конфліктів в конвеєрі до планувальника ОС). У точках синхронізації (хоча б в точці завершення всієї задачі) потоки змушені очікувати найдовше виконуваних, що призводить до простою значної частини процесорів.

4.3 Закони Амдала та Густафсона

Оптимістичний погляд на паралельні обчислення дає закон Густафсона-Барсіса. Замість питання про прискорення на n процесорах розглядається питання про уповільнення обчислень при переході на один процесор.

Закон прискорення, що масштабується:

$$S(n) = \frac{T(1)}{T(n)} = \frac{f \times T(n) + n \times (1 - f) \times T(n)}{f \times T(n) + (1 - f) \times T(n)} = n + (1 - n) \times f$$



4.3 Закони Амдала та Густафсона

Працюючи на багатопроцесорних системах, користувачі схильні до зміни тактики вирішення задачі.

Тепер зниження загального часу виконання програми поступається обсягом розв'язуваної задачі.

Така зміна мети обумовлює перехід від закону Амдала до закону Густафсона.

Наприклад, на 100 процесорах програма виконується 20 хвилин. При переході на систему від 1000 процесорів можна досягти часу виконання близько 2 хвилин. Однак для отримання більшої точності рішення має сенс збільшити на порядок обсяг розв'язуваної задачі (наприклад, вирішити систему рівнянь в приватних похідних на більш тонкій сітці). Тобто при збереженні загального часу виконання користувачі прагнуть отримати більш точний результат.

Збільшення обсягу розв'язуваної задачі призводить до збільшення частки паралельної частини, так як послідовна частина (введення/виведення, менеджмент потоків, точки синхронізації і т.п.) не змінюється.

Тема 5. Архітектури МПС з паралельною обробкою даних

5.1 Класифікація паралельних МПС по Флінну:

- ✓ SISD архітектурний клас;
- ✓ SIMD архітектурний клас;
- ✓ MISD архітектурний клас;
- ✓ MIMD архітектурний клас.

5.2 Паралельні МПС із загальною та розподіленою пам'яттю:

- ✓ SMP архітектура;
- ✓ MPP архітектура;
- ✓ NUMA архітектура (гібридна).

5.1 Класифікація паралельних МПС по Флінну

Флінном (1966) запропонований підхід до класифікації архітектур обчислювальних систем. В основу було покладено поняття *поток*, під яким розуміється *послідовність елементів, команд або даних, що обробляється процесором*. Відповідна система класифікації заснована на розгляді числа потоків інструкцій і потоків даних.

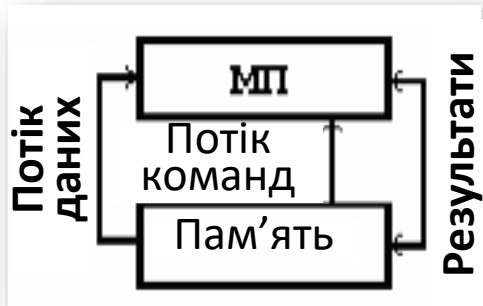
**Чотири
архітектурних
класу:**

SISD = Single Instruction Single Data,
MISD = Multiple Instruction Single Data ,
SIMD = Single Instruction Multiple Data,
MIMD = Multiple Instruction Multiple Data.

SISD – одиночний потік команд і одиночний потік даних.

До цього класу належать послідовні МПС, які мають один центральний процесор, здатний обробляти тільки один потік послідовно виконуваних інструкцій. В даний час практично всі високопродуктивні системи мають більше одного центрального процесора, проте, кожен з них виконують незв'язані потоки інструкцій, що робить такі системи комплексами SIMD-систем, діючих на різних просторах даних.

5.2 Класифікація паралельних МПС по Флінну



SISD – одиночний потік команд і одиночний потік даних

Класичні фоннеймановські машини потрапляють в тривіальний клас SISD, в якому одиночний потік інструкцій обробляє одиночний потік даних.

Класичні мови високого рівня (такі, як C++) також орієнтовані на програмування в класі SISD. В даний час випуск SISD процесорів майже припинений через їх низьку продуктивність, яка обумовлена низьким рівнем паралелізму обчислень (використовується тільки дрібнозернистий паралелізм).

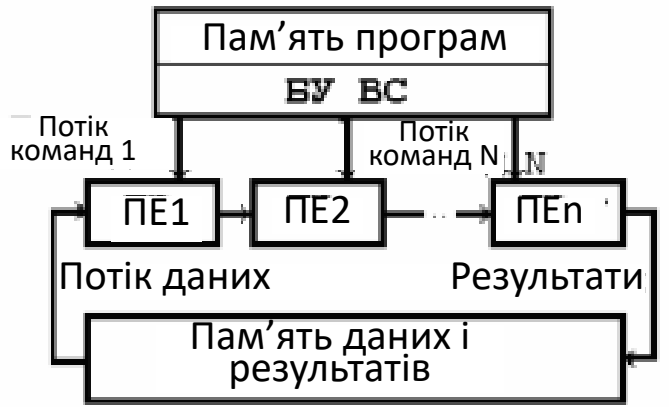
Для збільшення швидкості обробки команд і швидкості виконання арифметичних операцій може застосовуватися конвеєрна обробка.

Прикладами комп'ютерів з архітектурою SISD є більшість робочих станцій Compaq, Hewlett-Packard і Sun Microsystems.

MISD - множинний потік команд і одиночний потік даних.

Теоретично в цьому типі МПС безліч інструкцій має виконуватися над єдиним потоком даних. Визначення має на увазі наявність в архітектурі багатьох процесорів, що обробляють один і той же потік даних. Однак розвиток ця архітектура МПС до сих пір не отримала.

5.2 Класифікація паралельних МПС по Флінну



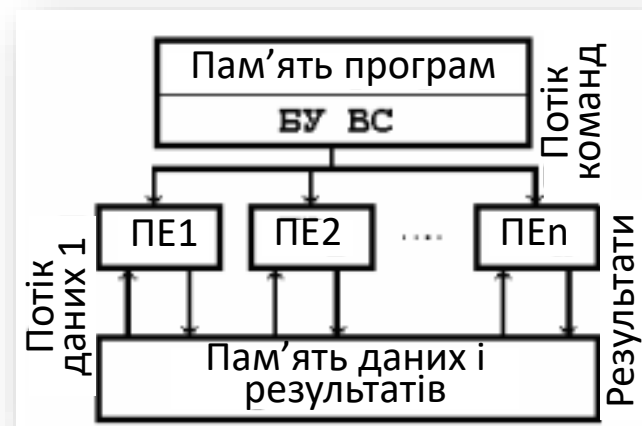
MISD - множинний потік команд і одиночний потік даних

SIMD - одиночний потік команд і множинний потік даних.

Ці системи зазвичай мають велику кількість процесорів (від 1024 до 16384), які можуть виконувати одну і ту ж інструкцію щодо різних даних в жорсткій конфігурації. Єдина інструкція паралельно виконується над багатьма елементами даних.

Прикладами SIMD машин є системи CPP DAP, Gamma II і

Quadrics Aremille. Іншим підкласом SIMD-систем є векторні МПС. Вони маніпулюють масивами подібних даних подібно до того, як скалярні системи обробляють окремі елементи таких масивів. Це робиться за рахунок використання спеціально сконструйованих векторних центральних процесорів. Клас орієнтований на виконання програм, для яких характерна обробка великих регулярних масивів чисел.

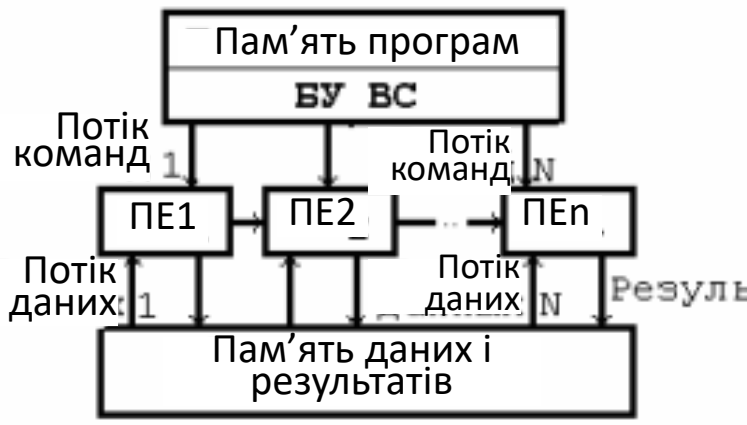


5.2 Класифікація паралельних МПС по Флінну

Саме представники класу SIMD вперше досягли продуктивності порядку GFLOPS. Найбільш популярна ідея класу SIMD - векторне процесування. Векторний процесор підтримує обробку не тільки скалярних, але і векторних операндів. Ефективне декодування інструкцій і зручні дані позначаються на продуктивності вкрай позитивно. Тому векторну обробку впроваджують і в процесори класів SISD і MIMD. Наприклад, SIMD розширення IA32 – технології MMX і SSE.

MIMD - множинний потік команд і множинний потік даних.

Ці системи паралельно виконують декілька потоків інструкцій над різними потоками даних. На відміну від багатопроцесорних SISD-систем команди і дані пов'язані, тому що вони представляють різні частини одного і того ж завдання. Наприклад, MIMD-системи можуть паралельно виконувати безліч підзадач, з метою скорочення часу виконання основного завдання.



Всі сучасні передові процесори, як загального, так і спеціального призначення, потрапляють в клас MIMD. Вони одночасно виконують відразу кілька незалежних потоків інструкцій, апаратно забезпечуючи крупнозернистий паралелізм.

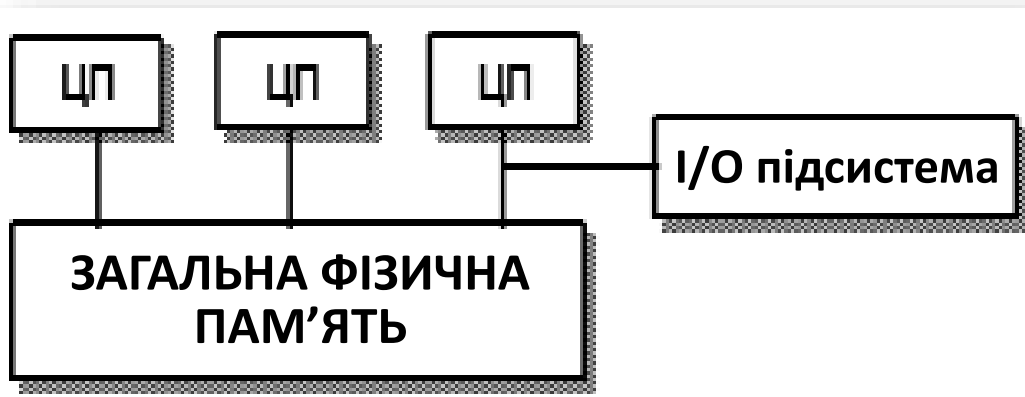
Вада класифікації Флінна полягає в неврахуванні сучасних напрямків процесоробудування, що призвело до виникнення безлічі нових класифікацій.

5.2 SMP архітектура

SMP архітектура (symmetric multiprocessing) – симетрична багатопроцесорна архітектура

Головна особливість SMP – наявність загальної фізичної пам'яті, що розділяється всіма процесорами.

Пам'ять є способом передачі повідомлень між процесорами, при цьому всі обчислювальні пристрої при зверненні до неї мають рівні права і одну і ту ж адресацію для всіх елементів пам'яті (тому SMP архітектура називається симетричною). Остання обставина дозволяє дуже ефективно обмінюватися даними з іншими обчислювальними пристроями



SMP-система будується на основі високошвидкісної системної шини (SGI PowerPath, Sun Gigaplane, DEC TurboLaser), до слотів якої підключаються функціональні блоки трьох типів: процесори (ЦП), операційна система (ОП) і підсистема введення/виведення (I/O). Для під'єднання до модулів I/O використовуються вже більш повільні шини (PCI, VME64).

5.2 SMP архітектура

Найбільш відомими SMP-системами є SMP-сервера і робочі станції на базі процесорів Intel (IBM, HP, Compaq, Dell, ALR, Unisys, DG, Fujitsu і ін.). Вся система працює під керуванням єдиної ОС (зазвичай UNIX-подібної, але для Intel-платформ підтримується Windows NT). ОС автоматично (в процесі роботи) розподіляє процеси по процесорам, але іноді можлива і явна прив'язка.

Основні переваги SMP-систем:

– простота і універсальність для програмування;

Архітектури SMP не накладаються обмежень на модель програмування, використовувану при створенні програми, зазвичай використовується модель паралельних гілок, коли всі процесори працюють абсолютно незалежно один від одного – проте, можна реалізовувати і моделі, що використовують міжпроцесорний обмін. Застосування загальною пам'яті збільшує швидкість такого обміну, користувач також має доступ відразу до всього об'єму пам'яті. Для SMP-систем існують порівняно ефективні засоби автоматичного розпаралелювання.

– легкість в експлуатації;

Як правило, SMP-системи використовують систему охолодження, засновану на повітряному кондиціонуванні, що полегшує їх технічне обслуговування.

– відносно невисока ціна.

5.2 SMP архітектура

Основний недолік SMP-систем:

– системи із загальною пам'яттю, побудовані на системній шині, погано масштабуються.

Цей значний недолік SMP-системи не дозволяє вважати їх по-справжньому перспективними. Причини поганого масштабування полягають у тому, що в даний момент шина здатна обробляти тільки одну транзакцію, внаслідок чого виникають проблеми вирішення конфліктів при одночасному зверненні декількох процесорів до одних і тих же областей загальної фізичної пам'яті. Обчислювальні елементи починають один одному заважати. Коли відбудеться такий конфлікт, залежить від швидкості зв'язку та від кількості обчислювальних елементів. На теперішній час конфлікти можуть відбуватися при наявності 8-24 процесорів. Крім того, системна шина має обмежену (хоч і високу) пропускну здатність і обмежене число слотів. Все це з очевидністю перешкоджає збільшенню продуктивності при збільшенні числа процесорів і числа підключених користувачів.

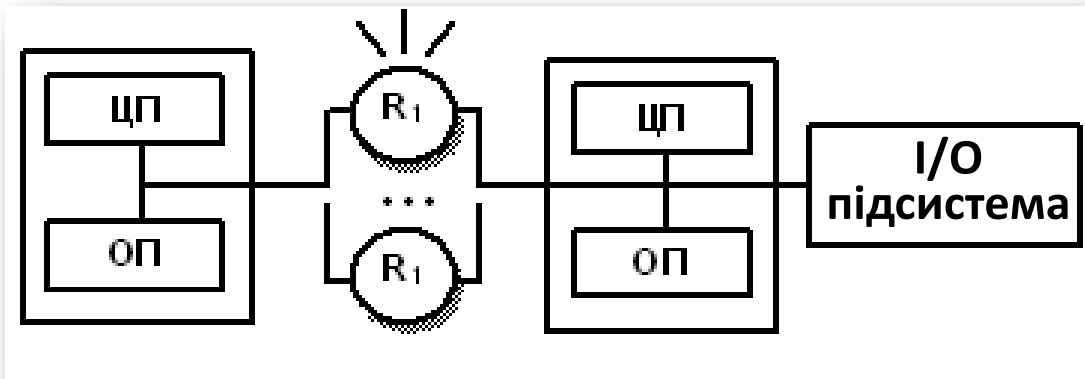
У реальних системах не більше 32 процесорів. Для побудови масштабованих систем на базі SMP використовуються кластерні або NUMA-архітектури.

5.2 MPP архітектура

MPP архітектура (massive parallel processing) - масивно-паралельна архітектура.

Головна особливість MPP архітектури полягає в тому, що пам'ять фізично розділена.

В цьому випадку система будується з окремих модулів, що містять процесор (ЦП), локальний банк операційної пам'яті (ОП), два комунікаційних процесора (рутера) або мережевий адаптер, іноді – жорсткі диски і/або інші пристрої введення/виведення. Один рутер використовується для передачі команд, інший – для передачі даних. Такі модулі представляють собою повнофункціональні системи.



Доступ до банку ОП з даного модуля мають тільки ЦП з цього ж модуля. Модулі з'єднуються спеціальними комунікаційними каналами. Користувач може виділити логічний номер ЦП, до якого він підключений, і організувати обмін повідомленнями з іншими ЦП.

5.2 MPP архітектура

Використовуються два варіанти роботи ОС на машинах MPP:

1 – повноцінна ОС працює тільки на керуючій машині (Front-end), на кожному окремому модулі працює сильно скорочений варіант ОС, що забезпечує роботу тільки розташованої в ньому гілки паралельного додатка.

2 – на кожному модулі працює повноцінна UNIX-подібна ОС, яка встановлюється окремо на кожному модулі.

Основна перевага MPP-систем:

– хороша масштабованість;

На відміну від SMP-систем в машинах з роздільною пам'яттю кожен процесор має доступ тільки до своєї локальної пам'яті, у зв'язку з чим не виникає необхідності в потактовій синхронізації процесорів. Практично всі рекорди по продуктивності на сьогоднішній день встановлені на системах саме такої архітектури, які містять декілька тисяч процесорів (ASCI Red, ASCI Blue Pacific).

5.2 MPP архітектура

Основні недоліки MPP-систем:

– відсутність загальної пам'яті помітно знижує швидкість міжпроцесорного обміну, оскільки немає загального середовища для зберігання даних, призначених для обміну між процесорами;

Потрібна спеціальна техніка програмування для реалізації обміну повідомленнями між процесорами.

– кожен процесор може використовувати тільки обмежений обсяг локального банку пам'яті;

– вимагаються значні зусилля для того, щоб максимально використовувати системні ресурси;

Внаслідок зазначених архітектурних особливостей визначається висока ціна ПЗ для масивно-паралельних систем з роздільною пам'яттю.

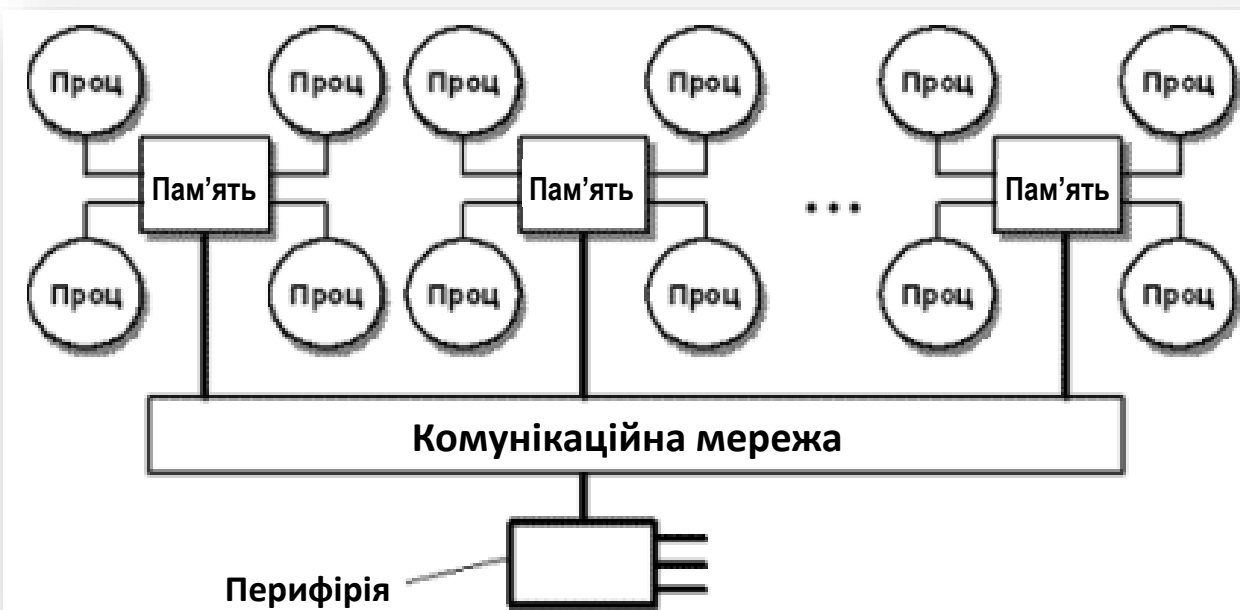
Системами з роздільною пам'яттю є суперкомп'ютери MBC-1000, IBM RS / 6000 SP, SGI / CRAY T3E, системи ASCI, Hitachi SR8000, системи Parsytec.

Системи останньої серії CRAY T3E від SGI, засновані на базі процесорів Dec Alpha 21164 з піковою продуктивністю 1200 МФлопс/с (CRAY T3E-1200), здатні масштабуватися до 2048 процесорів.

5.2 NUMA архітектура (гібридна)

Гібридна архітектура (NUMA - nonuniform memory access) характеризується неоднорідним доступом до пам'яті.

Втілює в собі зручності систем зі спільною пам'яттю і відносно дешевизну систем з роздільною пам'яттю.



Структурна схема системи NUMA (один із варіантів): чотири процесори, які пов'язуються між собою за допомогою кросбару в рамках одного SMP вузла. Вузли зв'язані мережею типу "метелик" (Butterfly). Пам'ять фізично розподілена по різних частинам системи, але логічно розділяється, так що користувач бачить єдиний адресний простір.

Система складається з однорідних базових модулів (плат), що складаються з невеликого числа процесорів і блоку пам'яті. Модулі об'єднані за допомогою високошвидкісного комутатора. Підтримується єдиний адресний простір, апаратно підтримується доступ до віддаленої пам'яті, тобто до пам'яті інших модулів.

5.2 NUMA архітектура (гібридна)

При цьому доступ до локальної пам'яті здійснюється в кілька разів швидше, ніж до віддаленої. По суті архітектура NUMA є MPP архітектурою, де в якості окремих обчислювальних елементів беруться SMP вузли.

Різновид архітектури cc-NUMA (Cache Coherent Non-Uniform Memory Access) – NUMA архітектура з неоднорідним доступом до пам'яті з забезпеченням когерентності кешей.

Поняття когерентності кешей описує той факт, що всі ЦП отримують однакові значення одних і тих же змінних в будь-який момент часу.

Оскільки кеш-пам'ять належить окремій системі, а не всій багатопроцесорній системі в цілому, дані, що потрапляють в кеш одній МПС, можуть бути недоступні іншій.

Слід проводити синхронізацію інформації, що зберігається в кеш-пам'яті ЦП.

Для забезпечення такої когерентності кешей існують кілька можливостей:

- використовувати механізм вистежування шинних запитів (Snoopy bus protocol), в якому кеші відстежують змінні, які передаються до будь-якого з ЦП і, при необхідності, модифікують власні копії таких змінних;
- виділяти спеціальну частину пам'яті, що відповідає за вистежування достовірності всіх використовуваних копій змінних.

Найбільш відомими системами архітектури cc-NUMA є HP 9000 V-class в SCA-конфігураціях, SGI Origin3000, Sun HPC 15000, IBM / Sequent NUMA-Q 2000. На зараз максимальне число процесорів в cc-NUMA-системах може перевищувати 1000 (серія Origin3000). Зазвичай вся система працює під управлінням єдиної ОС, як в SMP.

Тема 6. Розподілені мікроконтролерні системи на базі комунікаційних інтерфейсів RS-232, RS-422 та RS-485

- 6.1 Загальна характеристика інтерфейсу RS-232.
- 6.2 Властивості інтерфейсу RS-232.
- 6.3 Практична реалізація інтерфейсу RS-232.
- 6.4 Загальна характеристика інтерфейсів RS-422 і RS-485.
- 6.5 Властивості інтерфейсу RS-485.
- 6.6 Приймачі-передавачі інтерфейсів RS-422 і RS-485.
- 6.7 Стандартні параметри інтерфейсів RS-422 і RS-485.

6.1 Загальна характеристика інтерфейсу RS-232

Найбільш поширений послідовний інтерфейс. Розроблені велика кількість компонентів, елементної бази, програмних модулів.

Популярність інтерфейсу **RS-232** визначається невисокою вартістю портів і кабелів, стабільністю використання даного стандарту протягом довгого часу різними виробниками.

Підтримується асинхронний і синхронний режими обміну.

Інтерфейс **RS-232** побудований на уніполярних лініях передачі даних. Тому його продуктивність і максимальна довжина кабелю невеликі. RS-232 використовується для підключення периферійного обладнання до ПК. Кабельне улаштування недороге, що дає можливість застосовувати окремі лінії для квітування, синхронізації і переривання.

RS-232 – радіальний інтерфейс, тому поняття адреси в ньому відсутнє.

Застосування інтерфейсу **RS-232** в нових розробках продовжується в наслідок:

- ✓ **дешевизни кабельного улаштування і практичної реалізації портів;**
- ✓ **великого парку працюючого обладнання, який застосовує цей стандарт;**
- ✓ **можливість організації гальванічної розв'язки.**

6.2 Властивості інтерфейсу RS-232

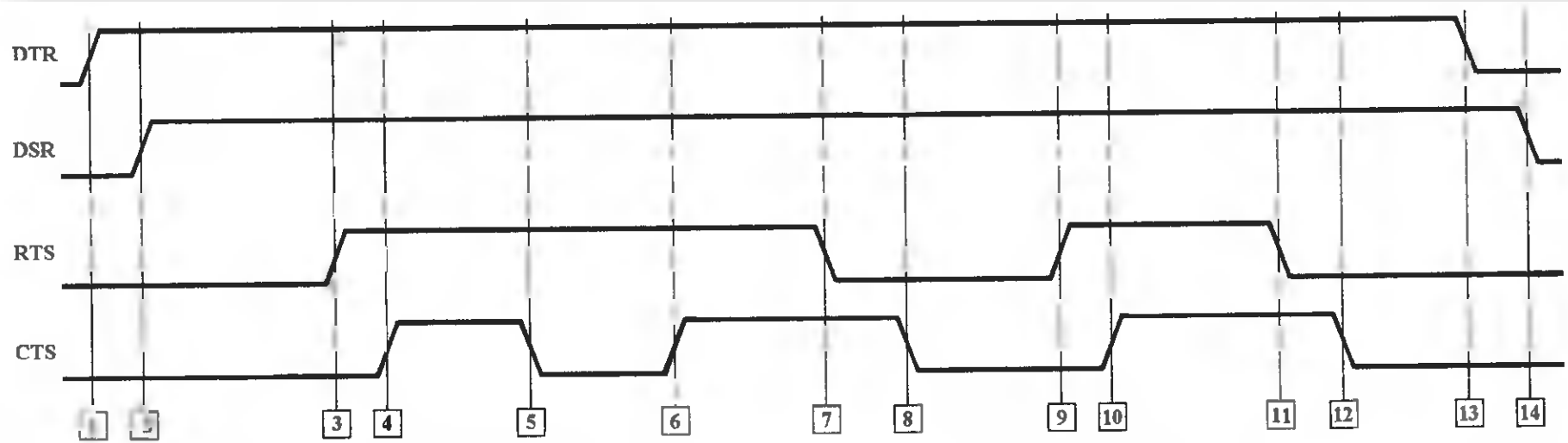
Найменування	Призначення
PG (Protected Ground)	Захисна земля
SG (Signal Ground)	Сигнальна земля
TD (Transmit Data)	Дані. Від контролера до кінцевого пристрою
RD (Receive Data)	Дані. Від кінцевого пристрою до контролера
RTS (Request To Send)	Запит передачі даних. Сигнал готовності даних для передачі від контролера до кінцевого пристрою. В напівдуплексному режимі застосовується для управління напрямом передачі даних.
CTS (Clear To Send)	Готовність регістра прийому кінцевого пристрою
DSR (Data Set Ready)	Готовність кінцевого пристрою до обміну даними
DTR (Data Terminal Ready)	Готовність контролера до обміну даними
DCD (Data Carrier Detected)	Готовність віддаленої апаратури кінцевого пристрою до обміну даними
RI (Ring Indicator)	Переривання від кінцевого пристрою

Призначення сигналів RS-232

Послідовність керувальних сигналів RS-232:

1. Встановленням DTR контролер вказує на потребу використати кінцевий пристрій.
2. Встановленням DSR кінцевий пристрій сигналізує про свою готовність до роботи.
3. Встановленням RTS контролер запитує дозвіл на передачу і заявляє про свою готовність приймати дані від кінцевого пристрою.
4. Встановленням CTS кінцевий пристрій інформує про свою готовність до приймання даних.
5. Зняттям CTS кінцевий пристрій сигналізує про неможливість подальшого приймання (наприклад, буфер прийому повний) – контролер повинен призупинити передачу даних.
6. Встановленням CTS кінцевий пристрій дозволяє комп'ютеру продовжити передачу (наприклад, в буфері з'явилося місце).
7. Зняттям RTS контролер інформує кінцевий пристрій про свою неготовність до обміну даними. Це може означати як заповнення буфера прийому даних контролера, так і відсутність даних для передачі в кінцевий пристрій

6.2 Властивості інтерфейсу RS-232

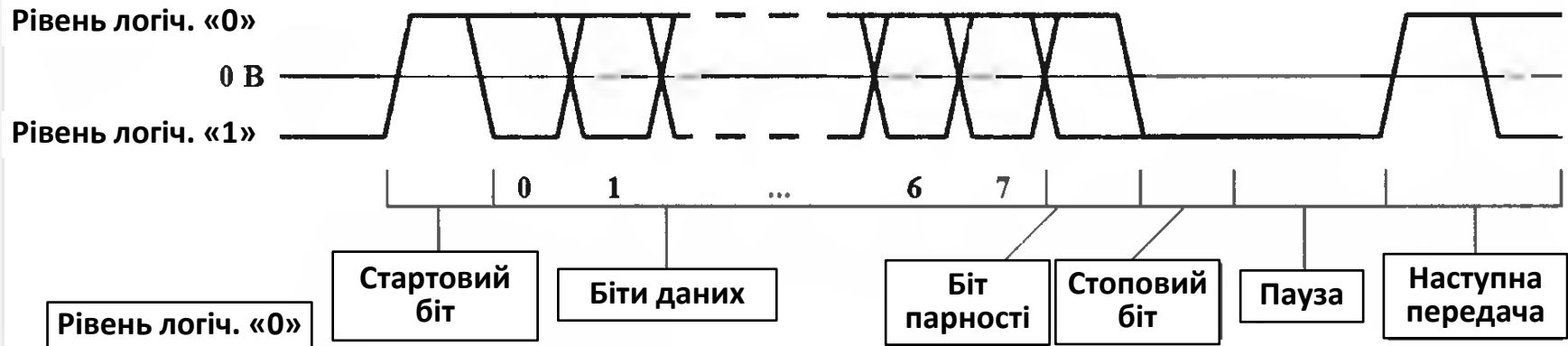


8. Кінцевий пристрій підтверджує зняття RTS зняттям CTS.
9. Контролер повторно встановлює RTS для продовження обміну даними.
10. Кінцевий пристрій підтверджує готовність до обміну встановленням CTS.
11. Зняттям RTS контролер вказує на завершення обміну.
12. Кінцевий пристрій підтверджує зняття RTS зняттям CTS.
13. Контролер знімає DTR для переведення кінцевого пристрою в автономний або «сплячий» режим.
14. Кінцевий пристрій підтверджує зняття DTR зняттям DSR.

В лініях обміну даними у проміжках 4-5, 6-7 та 10-11 циркулюють послідовні дані, які передаються **побайтно**.

Для синхронізації приймача з передавачем і відділення байтів один від одного в потік вставляються стартові і стопові біти.

6.2 Властивості інтерфейсу RS-232



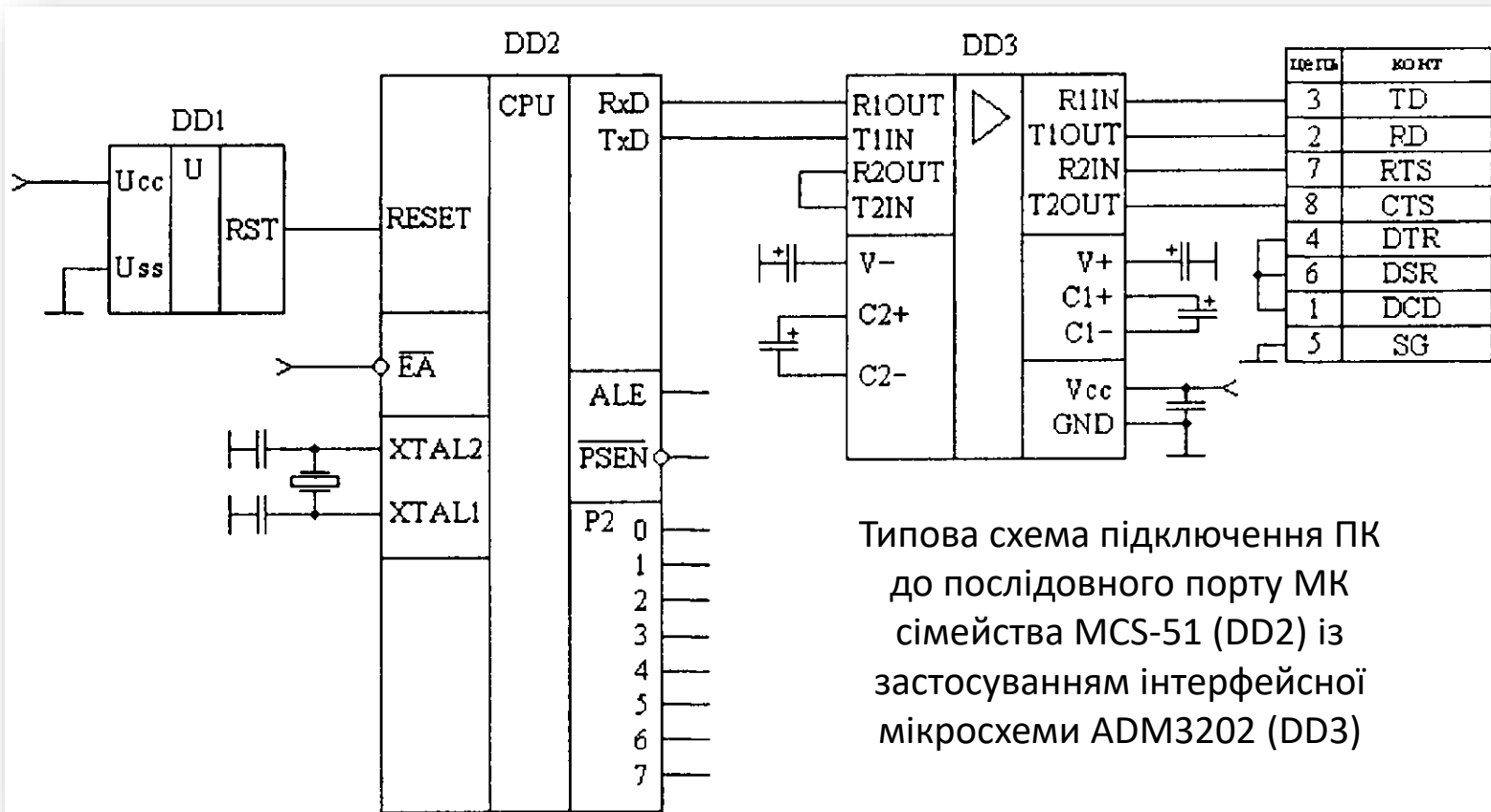
Коли лінія вільна, на ній присутній рівень лог. 1. Перед байтом даних передається стартовий біт – рівень лог. 0. Виявивши його, приймач починає приймання байту даних. Після даних передається біт парності, потім один або декілька стопових біт. Стопові біти передаються рівнями лог. 1 і слугують для маркування кінця байту даних. Після них може слідувати пауза невизначеної тривалістю або початок нової передачі.

Логічні рівні інтерфейсу RS-232:

	Рівень логічного «0»	Рівень логічної «1»
Передавач	от +5 до +12 В	от –12 до –5 В
Приймач	от +3 до +12 В	от –12 до –3 В

6.3 Практична реалізація інтерфейсу RS-232

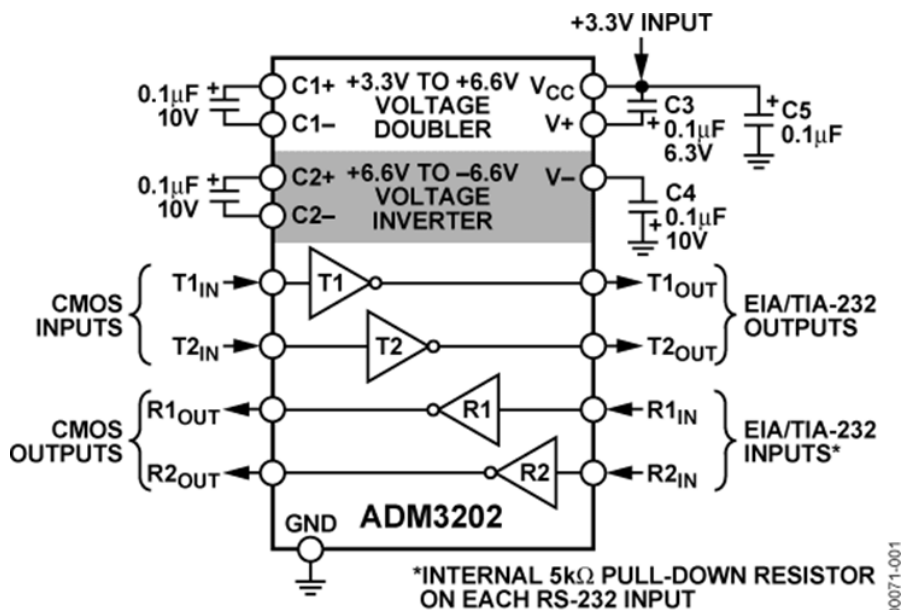
Можливість роботи МК в асинхронному режимі дозволяє використовувати послідовний порт для зв'язку з універсальним ПК через його послідовний COM-порт. Рівні сигналів послідовного порту МК не відповідають специфікаціям стандартного інтерфейсу RS-232, використовуваного в COM-порте універсальних ПК, тому для підключення доводиться додатково використовувати спеціалізовані мікросхеми узгодження рівнів. Ці ж мікросхеми забезпечують захист МК від виведення з ладу статичним потенціалом при підключенні роз'ємів.



6.3 Практична реалізація інтерфейсу RS-232

Зазвичай для обміну інформацією використовуються тільки сигнальні ланцюги COM-порту комп'ютера. Буфери інтерфейсної мікросхеми, які залишилися, можуть бути використані для контролю живлення мікроконтролерної схеми. Для цього сигнал запиту готовності віддаленого пристрою від ПК пропускається через інтерфейсну мікросхему. Потім цей же сигнал подається на вхід підтвердження готовності віддаленого пристрою. Якщо на інтерфейсну мікросхему не буде подано живлення, то сигнал підтвердження готовності не буде сформований.

Використання послідовного порту ПК дозволяє не тільки управляти мікроконтролерним пристроєм, використовуючи клавіатуру ПК, але і відображати внутрішню інформацію цього пристрою, використовуючи дисплей ПК. Це значно розширює можливості введення та виведення інформації в мікроконтролерних пристроях. Останнім часом додатково з'явилася можливість заносити програму у внутрішню пам'ять програм найбільш сучасних МК.



Основні властивості і переваги застосування прийомопередавача ADM3202:

швидкість передачі даних 460 Кбіт/с; характеристики гарантовано забезпечуються при 3.3 В; задовольняє вимогам специфікації EIA-232E; конденсатори накачування заряду 0.1 мкФ; заміщення MAX3222/32 і LTC1385; режим відключення з низьким енергоспоживанням (ADM3222E і ADM1385); корпуси DIP, SO, SOIC, SSOP і TSSOP; захист від електростатичного розряду відповідно до IEC1000-4-2 (801.2) за висновками RS-232 (тільки в ADM3202): ± 8 кВ: контактний розряд, ± 15 кВ: безконтактний розряд

6.4 Загальна характеристика інтерфейсів RS-422 і RS-485

Поширені послідовні інтерфейси. Розроблені велика кількість компонентів, елементної бази, програмних модулів.

Популярність інтерфейсу **RS-422** і **RS-485** визначається невисокою вартістю портів і кабелів, стабільністю використання даного стандарту протягом довгого часу різними виробниками.

Підтримується асинхронний і синхронний режими обміну.

RS-485 – це номер стандарту, вперше прийнятого Асоціацією електронної промисловості (EIA). Зараз цей стандарт - TIA / EIA-485 Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems (Електричні характеристики передавачів і приймачів, що використовуються в балансних цифрових багатоточкових системах).

Стандарт фізичного рівня зв'язку. Фізичний рівень – це канал зв'язку і спосіб передачі сигналу (1 рівень моделі взаємозв'язку відкритих систем OSI).

Інтерфейс **RS-485** забезпечує обмін даними між декількома пристроями по **одній двохпроводній лінії зв'язку в напівдуплексному режимі**. Широко використовується в промисловості при створенні АСУ ТП.

Головна відмінність RS-485 від широко поширеного RS-232 – можливість об'єднання декількох пристроїв.

6.5 Властивості інтерфейсу RS-485

Швидкість і дальність. RS-485 забезпечує передачу даних зі швидкістю до 10 Мбіт/с. Максимальна дальність залежить від швидкості: при швидкості 10 Мбіт/с максимальна довжина лінії – 120 м, при швидкості 100 Кбіт/с – 1200 м.

Кількість пристроїв, що з'єднуються. Кількість пристроїв, що підключаються до однієї лінії інтерфейсу, залежить від типу застосованих в пристрої приймачів. Один передавач розрахований на управління 32 стандартними приймачами. Випускаються приймачі з вхідним опором 1/2, 1/4, 1/8 від стандартного. При використанні таких приймачів загальне число пристроїв може бути збільшено відповідно: 64, 128 або 256.

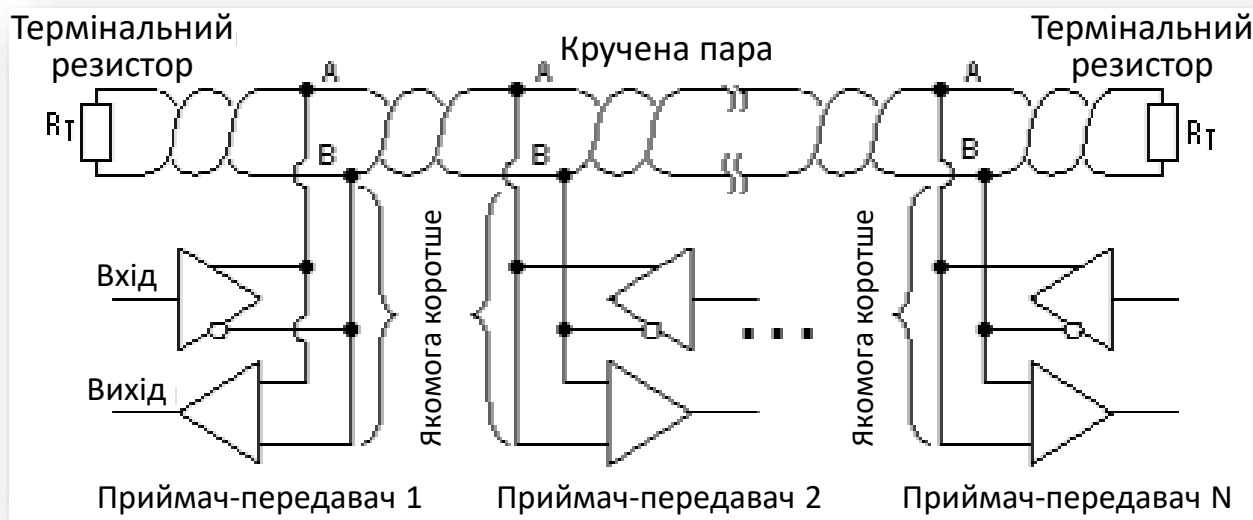


Схема підключення. Локальна мережа на основі інтерфейсу RS-485, що об'єднує кілька приймачів-передавачів. При підключенні слід правильно приєднати сигнальні ланцюги, зазвичай звані А і В. Переполюсовка не страшна, але пристрій не працюватиме.

6.5 Властивості інтерфейсу RS-485

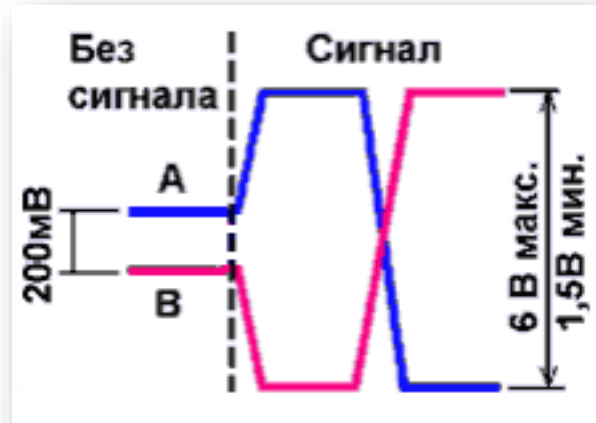
Протоколи і роз'єми. Стандарт не нормує формат інформаційних кадрів і протокол обміну. Найбільш часто для передачі байтів даних використовуються ті ж фрейми, що і в інтерфейсі RS-232: стартовий біт, біти даних, біт паритету (якщо потрібно), стоповий біт. Протоколи обміну в більшості систем працюють за принципом "ведучий-ведений". Один пристрій на магістралі є провідним (master) і ініціює обмін посилкою запитів підлеглим пристроїв (slave), які розрізняються логічними адресами. Одним з популярних протоколів є протокол Modbus RTU. Тип з'єднувачів і розпаювання також не обумовлюються стандартом. Зустрічаються з'єднувачі DB9, клемні з'єднувачі і т.д.

Загальні рекомендації.

- ✓ Кращим середовищем передачі сигналу є кабель на основі крученої пари.
- ✓ Кінці кабелю мають бути заглушені термінальними резисторами (зазвичай 120 Ом).
- ✓ Мережа повинна бути прокладена по топології шини, без відгалужень.
- ✓ Пристрої слід підключати до кабелю проводами мінімальної довжини.
- ✓ Кручена пара є оптимальним рішенням для прокладки мережі, оскільки володіє найменшим паразитним випромінюванням сигналу і добре захищена від наведень. В умовах підвищених зовнішніх перешкод застосовують кабелі з екранованою крученою парою, при цьому екран кабелю з'єднують із захисною "землею" пристрою.

Рівні сигналів. Інтерфейс **RS-485** використовує балансну (диференціальну) схему передачі сигналу. Це означає, що рівні напруги на сигнальних ланцюгах А і В міняються в протифазі. Передавач повинен забезпечувати рівень сигналу 1,5 В при максимальному навантаженні (32 стандартних входи і 2 термінальних резистора) і не більше 6 В на холостому ході. Рівні напруги вимірюють диференційно, один сигнальний провід щодо іншого. На стороні приймача **RS-485** мінімальний рівень сигналу повинен бути не менше 200 мВ.

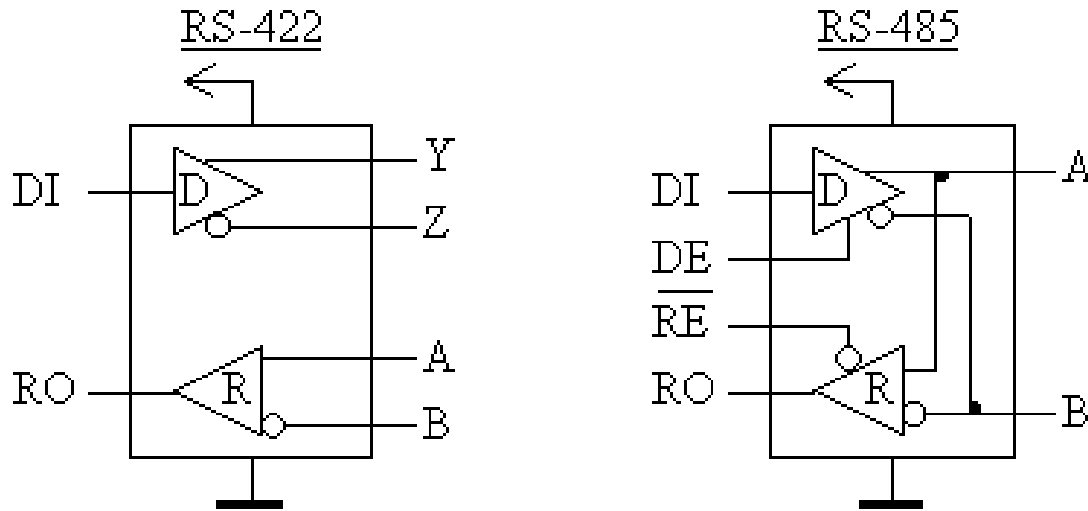
6.5 Властивості інтерфейсу RS-485



RS-485 – напівдуплексний інтерфейс. Прийом і передача йдуть по одній парі проводів з поділом за часом. У мережі може бути багато передавачів, так як вони можуть відключатися в режимі прийому.

*Передача одного сигналу по двох проводах. По одному дроту (A) йде оригінальний сигнал, а по іншому (B) – його інверсна копія. Іншими словами, якщо на одному дроті "1", то на іншому "0" і навпаки. Таким чином, між двома проводами крученої пари завжди є різниця потенціалів: при "1" вона позитивна, при "0" - негативна. Саме цією різницею потенціалів і передається сигнал. Такий спосіб передачі забезпечує високу стійкість до синфазної перешкоди. **Синфазною називають перешкоду, що діє на обидва дроти лінії однаково.** Наприклад, електромагнітна хвиля, проходячи через ділянку лінії зв'язку, наводить в обох проводах потенціал. Якщо сигнал передається потенціалом в одному дроті щодо загального, як в RS-232, то наводка на цей провід може спотворити сигнал відносно добре поглинаючого наведення загального ("земля") проводу. Крім того, на опорі довгого загального проводу буде падати різниця потенціалів земель – додаткове джерело спотворень. А при диференціальній передачі спотворення не відбувається. Якщо два дроти пролягають близько один до одного, та ще перевиті, то наводка на обидва дроти однакова. Потенціал в обох однаково навантажених проводах змінюється однаково, при цьому інформативна різниця потенціалів залишається без змін.*

6.6 Приймачі-передавачі інтерфейсів RS-422 і RS-485



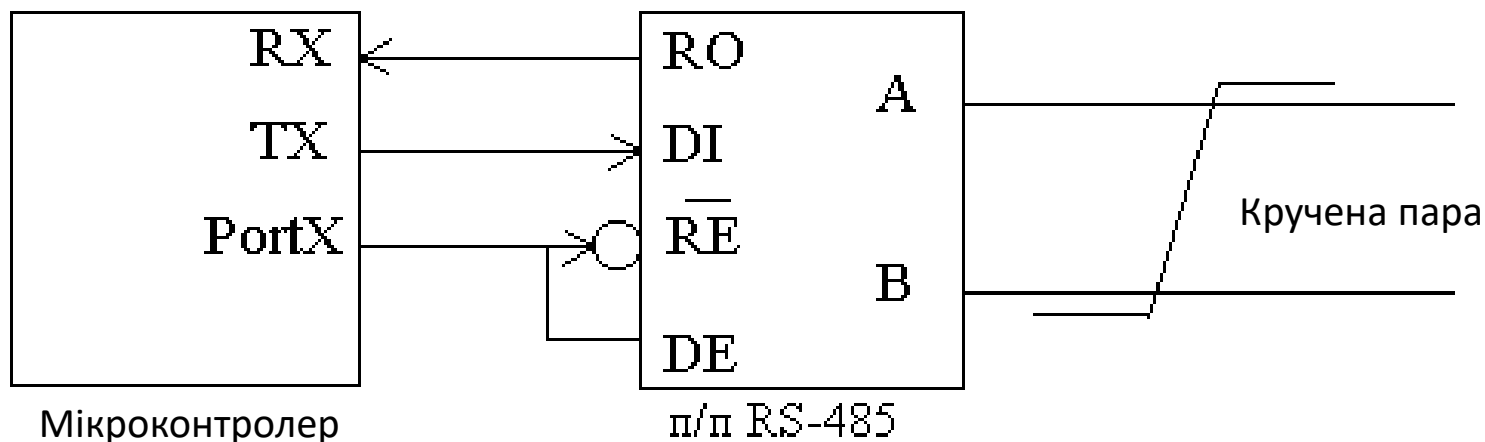
RS-422 – повнодуплексний інтерфейс. Прийом і передача йдуть по двом окремим парам проводів. На кожній парі проводів може бути тільки по одному передавачу.

RS-485 – напівдуплексний інтерфейс. Прийом і передача йдуть по одній парі проводів з поділом за часом. У мережі може бути багато передавачів, так як вони можуть відключатися в режимі прийому.

D (driver) - передавач; R (receiver) - приймач; DI (driver input) - цифровий вхід передавача; RO (receiver output) - цифровий вихід приймача; DE (driver enable) - дозвіл роботи передавача; RE (receiver enable) - дозвіл роботи приймача; A - прямий диференціальний вхід / вихід; B - інверсний диференціальний вхід / вихід; Y - прямий диференціальний вихід (RS-422); Z - інверсний диференціальний вихід (RS-422).

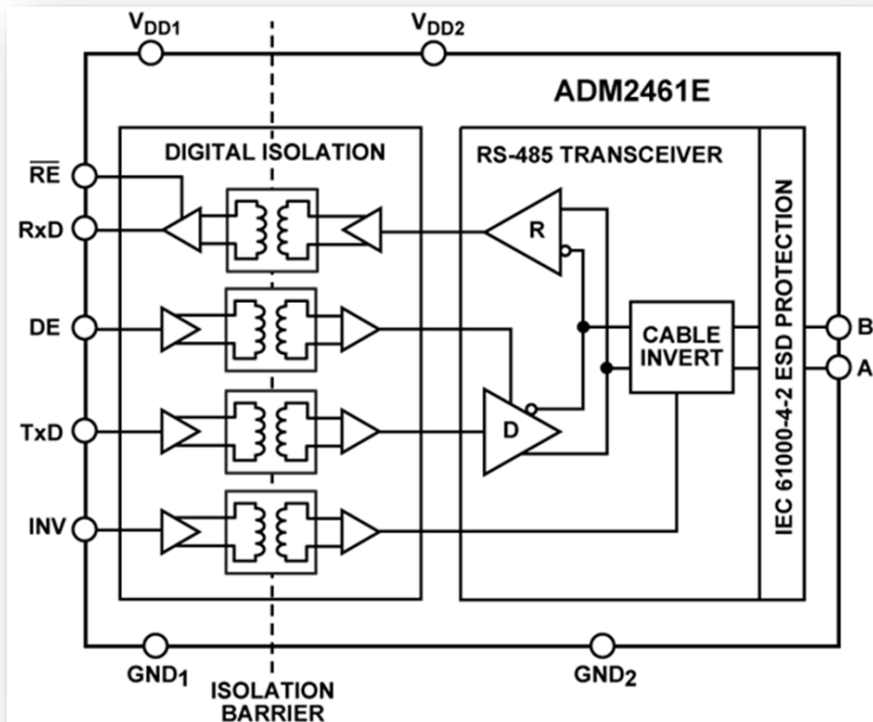
Приймачі-передавачі RS-485. Цифровий вихід приймача (RO) підключається до порту приймача UART (RX). Цифровий вхід передавача (DI) до порту передавача UART (TX). Оскільки на диференціальній стороні приймач і передавач з'єднані, то під час прийому потрібно відключати передавач, а під час передачі - приймач. Для цього служать керуючі входи - дозвіл приймача (RE) і дозвіл передавача (DE). Так як вхід RE інверсний, то його можна з'єднати з DE і переключати приймач і передавач одним сигналом з будь-якого порту МК. При рівні "0" - робота на прийом, при "1" - на передачу.

6.6 Приймач-передавачі інтерфейсів RS-422 і RS-485

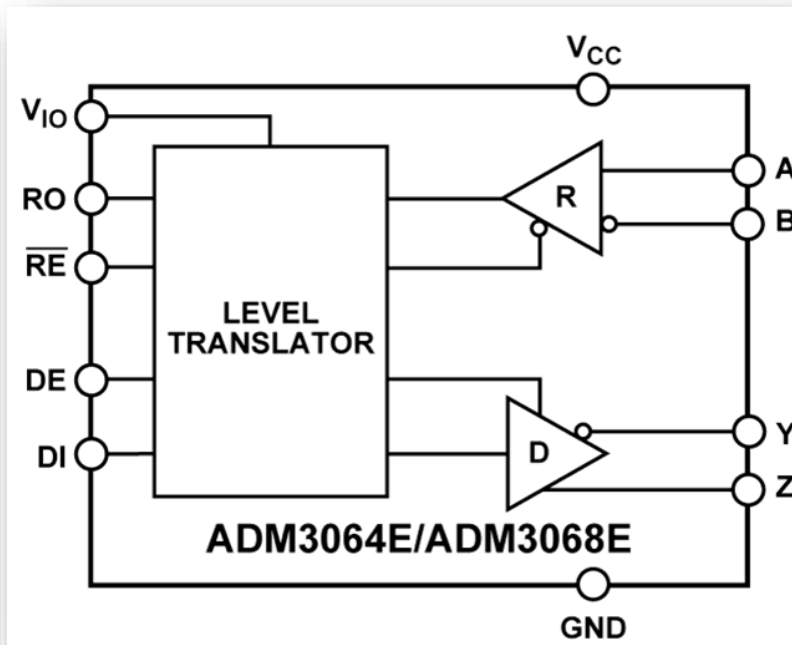


Приймач, отримуючи на диференційних входах (AB) різницю потенціалів (U_{AB}) переводить їх у цифровий сигнал на виході RO. Чутливість приймача може бути різною, але гарантований пороговий діапазон розпізнавання сигналу виробники мікросхем приймачів пишуть в документації. Зазвичай ці пороги становлять ± 200 мВ. Тобто, коли $U_{AB} > +200$ мВ – приймач визначає "1", коли $U_{AB} \leq 200$ мВ – приймач визначає "0". Якщо різниця потенціалів в лінії настільки мала, що не виходить за порогові значення – правильне розпізнавання сигналу не гарантовано.

6.6 Приймач-передавачі інтерфейсів RS-422 і RS-485



ADM2461E. 500 kbps, 5.7 kV RMS, Signal Isolated Half Duplex **RS-485** Transceiver with ± 15 kV IEC ESD



ADM3064E. Трансивери **RS-422** із захистом ліній введення/ виведення від електростатичного розряду ± 12 кВ за стандартом IEC, швидкодія 500 Кбіт/с, живлення 3,0-5,5 В

6.7 Стандартні параметри інтерфейсів RS-422 і RS-485

Стандартні параметри інтерфейсів	RS-422	RS-485
Допустиме число передавачів / приймачів	1 / 10	32 / 32
Максимальна довжина кабеля	1200 м	1200 м
Максимальна швидкість зв'язку	10 Мбит/с	10 Мбит/с
Діапазон напруги «1» передавача	+2...+10 В	+1.5...+6 В
Діапазон напруги «0» передавача	-2...-10 В	-1.5...-6 В
Діапазон синфазної напруги передавача	-3...+3 В	-1...+3 В
Допустимий діапазон напруги приймача	-7...+7 В	-7...+12 В
Пороговий діапазон чутливості приймача	±200 мВ	±200 мВ
Максимальний струм короткого замикання драйвера	150 мА	250 мА
Допустимий опір навантаження передавача	100 Ом	54 Ом
Вхідний опір приймача	4 кОм	12 кОм
Максимальний час наростання сигналу передавача	10% бита	30% бита

Тема 7. Мультимікроконтролерні системи на базі інтерфейсів SPI та I²C

7.1 Загальна характеристика інтерфейсу SPI.

7.2 Підключення ІС за допомогою інтерфейсу SPI:

- ✓ радіальне підключення;
- ✓ незалежне підключення;
- ✓ каскадне підключення.

7.3 Протокол передачі SPI інтерфейсу:

- ✓ режими інтерфейсу;
- ✓ похідні і сумісні протоколи.

7.4 Загальна характеристика інтерфейсу I²C.

7.5 Підключення ІС за допомогою інтерфейсу I²C:

7.6 Протокол обміну сигналами в інтерфейсі I²C:

- ✓ формування СТАРТ і СТОП станів;
- ✓ порядок передачі даних по шині;
- ✓ механізм синхронізації обміну сигналами;
- ✓ адресація пристроїв по шині.

7.7 Розширення інтерфейсу I²C.

7.1 Загальна характеристика інтерфейсу SPI

SPI (3-wire) – популярний інтерфейс для послідовного обміну даними між ІС.

Інтерфейс SPI, поряд з I²C, відноситься до найбільш широко-використовуваних інтерфейсів для з'єднання ІС. Спочатку він був придуманий компанією Motorola, а в даний час використовується в продукції багатьох виробників. Його найменування є аббревіатурою від "*Serial Peripheral Bus*", що відображає **його призначення - шина для підключення зовнішніх пристроїв**.

Шина SPI організована за принципом "провідний-підлеглий". В якості ведучого шини зазвичай виступає МК, але їм також може бути програмована логіка, DSP-контролер або спеціалізована ІС. Підключення до ведучого шини зовнішні пристрої утворюють підлеглих шини. В їх ролі виступають різного роду ІС, в тому числі запам'ятовуючі пристрої (EEPROM, Flash-пам'ять, SRAM), годинник реального часу (RTC), АЦП/ЦАП, цифрові потенціометри, спеціалізовані контролери та ін.

Головним складовим блоком інтерфейсу SPI є **звичайний зсувний регістр**, сигнали синхронізації і введення/виведення бітового потоку якого і утворюють інтерфейсні сигнали.

7.1 Загальна характеристика інтерфейсу SPI

Протокол SPI правильніше назвати не протоколом передачі даних, а **протоколом обміну даними між двома зсувними регістрами**, кожен з яких одночасно виконує і функцію приймача, і функцію передавача. Неодмінною умовою передачі даних по шині SPI є генерація сигналу синхронізації шини. Цей сигнал має право генерувати тільки ведучий шини і від цього сигналу повністю залежить робота підлеглого шини.

Існує три типи підключення до шини SPI, в кожному з яких беруть участь чотири сигналу (їх основне і альтернативні позначення див. в табл. 1).

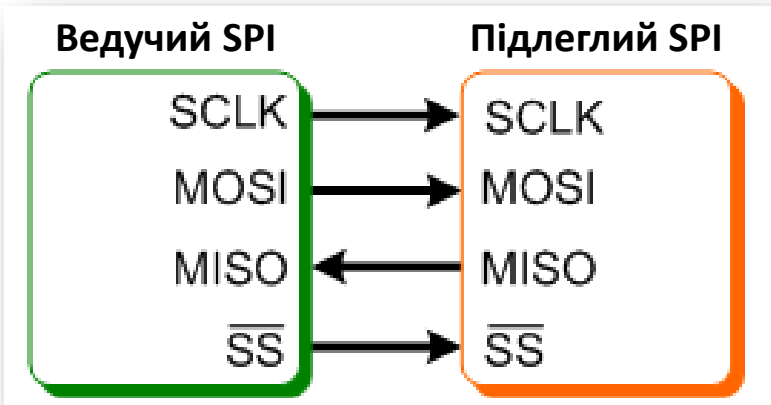
Таблиця 1 – Електричні сигнали шини SPI

Ведучий шини			Підлеглий шини		
Основне позначення	Альтернативне позначення	Опис	Основне позначення	Альтернативне позначення	Опис
MOSI	DO, SDO, DOUT	Вихід послідовної передачі даних	MOSI	DI, SDI, DIN	Вхід послідовного прийому даних
MISO	DI, SDI, DIN	Вхід послідовного прийому даних	MISO	DO, SDO, DOUT	Вихід послідовної передачі даних
SCLK	DCLOCK, CLK, SCK	Вихід синхронізації передачі даних	SCLK	DCLOCK, CLK, SCK	Вхід синхронізації прийому даних
SS	CS	Вихід вибору підлеглого (вибір мікросхеми)	SS	CS	Вхід вибору підлеглого (вибір мікросхеми)

7.2 Підключення ІС за допомогою інтерфейсу SPI

Найпростіше підключення, в якому беруть участь тільки дві ІС, показано на рисунку.

Провідний шини передає дані по лінії MOSI синхронно зі згенерованим їм же сигналом SCLK, а підлеглий захоплює передані біти даних за певними фронтами прийнятого сигналу синхронізації. Одночасно з цим підлеглий відправляє свою послідовність даних. Представлену схему можна спростити виключенням лінії MISO, якщо використовується підпорядкована ІС не передбачає відповідну передачу даних або в ній немає потреби. Односторонню передачу даних можна зустріти у таких ІС як ЦАП, цифрові потенціометри, програмовані підсилювачі і драйвери. **Розглянутий варіант підключення підпорядкованої ІС вимагає 3 або 4 ліній зв'язку.**



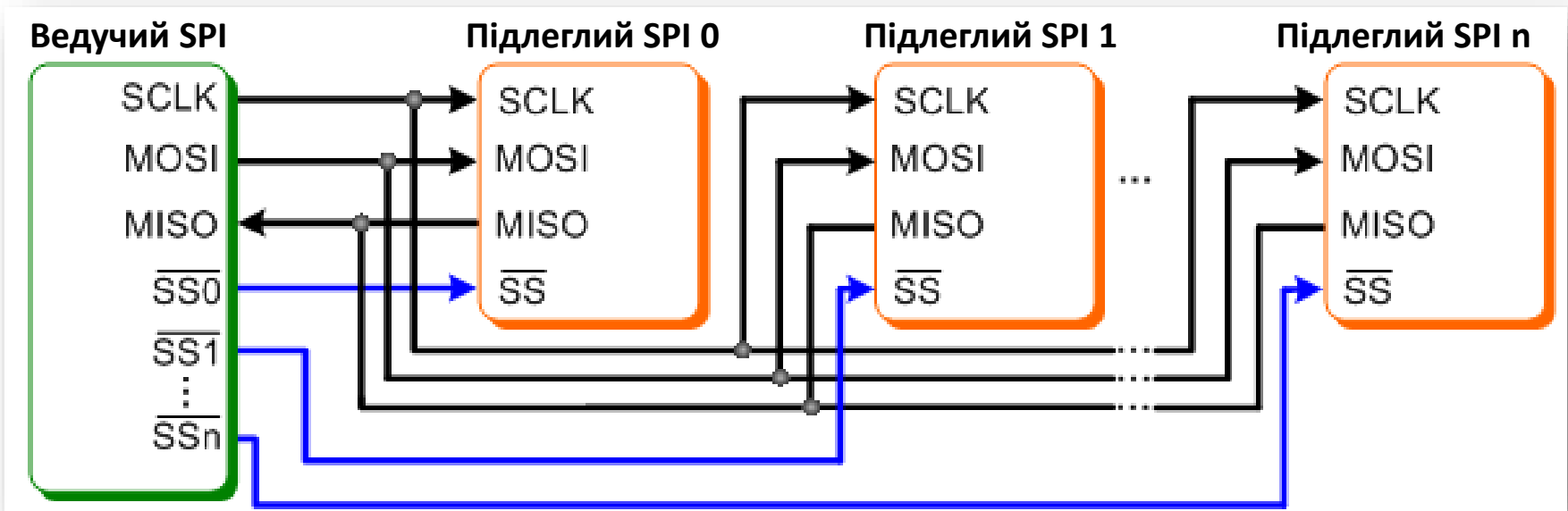
Щоб підпорядкована ІС приймала і передавала дані, крім наявності сигналу синхронізації, необхідно також, щоб лінія SS була переведена в низький стан. В іншому випадку, підпорядкована ІС буде неактивна. Коли використовується тільки одна зовнішня ІС, може виникнути спокуса виключення і лінії SS

за рахунок жорсткої установки низького рівня на вході вибору підпорядкованої мікросхеми. Таке рішення вкрай небажано і може привести до збоїв або взагалі неможливості передачі даних, тому що вхід вибору мікросхеми служить для переведення ІС в її початковий стан і іноді ініціює виведення першого біта даних.

7.2 Підключення ІС за допомогою інтерфейсу SPI

При необхідності підключення до шини SPI декількох мікросхем використовується або незалежне (паралельне) підключення, або каскадне (послідовне).

Незалежне підключення більш поширене, тому що досягається при використанні будь-яких SPI-сумісних мікросхем. Всі сигнали, крім вибору мікросхем, з'єднані паралельно, а ведучий шини, переведенням того чи іншого сигналу \overline{SS} в низький стан, задає, з якою підпорядкованої ІС він буде обмінюватися даними. Головним недоліком такого підключення є необхідність в додаткових лініях для адресації підлеглих мікросхем (*загальне число ліній зв'язку дорівнює $3 + n$, де n – кількість підлеглих мікросхем*).

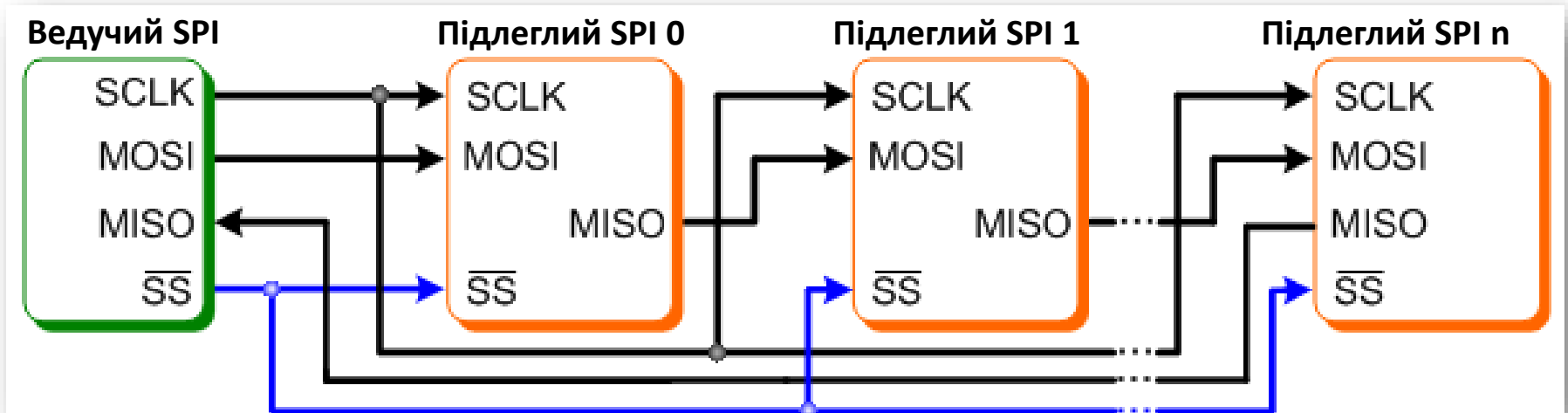


Незалежне (паралельне) підключення

7.2 Підключення ІС за допомогою інтерфейсу SPI

Каскадне включення звільнено від цього недоліку, тому що тут з кількох мікросхем утворюється один великий зсувний регістр. Для цього вихід передачі даних однієї ІС з'єднується зі входом прийому даних іншої, як показано на рисунку. Входи вибору мікросхем тут з'єднані паралельно і, таким чином, **загальне число ліній зв'язку збережено рівним 4**.

Однак використання каскадного підключення можливо тільки в тому випадку, якщо його підтримка вказана в документації на використовувані мікросхеми. Щоб з'ясувати це, важливо знати, що таке підключення по-англійськи називається "daisy-chaining".



Каскадне (послідовне) підключення

7.3 Протокол передачі SPI інтерфейсу

Протокол передачі по інтерфейсу SPI простий і, по суті, ідентичний логіці роботи зсувного регістру, яка полягає у виконанні операції зсуву і, відповідно, побітного введення і виведення даних за певними фронтами сигналу синхронізації. ***Установка даних при передачі і вибірка при прийомі завжди виконуються по протилежних фронтах синхронізації.*** Це необхідно для гарантування вибірки даних після надійного їх встановлення. Якщо до цього врахувати, що в якості першого фронту в циклі передачі може виступати наростаючий або падаючий фронт, то ***всього можливо чотири варіанти логіки роботи інтерфейсу SPI.*** Ці варіанти отримали назву режимів SPI і описуються двома параметрами:

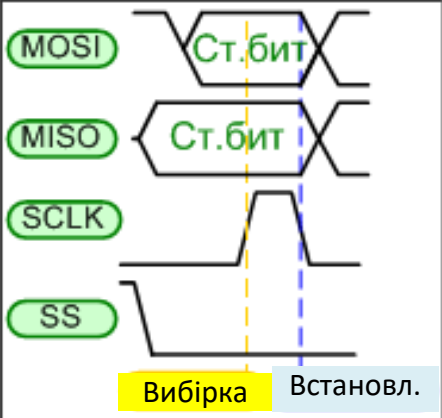
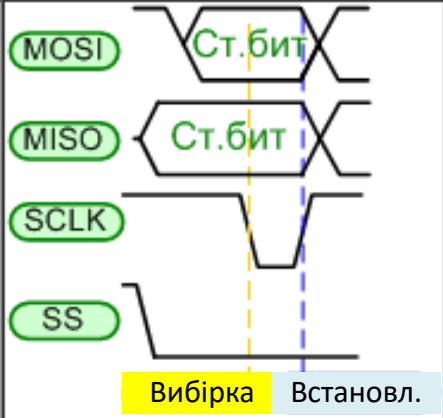
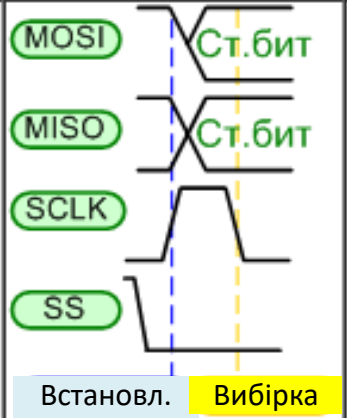
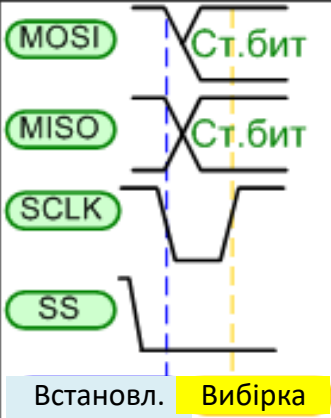
CPOL – вихідний рівень сигналу синхронізації (якщо CPOL = 0, то лінія синхронізації до початку циклу передачі і після його закінчення має низький рівень (тобто перший фронт наростаючий, а останній – падаючий), інакше, якщо CPOL = 1, - високий (тобто перший фронт падаючий, а останній - наростаючий));

CPHA – фаза синхронізації; від цього параметра залежить, в якій послідовності виконується установка і вибірка даних (якщо CPHA = 0, то по передньому фронту в циклі синхронізації буде виконуватися вибірка даних, а потім, по задньому фронту, - установка даних; якщо ж CPHA = 1, то установка даних буде виконуватися по передньому фронту в циклі синхронізації, а вибірка – по задньому).

Інформація по режимам SPI узагальнена в наступній таблиці.

7.3 Протокол передачі SPI інтерфейсу

Таблиця 2 – Режими SPI

Режим SPI	0	1	2	3
CPOL	0	1	0	1
CPHA	0	0	1	1
Часова діаграма першого циклу синхронізації				

Провідна і підпорядкована мікросхеми, що працюють в різних режимах SPI, є несумісними, тому, перед вибором підлеглих мікросхем важливо уточнити, які режими підтримуються провідним шини. **Апаратні модулі SPI, інтегровані в мікроконтролери, в більшості випадків підтримують можливість вибору будь-якого режиму SPI і, тому, до них можливе підключення будь-яких підлеглих SPI-мікросхем (відноситься тільки до незалежного варіанту підключення).** Крім того, протокол SPI в будь-якому з режимів легко реалізується програмно.

7.3 Протокол передачі SPI інтерфейсу

Похідні і сумісні протоколи:

MICROWIRE.

Протокол MICROWIRE компанії National Semiconductor повністю ідентичний протоколу SPI в режимі 0 (CPOL = 0, CPHA = 0).

3-провідний інтерфейс компанії Maxim.

Відмінність цього інтерфейсу полягає в тому, що замість повнодуплексної передачі за двома односпрямованими лініями тут виконується напівдуплексна передача по одній двобічній лінії DQ.

QSPI.

Більш високорівневий протокол, ніж SPI, що дозволяє автоматизувати передачу даних без участі ЦПУ.

Крім того, інтерфейс SPI є основою для побудови ряду спеціалізованих інтерфейсів, в тому числі налагоджувальний інтерфейс JTAG і інтерфейси карт Flash-пам'яті, в тому числі SD і MMC.

7.4 Загальна характеристика інтерфейсу I²C

Кожен, хто займався розробкою радіоелектронної техніки, стикався з ситуацією, коли для узгодження рівнів сигналів, вибірки і адресації функціонально-закінчених вузлів, доводиться використовувати величезну кількість проміжних ІС.

Для збільшення ефективності, спрощення схемотехніки, Philips розробила просту двосторонню двохпровідну шину для так званого "межмікросхемного" (inter-IC) управління.

Шина отримала назву - InterIC, або IIC (I2C) шина.

Стандарт досить давній, придуманий більше 30 років тому, а тому – в різних модифікаціях і під різними назвами – дуже широко поширений в електронних пристроях різного призначення (у Atmel його звать TWI). Можливо, другий за популярністю інтерфейс після UART, за якими найчастіше організується обмін даними між МК, датчиками і виконавчими пристроями.

В даний час тільки Philips виробляє більше 150 найменувань I2C-сумісних пристроїв, функціонально призначених для роботи в електронному обладнанні. У їх числі ІС пам'яті, відеопроцесорів і модулів обробки аудіо- та відео-сигналів, АЦП і ЦАП, драйвери РК-індикаторів, процесори з вбудованим контролером I2C шини і тощо.

Проста двохпровідна послідовна шина I2C мінімізує кількість сполучень між ІС, ІС мають менше контактів і потрібно менше доріжок. Як результат – друковані плати стають простішими і технологічними при виготовленні. Інтегрований I2C-протокол усуває необхідність в дешифраторі адреси та іншої зовнішньої логіці узгодження.

7.4 Загальна характеристика інтерфейсу I²C

I²C шина є однією з модифікацій послідовних протоколів обміну даних. У стандартному режимі забезпечується передача послідовних 8-бітних даних зі швидкістю до 100 кбіт/с і до 400 кбіт/с в "швидкому" режимі. Для здійснення процесу обміну інформацією по I²C шині, використовується всього два сигнали – лінія даних SDA і лінія синхронізації SCL . Для забезпечення реалізації двобічної шини без застосування складних арбітрів шини вихідні каскади пристроїв, підключених до шини, мають відкритий стік або відкритий колектор для забезпечення функції монтажного "1" .

Максимальна допустима кількість IC, приєднаних до однієї шини, обмежується максимальною ємністю шини 400 пФ.

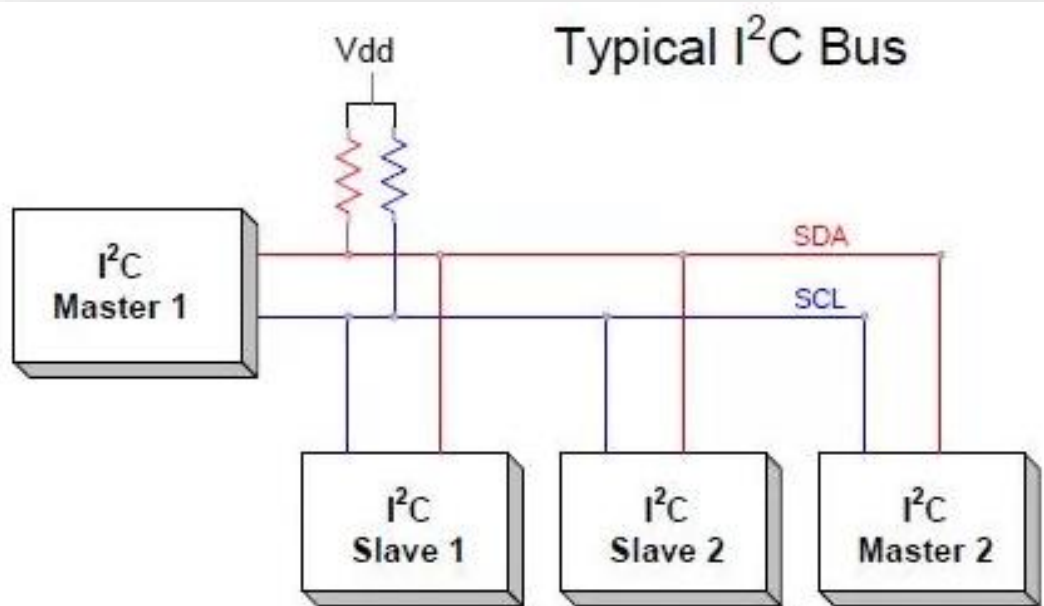
Всього на одній двохпровідній шині може бути до 127 пристроїв, але з огляду на оновлений стандарт можливо підключення до 1000 IC .

Вбудований в мікросхеми апаратний алгоритм перешкодопридушення забезпечує цілісність даних в умовах перешкод значної величини. Все I²C-сумісні пристрої мають інтерфейс, який дозволяє їм зв'язуватися один з одним по шині навіть у тому випадку, якщо їх напруга живлення істотно відрізняється.

7.5 Підключення ІС за допомогою інтерфейсу I²C

Кожен пристрій розпізнається за унікальною адресою і може працювати як передавач або приймач, в залежності від призначення пристрою. Крім того, пристрої можуть бути класифіковані як провідні і ведені при передачі даних. Ведучий – це пристрій, який ініціює передачу даних і виробляє сигнали синхронізації. При цьому будь-який пристрій, що адресується, вважається відомим по відношенню до ведучого. Виходячи з специфікації роботи шини, **в кожен окремий момент в шині може бути тільки один ведучий, а саме той пристрій, що забезпечує формування сигналу SCL шини.**

Ведучий може виступати як в ролі ведучого-передавача, так і ведучого-приймача. Проте - **шина дозволяє мати кілька ведучих, накладаючи певні особливості їхньої поведінки в формуванні сигналів управління і контролю стану шини.**

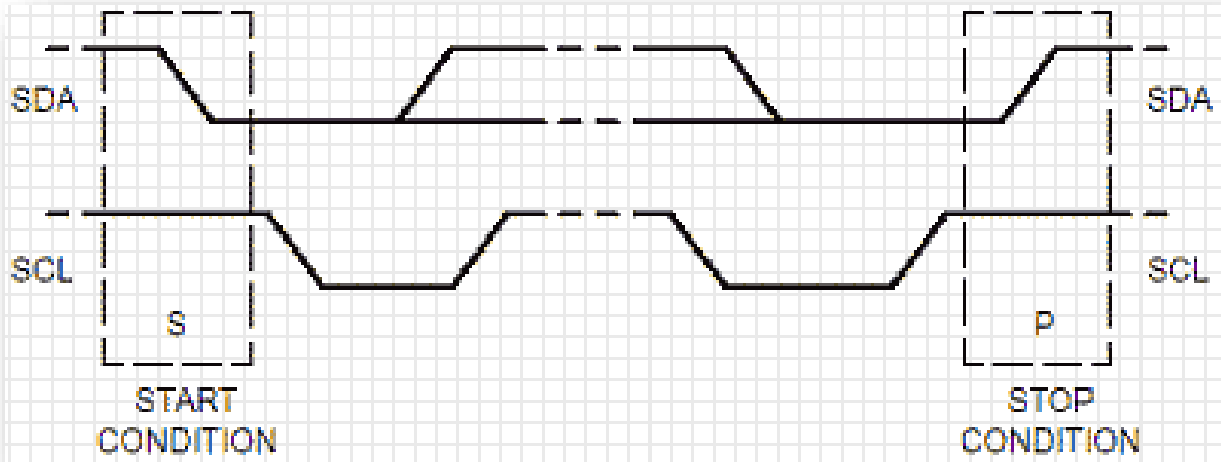


Можливість підключення більше одного ведучого до шини означає, що більш ніж один ведучий може спробувати почати пересилання в один і той же момент часу. Для усунення "зіткнень", які можуть виникнути в даному випадку, розроблена процедура арбітражу – поведінки ведучого при виявленні "захоплення" шини іншим ведучим.

7.6 Протокол обміну сигналами в інтерфейсі I²C

Процедура синхронізації двох пристроїв заснована на тому, що усі I²C-пристрої підключаються до шини за правилом монтажного I. В початковому стані обидва сигналу SDA і SCL знаходяться у високому стані.

Процедура обміну починається з того, що ведучий формує стан **СТАРТ** – **провідний генерує перехід сигналу лінії SDA з високого стану в низький при високому рівні на лінії SCL**. Цей перехід сприймається усіма пристроями, підключеними до шини як ознака початку процедури обміну.



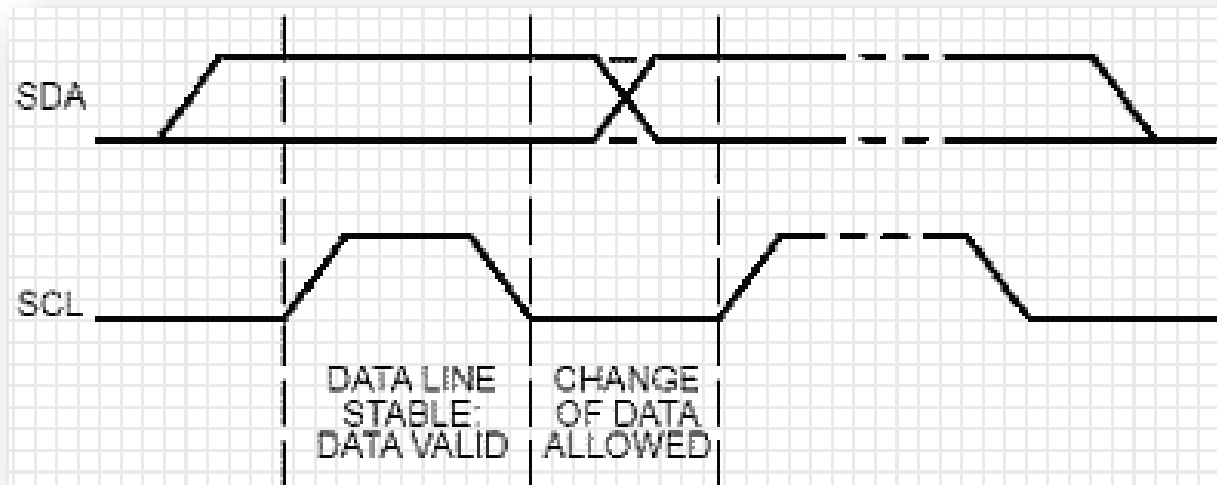
Генерація синхросигналу – це завжди обов'язок ведучого; кожен провідний генерує свій власний сигнал синхронізації при пересиланні даних по шині.

Процедура обміну завершується тим, що ведучий формує стан **СТОП** – **перехід стану лінії SDA з низького стану в високий при високому стані лінії SCL**.

Стани СТАРТ і СТОП завжди виробляються провідним. Вважається, що шина зайнята після фіксації стану СТАРТ. Шина вважається такою, що звільнилася через деякий час після фіксації стану СТОП.

7.6 Протокол обміну сигналами в інтерфейсі I²C

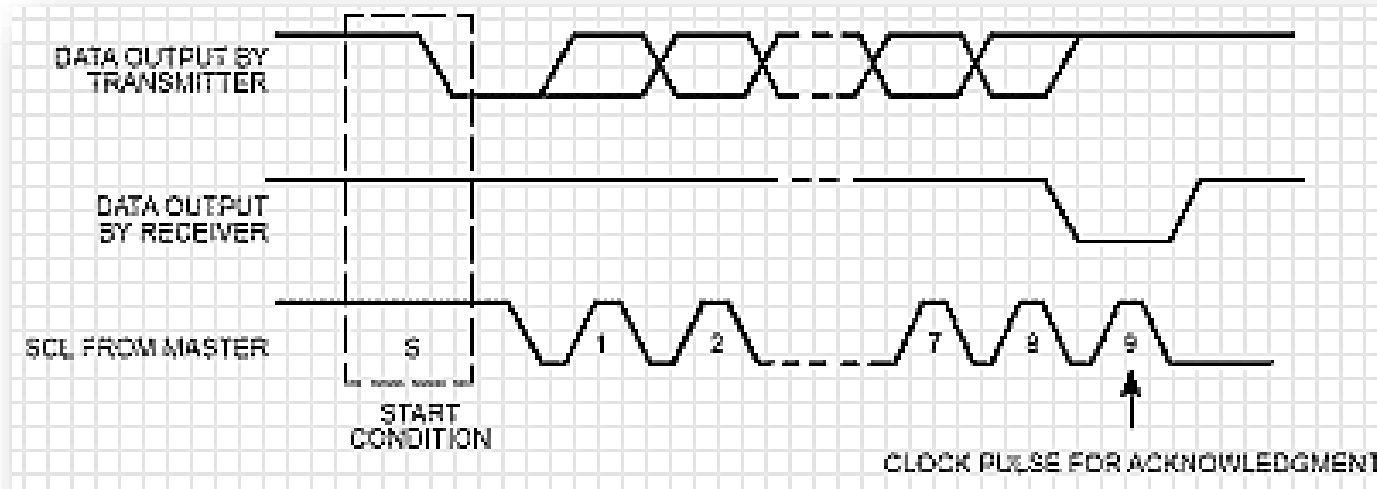
При передачі посілок по шині I²C кожен ведучий генерує свій синхросигнал на лінії SCL. Після формування стану СТАРТ провідний опускає стан лінії SCL в низький стан і виставляє на лінію SDA старший біт першого байту повідомлення. Кількість байт в повідомленні не обмежена. **Специфікація шини I²C дозволяє зміни на лінії SDA тільки при низькому рівні сигналу на лінії SCL. Дані дійсні і повинні залишатися стабільними тільки під час високого стану синхроімпульсу.**



Для підтвердження прийому байту від ведучого-передавача веденим-приймачем в специфікації протоколу обміну по шині I²C вводиться спеціальний біт підтвердження, який виставляється на шину SDA після прийому 8 біта даних. Таким чином, передача 8 біт даних від передавача до приймача завершуються додатковим циклом (формуванням 9-го тактового імпульсу лінії SCL), при якому приймач виставляє низький рівень сигналу на лінії SDA, як ознака успішного прийому байту.

7.6 Протокол обміну сигналами в інтерфейсі I²C

Підтвердження при передачі даних обов'язкове. Відповідний імпульс синхронізації генерується провідним. Передавач відпускає (висока) лінію SDA на час синхроімпульсу підтвердження. Приймач повинен утримувати лінію SDA протягом високого стану синхроімпульсу підтвердження в стабільному низькому стані.



У тому випадку, коли ведений-приймач не може підтвердити свою адресу (наприклад, коли він виконує в даний момент будь-які функції реального часу), лінія даних повинна бути залишена в високому стані. Після цього ведучий може видати сигнал СТОП для переривання пересилання даних. Якщо в пересиланні бере участь провідний-приймач, то він повинен повідомити про закінчення передачі відомому-передавача шляхом непідтвердження останнього байту. Ведений-передавач повинен звільнити лінію даних для того, щоб дозволити ведучому видати сигнал СТОП або повторити сигнал СТАРТ. Синхронізація виконується з використанням підключення до лінії SCL за правилом монтажного I.

7.6 Протокол обміну сигналами в інтерфейсі I²C

Це означає, що ведучий не має монопольного права на управління переходом лінії SCL з низького стану у високий. **У тому випадку, коли відомому необхідний додатковий час на обробку прийнятого біта, він має можливість утримувати лінію SCL в низькому стані до моменту готовності до прийому наступного біта.** Таким чином, лінія SCL буде перебувати в низькому стані протягом найдовшого низького періоду синхросигналів.

Пристрої з більш коротким низькими періодами будуть входити в стан очікування на час, поки не скінчиться довгий період. Коли у всіх задіяних пристроїв скінчиться низький період синхросигналу, лінія SCL перейде в високий стан. Всі пристрої почнуть проходити високий період своїх синхросигналів. Перший пристрій, у якого скінчиться цей період, знову встановить лінію SCL в низький стан. **Таким чином, низький період синхролінії SCL визначається найдовшим періодом синхронізації з усіх задіяних пристроїв, а високий період визначається найкоротшим періодом синхронізації пристроїв.**

Механізм синхронізації може бути використаний приймачами як засіб управління пересиланням даних на байтовому і бітовому рівнях.

На рівні байту, якщо пристрій може приймати байти даних з великою швидкістю, але вимагає певний час для збереження прийнятого байту або підготовки до прийому наступного, то він може утримувати лінію SCL в низькому стані після прийому і підтвердження байту, переводячи таким чином передавач в стан очікування.

7.6 Протокол обміну сигналами в інтерфейсі I²C

Кожен пристрій, підключений до шини, може бути програмно адресований за унікальною адресою. Для вибору приймача повідомлення провідний використовує унікальну адресну компоненту в форматі посилки. При використанні однотипних пристроїв, IC часто мають додатковий селектор адреси, який може бути реалізований як у вигляді додаткових цифрових входів селектора адреси, так і у вигляді аналогового входу. При цьому адреси таких однотипних пристроїв виявляються рознесені в адресному просторі пристроїв, підключених до шини.

У звичайному режимі використовується 7-бітова адресація. Процедура адресації на шині I²C полягає в тому, що перший байт після сигналу СТАРТ визначає, який ведений адресується провідним для проведення циклу обміну.

Виняток становить адреса "Спільний виклик", який адресує всі пристрої на шині. Коли використовується ця адреса, всі пристрої в теорії повинні послати сигнал підтвердження. Однак, пристрої, які можуть обробляти "загальний виклик", на практиці зустрічаються рідко.

Перші сім бітів першого байту утворюють адресу веденого. Восьмий, молодший біт, визначає напрямок пересилки даних. "Нуль" означає, що провідний буде записувати інформацію в обраного веденого. "Одиниця" означає, що провідний буде зчитувати інформацію з веденого.

Після того, як адреса послана, кожен пристрій в системі порівнює перші сім біт після сигналу СТАРТ зі своєю адресою. При збігу пристрій вважає себе обраним як ведений-приймач або як ведений-передавач, в залежності від біта напрямку.

7.6 Протокол обміну сигналами в інтерфейсі I²C

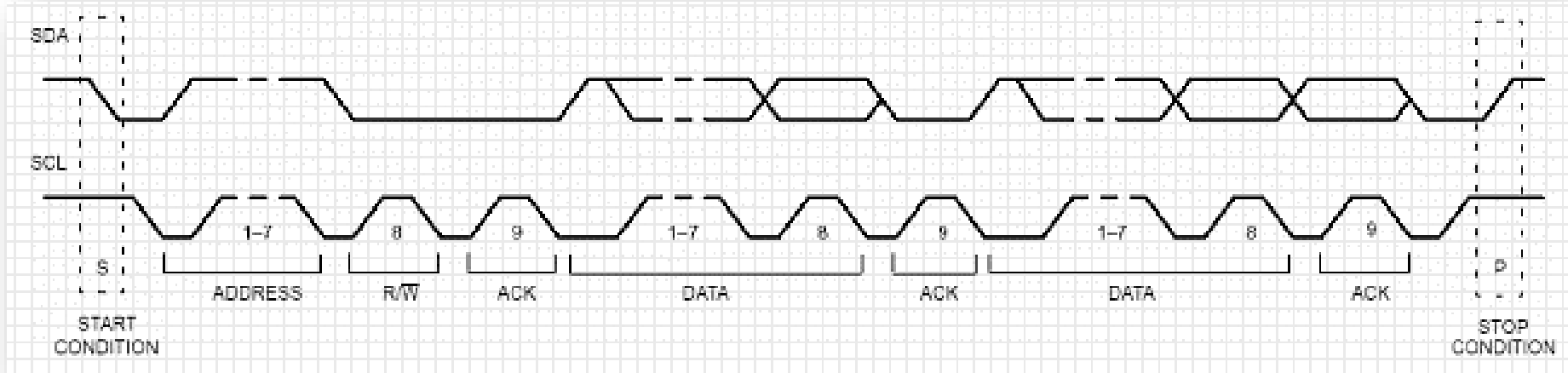
Адреса веденого може складатися з фіксованою і програмованою частин.

Часто трапляється, що в системі буде кілька однотипних пристроїв (наприклад ІС пам'яті, або драйверів LED-індикаторів), тому за допомогою програмованої частини адреси стає можливим підключити до шини максимально можливу кількість таких пристроїв. Кількість програмованих біт в адресі залежить від кількості вільних виводів мікросхеми.

Всі ІС, що підтримують роботу в стандарті шини I²C, мають набір фіксованих адрес, перелік яких зазначений виробником в описах контролерів.

Комбінація біт 11110XX адреси зарезервована для 10-бітної адресації.

У загальному вигляді процес обміну по шині від моменту формування стану СТАРТ до стану СТОП можна проілюструвати наступною часовою діаграмою.



7.7 Розширення інтерфейсу I²C

Стандартна шина I²C зі швидкістю передачі даних 100 кбіт/с і 7-бітовим адресом існує вже протягом більше 20 років в незмінному вигляді. Стандартна шина I²C прийнята повсюдно як стандарт для сотень типів мікросхем. В даний час специфікація шини I²C **розширена в двох напрямках: збільшення швидкодії і розширення адресного простору для розширення номенклатури розроблених нових пристроїв.**

Введення специфікації "швидкого" режиму, що дозволяє в чотири рази збільшити швидкість передачі даних до 400 кбіт/с. Необхідність в цьому "розширенні" стандарту була через необхідність пересилання великих обсягів інформації, і, як наслідок, необхідність збільшення пропускної здатності каналу.

Введення специфікації "10-бітної адресації", що дозволяє використовувати тисяча двадцять чотири додаткових адрес, тому що більшість з 112 адрес, допустимих при 7-бітної адресації, вже були використані більш ніж один раз. Для запобігання проблем з розміщенням адрес нових пристроїв, бажано мати більшу кількість адресних комбінацій. Приблизно десятикратне збільшення кількості доступних адрес отримано при використанні нової 10-бітної адресації.

Всі нові пристрої з I²C інтерфейсом працюють в швидкому режимі. Переважно, вони повинні вміти приймати і/або передавати дані на швидкості 400 кбіт/с. Як мінімум вони повинні бути здатні входити в синхронізацію в швидкому режимі, з тим щоб знизити швидкість передачі (шляхом подовження низької періоду SCL) до допустимої величини.

7.7 Розширення інтерфейсу I²C

Швидкі пристрої як правило сумісні знизу-вгору, що означає їх здатність працювати зі стандартними пристроями по повільній шині. Очевидно, що стандартні пристрої не здатні працювати в швидкій шині, тому що вони не можуть синхронізуватися на високій швидкості. Ведені швидкі пристрої можуть мати як 7-бітову, так і 10-бітову адресу. Однак, 7-бітова адреса більш краща, так як її апаратна реалізація більш проста і довжина посилки менше. **Пристрої з 7- і 10-бітовими адресами можуть одночасно використовуватися на одній шині, незалежно від швидкості передачі.**

У швидкому режимі протокол, формат, логічні рівні і максимальна місткість навантаження ліній шини залишається незмінними. Алгоритм синхронізації ліній SDA і SCL не змінений. Вхідні ланцюги швидких пристроїв повинні мати вбудоване придушення викидів і тригер Шмітта на обох лініях. Як правило при зникненні напруги живлення швидких пристроїв виводи, підключені до ліній I²C шини, повинні переходити в третій стан.

10-бітна адресація також не змінює формат шини. Для цього використовується зарезервована адресна комбінація 1111XXX перших семи біт першого байту. 10-бітна адресація не впливає на існуючу 7-бітну адресацію. Хоча є вісім можливих комбінацій послідовності 1111XXX, з них використовуються тільки чотири – 11110XX. Комбінації типу 11111XX зарезервовані для подальших поліпшень шини. Призначення бітів перших двох байтів 10-бітову адресу формується з перших двох байтів. Перші сім біт першого байту є комбінацією виду 11110XX, де два молодших біта (XX) є двома старшими (9 і 8) бітами 10-бітного адреси; восьмий біт першого байту – біт напрямку.

Тема 8. Розподілені системи на основі інтерфейсу 1-Wire та Bluetooth

8.1 Устрій інтерфейсу 1-Wire.

8.2 Порядок обміну сигналами за інтерфейсом 1-Wire.

- ✓ формування сигналів Reset і Presence;
- ✓ передача інформаційних бітів по шині;
- ✓ обмін на шині за допомогою спеціальних команд;
- ✓ визначення адреси пристрою.

8.3 Елементна база інтерфейсу 1-Wire.

8.4 Найбільш поширені застосування технології 1-Wire.

8.5 Загальна характеристика Bluetooth.

8.6 Топологія розподіленої мережі Bluetooth.

8.7 Основні параметри Bluetooth.

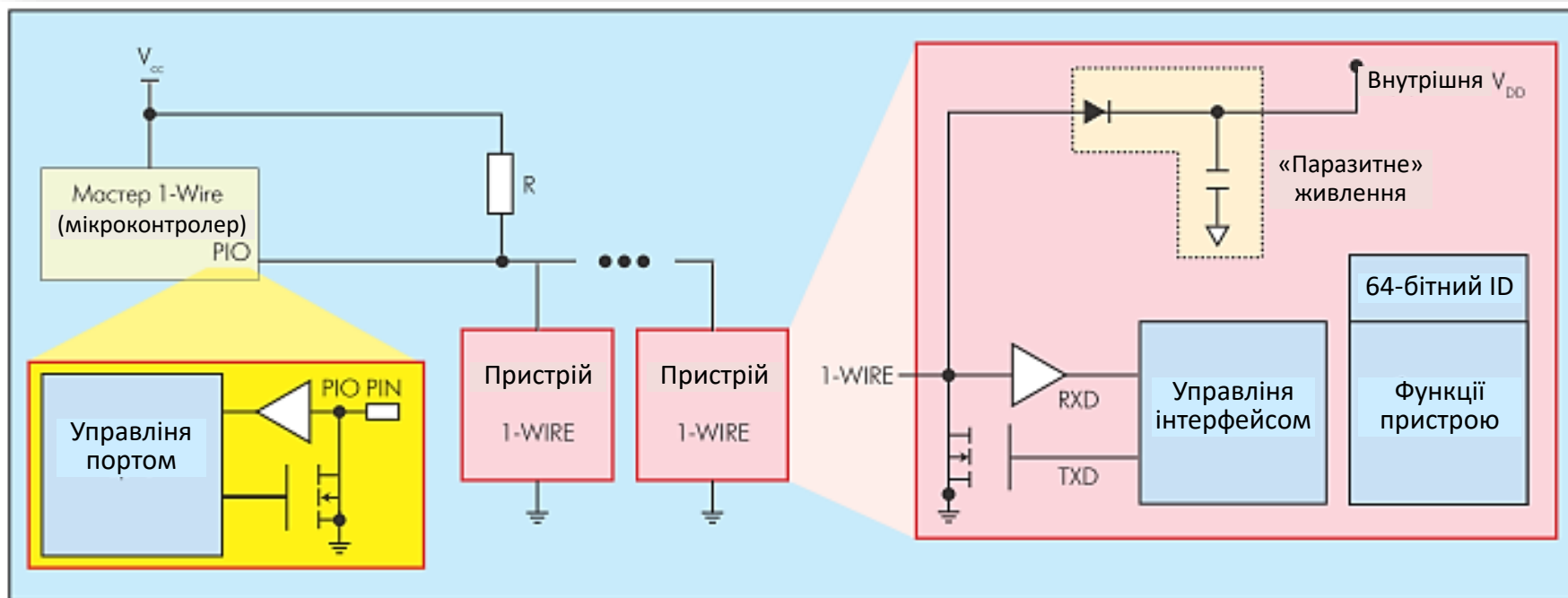
8.8 Архітектура Bluetooth.

8.9 Класи і профілі Bluetooth.

8.1 Устрій інтерфейсу 1-Wire

Інтерфейс 1-Wire був запропонований фірмою Dallas Semiconductor в кінці 90-х років минулого століття. Системи 1-Wire привабливі завдяки легкості монтажу, низькою вартості пристроїв, можливості розпізнавати пристрої при підключенні до функціонуючої мережі, великому числу пристроїв в мережі і т.д.

Типова система 1-Wire складається з керуючого контролера (майстра або ведучого) і одного або декількох пристроїв (ведених), приєднаних до загальної шини.



Пристрої підключаються до шини за схемою з відкритим стоком і резистором, що підтягує. Рівень сигналів в шині – від 3 до 5 В. У пасивному стані в лінії підтримується високий рівень напруги. Всі сигнали формуються за допомогою замикання сигнальної шини на землю (низький рівень напруги).

8.1 Устрій інтерфейсу 1-Wire

Головна особливість шини 1-Wire в тому, що вона використовує лише два дроти, один – сигнальний, інший – для заземлення пристроїв. По сигнальному проводу можливо і електроживлення пристроїв 1-Wire – так зване паразитне живлення. Джерелом живлення служить конденсатор, який заряджається від сигнальної лінії, що входить до складу ведених пристроїв ланцюга.

Більшість пристроїв 1-Wire підтримують дві швидкості передачі даних: стандартну – близько 15 кбіт/с і підвищену (overdrive) – близько 111 кбіт/с. Зрозуміло, що чим вище швидкість, тим більше обмежень на довжину шини і число пристроїв, які підключаються до неї.

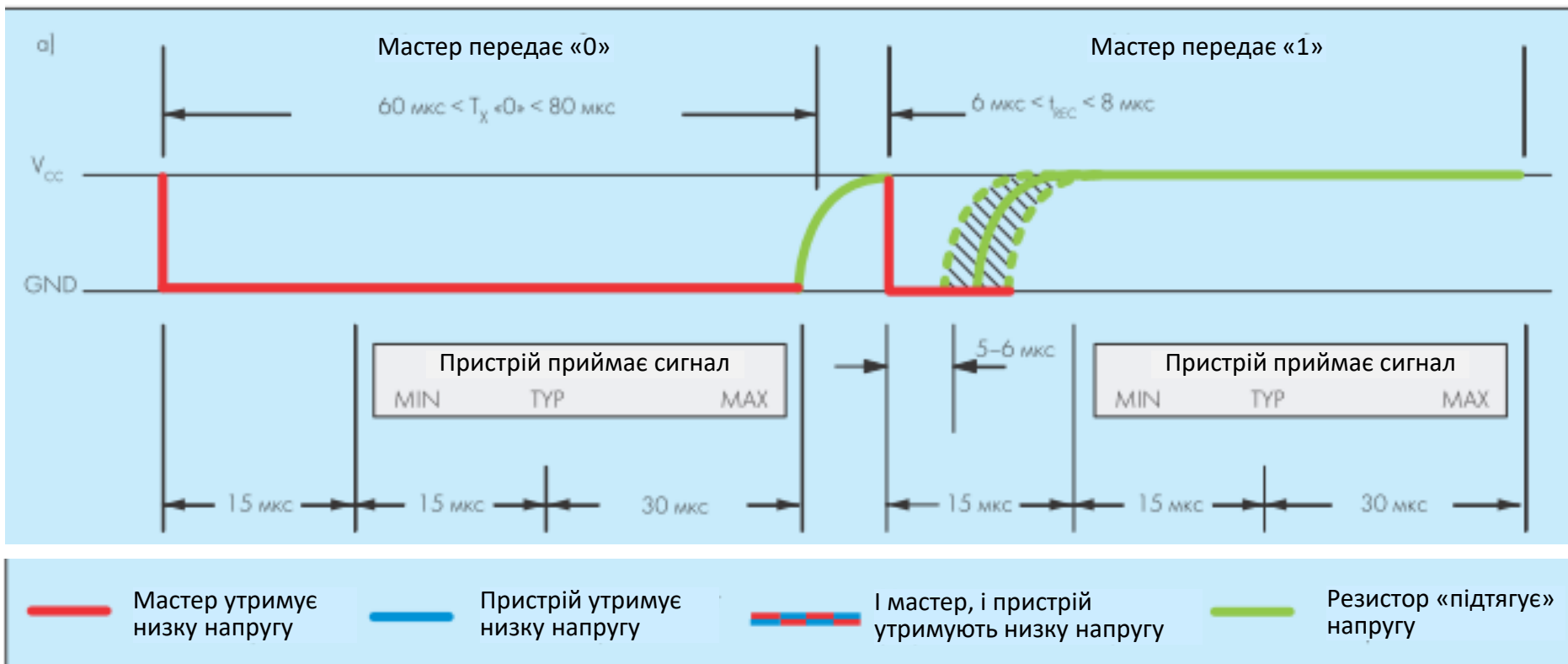
Режим передачі даних по шині 1-Wire – напівдуплексний: майстер і ведені пристрої передають дані по черзі.

Кожна транзакція через інтерфейс 1-Wire починається з того, що **майстер передає імпульс Reset**. Для цього він переводить напругу в шині на низький рівень і утримує його в цьому стані протягом 480 мкс. Потім майстер відпускає шину, і підтягаючий резистор повертає напругу до високого логічного рівня. **Всі ведені пристрої, виявивши сигнал Reset і дочекавшись його закінчення, передають свій сигнал – Presence**. Він являє собою сигнал низького рівня тривалістю 100-200 мс. Пристрій може генерувати сигнал Presence і без імпульсу Reset – наприклад, у такий спосіб воно повідомляє про себе при підключенні до шини.

Після передачі імпульсу Presence пристрій 1-Wire готове до прийому команд.

8.2 Порядок обміну сигналами за інтерфейсом 1-Wire

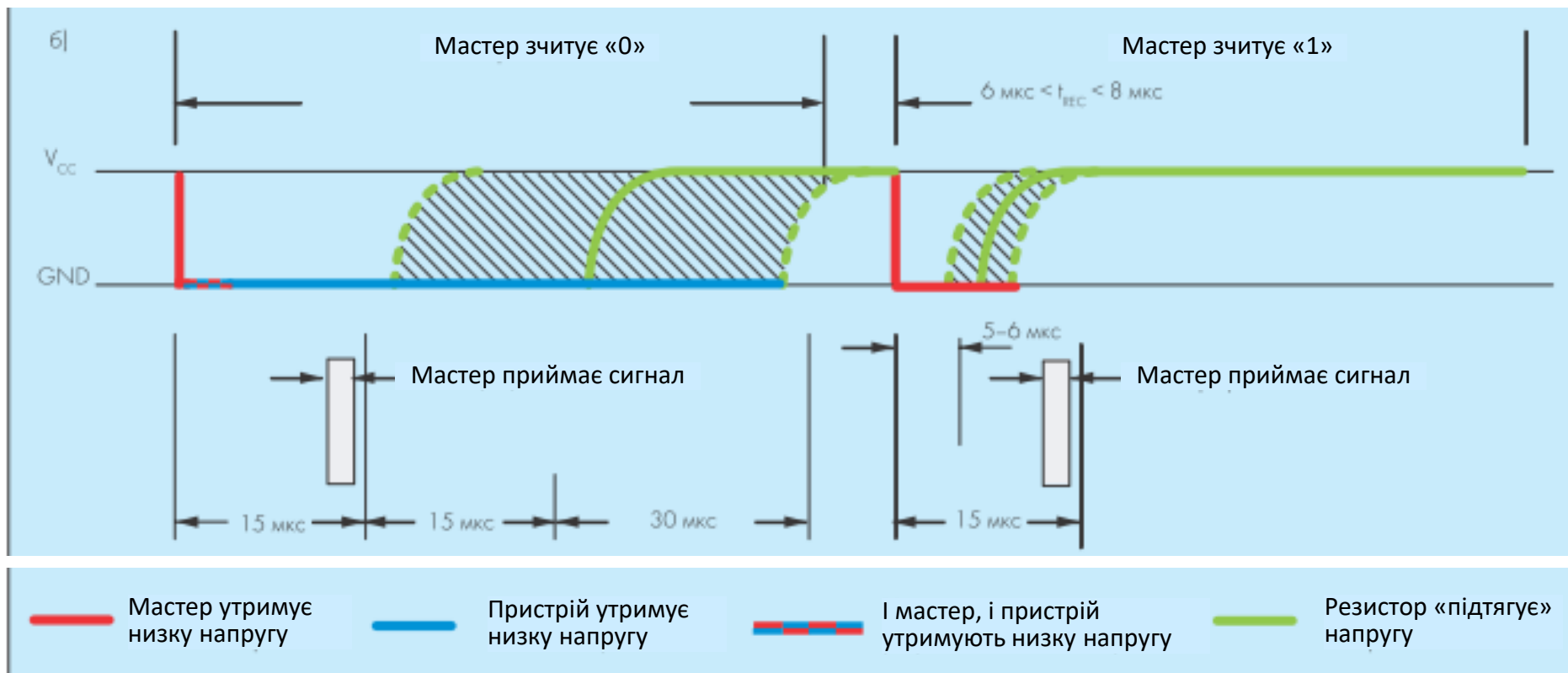
Весь інформаційний обмін в шині відбувається під керуванням майстра. Для передачі кожного біта виділяється спеціальний часовий проміжок (тайм-слот) тривалістю близько 80 мкс. На початку кожного тайм-слота майстер переводить лінію на нульовий рівень. Якщо далі майстер хоче передати 0, він утримує напругу на низькому рівні як мінімум 60 мкс. При передачі одиниці майстер утримує нульову напругу 5-6 мкс, а потім відпускає лінію і вичікує приблизно 60 мкс до початку формування наступного тайм-слота.



Передача інформаційних бітів по шині 1-Wire: майстер передає сигнали

8.2 Порядок обміну сигналами за інтерфейсом 1-Wire

Якщо майстер очікує дані від ведених пристроїв, він також позначає початок тайм-слота, обнуляючи лінію на 5-6 мкс, після чого перестає утримувати низьку напругу і протягом короткого часу слухає лінію. Якщо пристрій хоче передати нуль, воно само обнуляє лінію відразу після реєстрації імпульсу початку тайм-слота. Якщо пристрою потрібно передати одиницю, воно ніяких дій не робить. Відзначимо, що наведені значення тимчасових інтервалів відповідають стандартній швидкості передачі даних через інтерфейс 1-Wire. У режимі overdrive ці інтервали відповідно зменшуються.



Передача інформаційних бітів по шині 1-Wire: майстер зчитує сигнали

8.2 Порядок обміну сигналами за інтерфейсом 1-Wire

Весь обмін на шині 1-Wire відбувається за допомогою спеціальних команд. Їх число для кожного типу пристроїв різне. Але є і мінімальний набір стандартних команд, які підтримують всі 1-Wire-пристрої – так звані ROM-команди. **Формат команд простий – ідентифікатор команди (1 байт), за яким можуть слідувати дані (ідентифікатор пристрою, корисні дані і т.п.).** Всі пристрої в мережі знають довжину кожної команди.

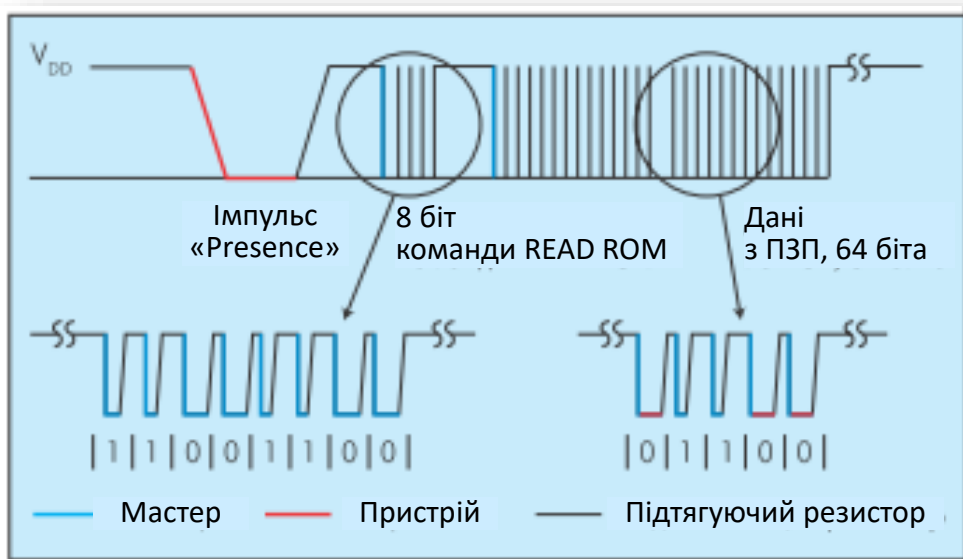
У кожного пристрою 1-Wire є 64-розрядний ідентифікатор (ID). Він складається з **8-розрядного коду сімейства, який ідентифікує тип пристрою і підтримувані їм функції, 48-розрядної серійного номера і 8-бітного поля коду циклічного надлишкового контролю (CRC-8).** ID вводиться при виготовленні пристрою і зберігається в ПЗП. Фірма Maxim гарантує, що один раз використана адреса ніколи не повториться в іншому пристрої. Справді, 48 біт це $2,81 \times 10^{14}$ різних чисел. Якщо виробляти 1000 млрд (10^{12}) різних пристроїв щорічно, то все серійні номери можна використовувати не раніше ніж через 281 рік – і це тільки для одного сімейства.

Весь обмін командами ініціює майстер. Початок нового циклу транзакцій він зазначає командою Reset, і, отримавши підтвердження, вибирає пристрій спеціальної командою MATCH ROM, передаючи її ідентифікатор (55_{16}) і 64 біта ID пристрою, що адресується. Отримавши таку команду, ведений пристрій з даним ID очікує нових команд від майстра, а всі інші залишаються в пасивному стані до наступної команди Reset. В системі з одним пристроєм можна не передавати ID, використовуючи команду SKIP ROM. В результаті ведене пристрій вважає себе обраним без отримання адреси.

8.2 Порядок обміну сигналами за інтерфейсом 1-Wire

Після того, як майстер вибрав пристрій для взаємодії, можна починати процес управління цим пристроєм і обміну даними з ним. Для цього використовуються команди, які специфічні для кожного типу пристроїв.

Але щоб почати роботу з певним пристроєм, майстер повинен знати його ID. Якщо в системі тільки один ведений пристрій, його адресу можна визначити за допомогою команди READ ROM. У відповідь на команду READ ROM пристрій передає свою 64-бітову адресу.



Читання адреси пристрою

Якщо ж в системі декілька пристроїв з невідомими ID, спроба використовувати команду READ ROM призводить до колізії. У цьому випадку для визначення адрес використовується спеціальний алгоритм пошуку, в основі якого лежить команда SEARCH ROM. Майстер передає команду SEARCH ROM. У відповідь всі пристрої, підключені до шини, висилають молодший біт своєї адреси.

Властивості шини 1-Wire такі, що при одночасній передачі сигналів усіма пристроями результат буде дорівнює логічному І значень усіх посланих бітів. Отже, сумарний відгук дорівнює 1, тільки коли сигнали від всіх пристроїв рівні 1.

8.2 Порядок обміну сигналами за інтерфейсом 1-Wire

Після прийому першого біта адреси майстер ініціює наступний тайм-слот, в якому пристрій передає інвертований перший біт. Зіставляючи значення результатів запиту істинного і інверсного бітів, можна отримати якусь інформацію про значеннях перших бітів адрес пристроїв (таблиця).

Інформація про значення біт в адресах пристроїв 1-Wire

Істинний біт	Інверсний біт	Інформація
0	0	В поточному біті адрес є як 0, так і 1. Це так звана «розбіжність» (discrepancy)
0	1	В біті адрес є присутні тільки нулі
1	0	В біті адрес є присутні тільки одиниці
1	1	В пошуку не бере участі жодний пристрій

Таким чином, при комбінаціях 0 1 і 1 0 майстер знає значення першого біта адреси, фіксує його і по тій же схемою може переходити до визначення наступного. Після отримання інверсного біта майстер передає певний біт веденим пристроям.

Якщо його значення збігається зі значенням поточного біта з адреси пристрою, то пристрій продовжує брати участь в пошуку і видає у відповідь наступний біт своєї адреси. Якщо не було "розбіжності", то значення біта, яке виставляється майстром, визначено. У разі розбіжності майстер посилає нульовий біт. Така послідовність - читання біта адреси та інверсного біта, передача біта майстром – повторюється для наступних 63 бітів адреси. Таким чином, алгоритм пошуку послідовно виключає всі пристрої, поки не залишається один останній – його адреса і визначається в першому циклі пошуку.

8.2 Порядок обміну сигналами за інтерфейсом 1-Wire

Після того, як адресу першого пристрою визначено, пошук триває для наступного пристрою. Алгоритм запам'ятовує місце останньої розбіжності і вибирає іншу гілку дерева пошуку (майстер посилає в цьому місці біт з іншим значенням). Процес триває до тих пір, поки не буде пройдена гілка, відповідна останньому пристрою. В результаті пошуку стають відомі адреси всіх пристроїв, під'єднаних до шини, і їх число.

Відзначимо, що **можливість ідентифікації і швидкого включення в мережу тільки що підключеного пристрою робить 1-Wire ефективним рішенням для багатьох додатків**. На практиці це означає, що прилад досить просто підключити до мережі, і всі подальші транзакції відбудуться автоматично. Наприклад, так можна зчитати дані з пам'яті датчика, прочитати код електронної мітки або електронного ключа, прийняти масив значень від приладової мережі і т.п.

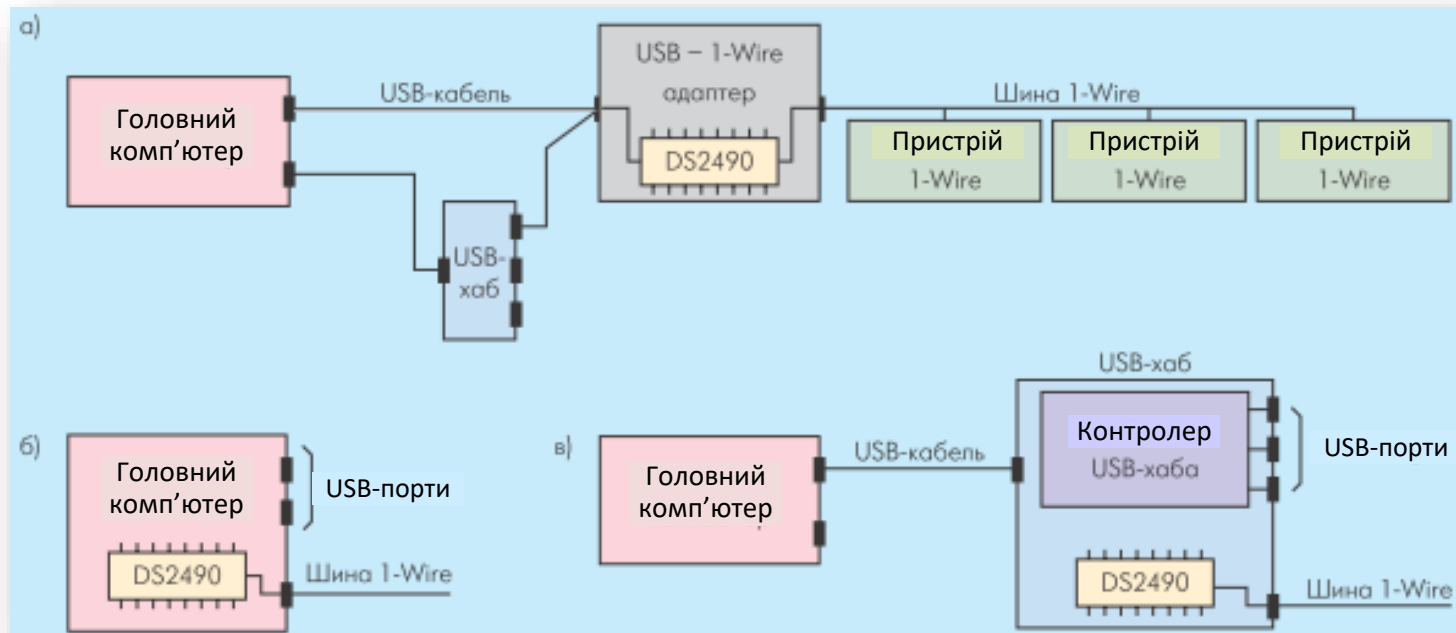
Не менш важливо, що **мережа 1-Wire відноситься до тих, що самосинхронізуються, тобто не вимагає окремої лінії для передачі тактових сигналів**. Величезне число ID пристроїв, що підключаються, вигідно виділяє її на тлі інших послідовних мереж.

Таким чином, завдяки своїм вигравшим можливостям – один провід для передачі даних і управління пристроями, під'єднання пристроїв через один контакт, живлення підключених пристроїв по дроту передачі даних, наявність у кожного пристрою унікальної адреси, низька вартість елементної бази – інтерфейс 1-Wire широко представлений в найрізноманітніших виробках сучасної електроніки.

8.3 Елементна база інтерфейсу 1-Wire

Для реалізації 1-Wire фірма Maxim/Dallas пропонує ряд пристроїв. У ролі майстра може виступати як ПК, так і спеціалізовані МК. В асортименті Maxim/Dallas присутні пристрої, які забезпечують перехід до 1-Wire інтерфейсу від стандартних інтерфейсів комп'ютера (наприклад, USB і RS-232), який керує роботою пристроїв 1-Wire.

Так, мікросхема DS2490 служить мостом між інтерфейсами USB і 1-Wire. DS2490 використовується в системах 1-Wire декількома способами: входить до складу адаптера USB – 1-Wire, який приєднується до USB-порту головного комп'ютера безпосередньо кабелем або через USB-хаб; вбудовується в головний комп'ютер або ж в USB-хаб.



Способи використання DS2490: в складі зовнішнього USB - 1-Wire адаптера (а); всередині головного комп'ютера (б); в складі USB-хаба (в)

8.4 Найбільш поширені застосування технології 1-Wire



Наявність унікальних 64-бітних адрес дозволяє широко використовувати пристрої 1-Wire в системах аутентифікації.

Тут вони часто застосовуються в пристроях iButton. Це мікросхема з введеним на етапі виробництва 64-бітним адресом, розміщена в круглий корпус з нержавіючої сталі діаметром 16 мм (MicroCAN). Такі пристрої функціонують, наприклад, в домофонних ключах.

ІС з підтримкою 1-Wire (наприклад, DS2401, DS2431, DS28E01-100) використовуються також для ідентифікації картриджів принтерів, медичних сенсорів, ємностей з реагентами та ін. Перевага мікросхем 1-Wire в тому, що для контролю ідентифікованого пристрою потрібен всього один контакт. Такі ІС укладені в спеціальний плоский корпус (SFN - Single Flat No lead) розміром 6×6 мм, який полегшує їх приєднання до пристрою.

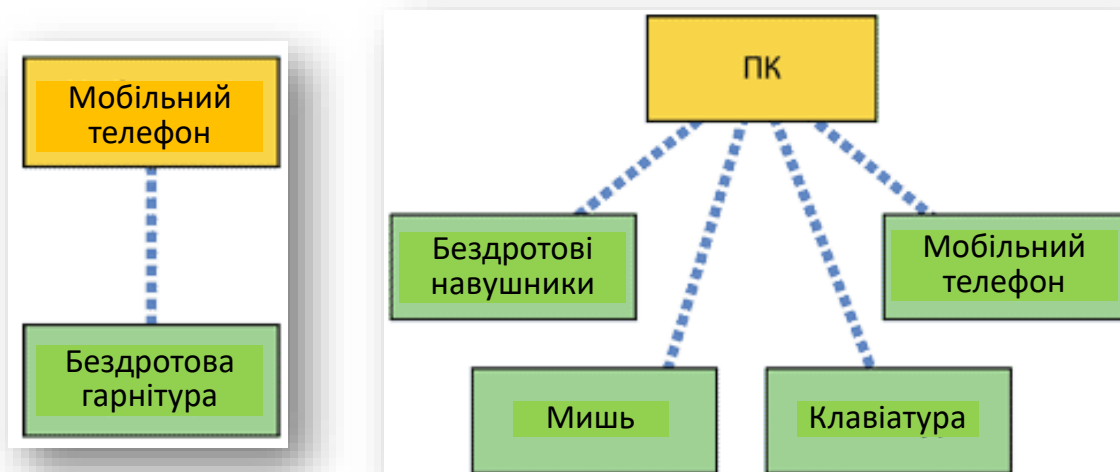
Ще одне поширене застосування 1-Wire - системи автоматизації. В першу чергу це системи багатоточкового вимірювання температури різних середовищ і моніторингу теплового режиму приміщень. Температуру можна вимірювати датчиками виробництва тієї ж Maxim/Dallas. Найбільш популярний з них – цифровий термометр DS18S20. Він має вихідну розрядність 9 біт і вимірює температуру в діапазоні від -55 до 125° С. Точність вимірювань складає 0,5° С в діапазоні -10 ... 85° С. Оскільки кожен термометр, як і будь-який пристрій 1-Wire, має унікальну 64-бітову адресу, до однієї шині 1-Wire можна підключати безліч таких приладів.

8.5 Загальна характеристика Bluetooth

Bluetooth – це бездротовий інтерфейс з невеликим радіусом дії, створений 1994 року інженерами шведської компанії Ericsson.

Основними перевагами Bluetooth у порівнянні з конкуруючими рішеннями є низький рівень енергоспоживання та невисока вартість приймачів, що дозволяє застосовувати його навіть у малогабаритних пристроях з мініатюрними елементами живлення. Крім того, виробники обладнання не повинні виплачувати ліцензійні відрахування за використання інтерфейсу Bluetooth у виробках.

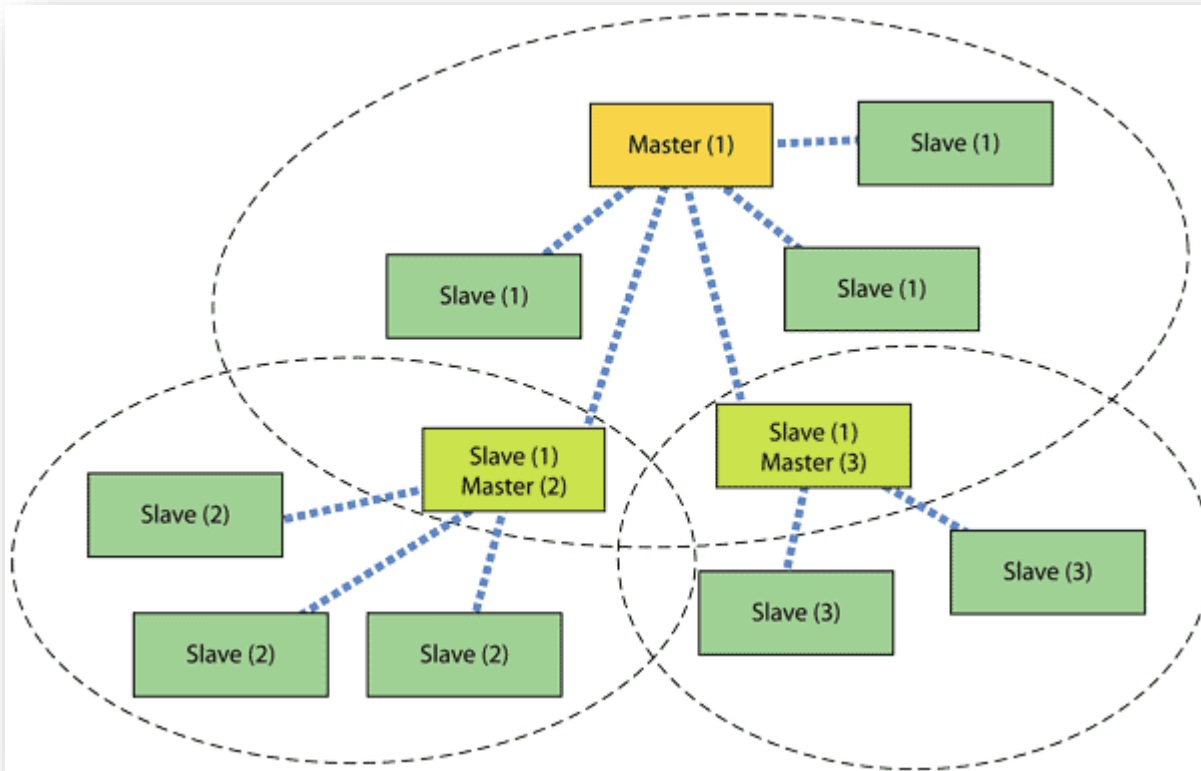
Основним призначенням Bluetooth є створення так званих **персональних мереж** (Private Area Networks, PAN), які забезпечують можливість обміну даними між розташованими поблизу (всередині одного будинку, приміщення, транспортного засобу тощо) настільними та портативними ПК, периферійними та мобільними пристроями та ін.



За допомогою **Bluetooth** можна об'єднати як два, так і кілька пристроїв. У першому випадку підключення здійснюється за схемою «точка-точка», у другому - за схемою «точка-багаточка».

8.6 Топологія розподіленої мережі Bluetooth

Незалежно від схеми, що застосовується, один з пристроїв є ведучим (master), інші – веденими (slave). Провідний пристрій задає шаблон, який використовуватимуть всі керовані пристрої, а також синхронізує їх роботу. З'єднані таким чином пристрої утворюють **пікомережу** (piconet).



Топологія розподіленої мережі, що об'єднує кілька пікомереж

В рамках однієї пікомережі можуть бути об'єднані одне провідне і до семи ведених пристроїв. Крім того, допускається наявність у пікомережі додаткових ведених пристроїв (понад сім), які мають статус заблокованих (parked): вони не беруть участь в обміні даними, але при цьому перебувають у синхронізації з провідним пристроєм.

8.7 Основні параметри Bluetooth

Декілька пікомереж можна об'єднати в розподілену мережу (scatternet). Для цього пристрій, що працює як ведений в одній пікосеті, повинен виконувати функції ведучого в іншій. При цьому пікомережі, що входять до складу однієї розподіленої мережі, не синхронізовані один з одним і використовують різні шаблони. Максимальна кількість пікомереж у складі розподіленої мережі не може перевищувати десяти. Таким чином, **розподілена мережа дозволяє об'єднати в цілому до 71 пристрою.**

Передача даних ведеться радіоканалом в частотному діапазоні 2,4-2,4835 ГГц з використанням методу псевдовипадкової перебудови робочої частоти (Frequency-Hopping Spread Spectrum, FHSS). Цей діапазон розбитий на 79 каналів, кожен із яких займає смугу шириною 1 МГц. У верхній і нижній частинах діапазону передбачені смуги, що не використовуються (захисні). Для передачі даних застосовується гаусова фазова модуляція, яка передбачає зміну несучої частоти в часі відповідно до кривої гауса, що дозволяє обмежити спектр випромінюваного сигналу.

Обмін даними здійснюється всередині часових інтервалів (тайм-слотів) завдовжки 625 мкс. Після передачі кожного слота провадиться перехід на інший частотний канал. На каналному рівні обмін даними здійснюється пакетами, кожен із яких може мати довжину від одного до п'яти слотів. Частина слотів може бути зарезервована для синхронних каналів (які використовуються для передачі поточкових даних). Таким чином, паралельно із синхронними даними можуть передаватися і асинхронні.

8.7 Основні параметри Bluetooth

Специфікація Bluetooth передбачає два види зв'язку: **синхронний із встановленням з'єднання (Synchronous Connection-Oriented, SCO)** та **асинхронний без встановлення з'єднання (Asynchronous Connection-Less, ACL)**. Перший варіант використовується для організації каналу «точка-точка» між ведучим і веденими пристроями. Другий служить для зв'язку за схемою «точка-багатоточка» між провідним і всіма веденими пристроями даної пікомережі.

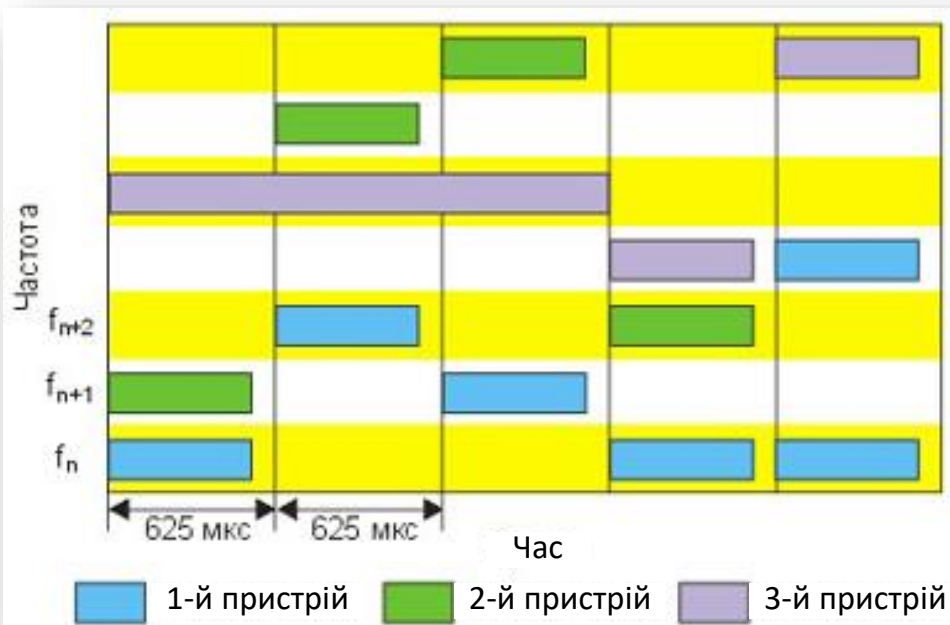
Перший вид, SCO, розрахований для встановлення симетричного з'єднання "точка-точка" і **служить переважно для передачі мовних повідомлень. Швидкість передачі SCO дорівнює 64 Кбіт/с. Другий ACL, призначений для пакетної передачі даних.** Він підтримує симетричні та асиметричні з'єднання типу "точка-багатоточка". **Швидкість передачі пакетної інформації при ACL становить близько 721 Кбіт/с.** Пакети мають фіксований формат. На початку блоку знаходиться 72-бітний код доступу. Він може використовуватися, зокрема, для синхронізації пристроїв. За ним слідує 54-біт заголовок пакета, що містить контрольну суму пакета та інформацію про його параметри (наприклад, про повторну передачу блоку даних). Замикає пакет область, що безпосередньо містить інформацію, що пересилається. Розмір цієї області варіюється від 0 до 2745 біт.

Основним принципом побудови систем **Bluetooth** є використання методу розширення спектру при стрибкоподібній зміні частоти FHSS. Весь виділений для Bluetooth радіозв'язку частотний діапазон 2,4-2,4835 ГГц розбитий на N частотних каналів. **Смуга кожного каналу 1 МГц, рознесення каналів – 140-175 кГц. Для кодування пакетної інформації використовують частотну маніпуляцію.**

8.7 Основні параметри Bluetooth

Для США та Європи $N = 79$. Виняток становлять Іспанія та Франція, де для Bluetooth застосовується 23 частотні канали. Зміна каналів проводиться за псевдовипадковим законом 1600 разів на секунду.

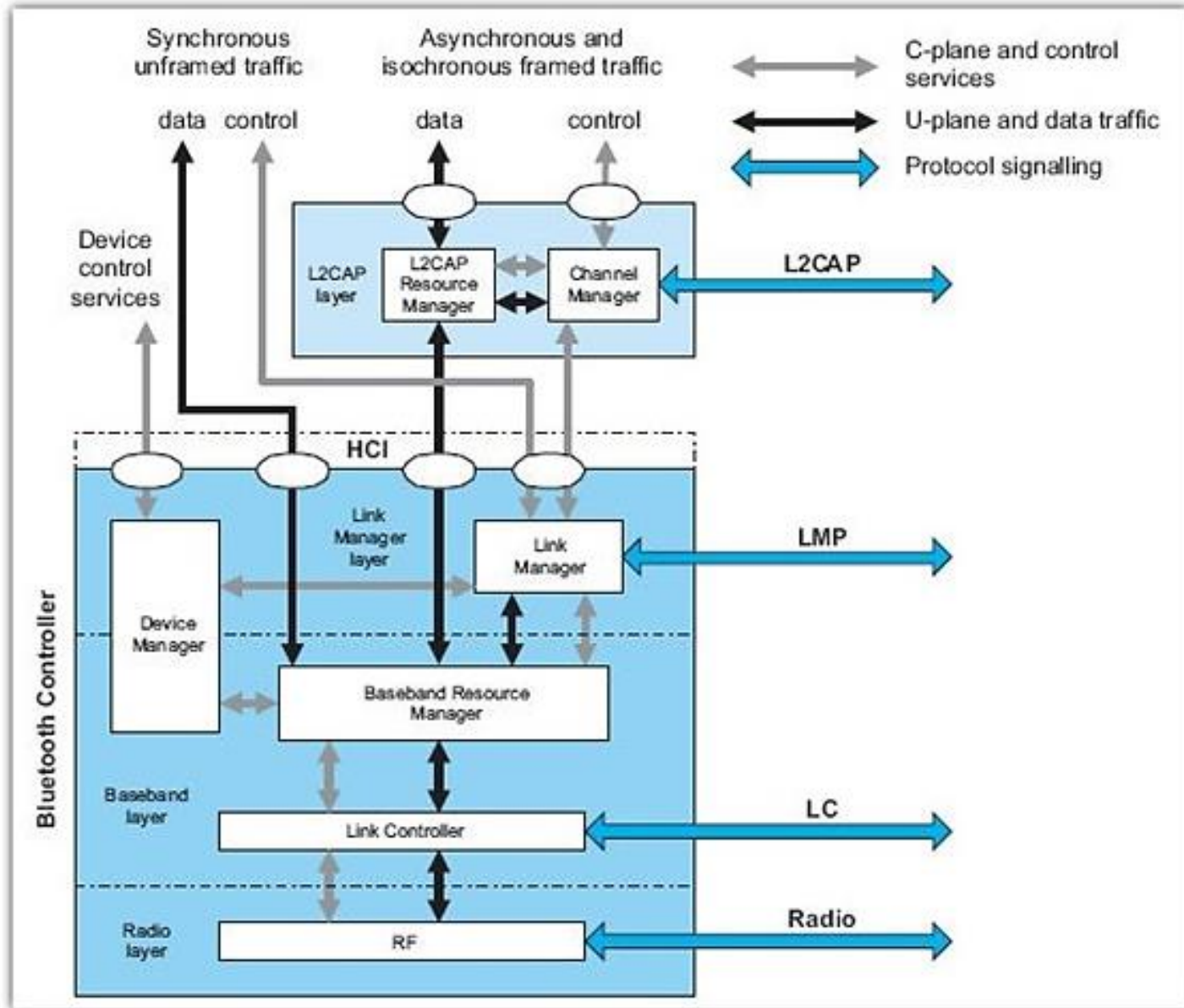
Постійне чергування частот дозволяє радіоінтерфейсу Bluetooth транслювати інформацію по всьому діапазону ISM та уникнути впливу перешкод з боку пристроїв, що працюють у цьому діапазоні. Якщо цей канал зашумлений, то система перейде на інший, і так відбуватиметься доти, доки не виявиться канал, вільний від перешкод.



На рисунку показана частотно-часова площа, що ілюструє одночасну роботу трьох Bluetooth-модулів. **Модулі працюють тактами (слотами) тривалістю 625 мкс.** Кожному модулю в межах кожного такту призначається відповідний частотний канал та режим передачі або прийому.

Частотно-часова діаграма роботи модулів Bluetooth

8.8 Архітектура Bluetooth



Структурна схема архітектури Bluetooth

8.8 Архітектура Bluetooth

Важливою частиною архітектури є **Host to Controller інтерфейс (HCI)**, що забезпечує взаємодію програмної підсистеми **Host** із апаратною підсистемою **Controller**. Вся взаємодія верхніх рівнів Bluetooth-системи з її апаратною частиною відбувається через HCI-команди, які ініціюються драйвером.

Основні блоки архітектури:

RF. Блок **Radio** займається перетворенням бітової послідовності в радіо сигнали. Питання модуляції, спектральних характеристик та фізики процесів забезпечення бітової швидкості – все це вирішується на нижньому рівні моделі.

Baseband Layer = Link Controller + Baseband Manager + Device Manager.

Рівень **Baseband** представлений у вигляді трьох блоків, спільне завдання яких полягає в управлінні фізичними каналами, поверх яких встановлюються фізичні з'єднання. Bluetooth-адресація, синхронізація генераторів пристроїв, керування кодами доступу до фізичних каналів, пошук пристроїв і встановлення фізичного каналу між ними – все це завдання **Baseband**-рівня.

Link Manager.

Після того, як два нижні рівні забезпечили фізичним з'єднанням між пристроями, справа стає за організацією логічних каналів, які згодом стануть базою для передачі трафіку додатків. **Link Manager** відповідає за встановлення, зміну та звільнення логічних з'єднань між пристроями, а також за оновлення параметрів фізичних з'єднань. Для цього **Link Manager** використовує **Link Management** протокол (**LMP**).

L2CAP Layer = Channel Manager + L2CAP Resource Manager.

Це високорівневий блок **Bluetooth Host**, окупований рівнем **L2CAP**. **Logical Link Control and Adaptation Protocol (L2CAP)** – протокол, що працює поверх створених логічних з'єднань, що забезпечує сегментацію та відновлення пакетних даних від усіх вищерозміщених додатків.

8.9 Класи і профілі Bluetooth

Залежно від потужності та ефективного радіусу дії приймачі Bluetooth поділяються на 3 класи:

Class 1 – системи моніторингу та управління промисловим обладнанням, найбільш «далекобійні»;

Class 2 – найбільш поширений варіант, який застосовується в більшості мобільних електронних пристроях і ПК;

Class 3 – малопотужні системи, якими оснащується медична апаратура.

Будь-який пристрій, обладнаний інтерфейсом **Bluetooth**, підтримує заданий виробником набір **профілів**. Кожен профіль забезпечує підтримку певних функцій (наприклад, передачу файлів або потоку медіаданих, забезпечення з'єднання з мережею тощо), які можуть бути задіяні при підключенні двох або більше пристроїв за допомогою Bluetooth. Таким чином, **набір профілів визначає функціональні можливості пристрою, доступні через з'єднання Bluetooth.**

Щоб задіяти з'єднання Bluetooth для виконання певної задачі, потрібна наявність підтримки відповідного профілю як у ведучого, так і у веденого пристрою. Так, передати через Bluetooth-з'єднання список контактів з одного мобільного телефону на інший можна лише за умови, що обидва апарати підтримують профіль OPP (Object Push Profile). А, наприклад, для використання мобільного телефону як бездротовий мобільний модем необхідно, щоб цей апарат і застосовуваний комп'ютер підтримували профіль DUN (Dial-up Networking Profile). Якщо ж Bluetooth-з'єднання між двома пристроями встановлено, але виконати будь-яку дію (скажімо, передати файл) не вдається, то ймовірною причиною цієї проблеми може бути відсутність підтримки відповідного профілю в одного з пристроїв.

Існує велика кількість різноманітних профілів Bluetooth, які описують різні варіанти та способи використання підключених пристроїв. **Кожен профіль Bluetooth обов'язково містить таку інформацію: залежність від інших профілів; пропонований формат інтерфейсу користувача; частини стека протоколів Bluetooth, які застосовуються цим профілем.**

8.9 Класи і профілі Bluetooth

Все різноманіття профілів можна розділити на дві групи: **базові та прикладні**. Далі наведено коротку інформацію про **три базові профілі**:

GAP (Generic Access Profile) – це спільний профіль доступу Bluetooth. Підтримується всіма без винятку Bluetooth-пристроями і є базисом для функціонування всіх інших профілів;

SPP (Serial Port Profile) – профіль емуляції послідовного порту. Базується на профілі GAP і описує механізм обміну даними між двома пристроями, аналогічний тому, що задіяний при підключенні через провідний послідовний інтерфейс (RS-232, USB і ін.);

GOEP (Generic Object Exchange Profile) – загальний профіль обміну об'єктами, що базується на GAP та SPP. Описує механізм обміну даними між двома пристроями з використанням протоколу передачі OBEX (OBject EXchange) та вимоги до об'єктів, що передаються.

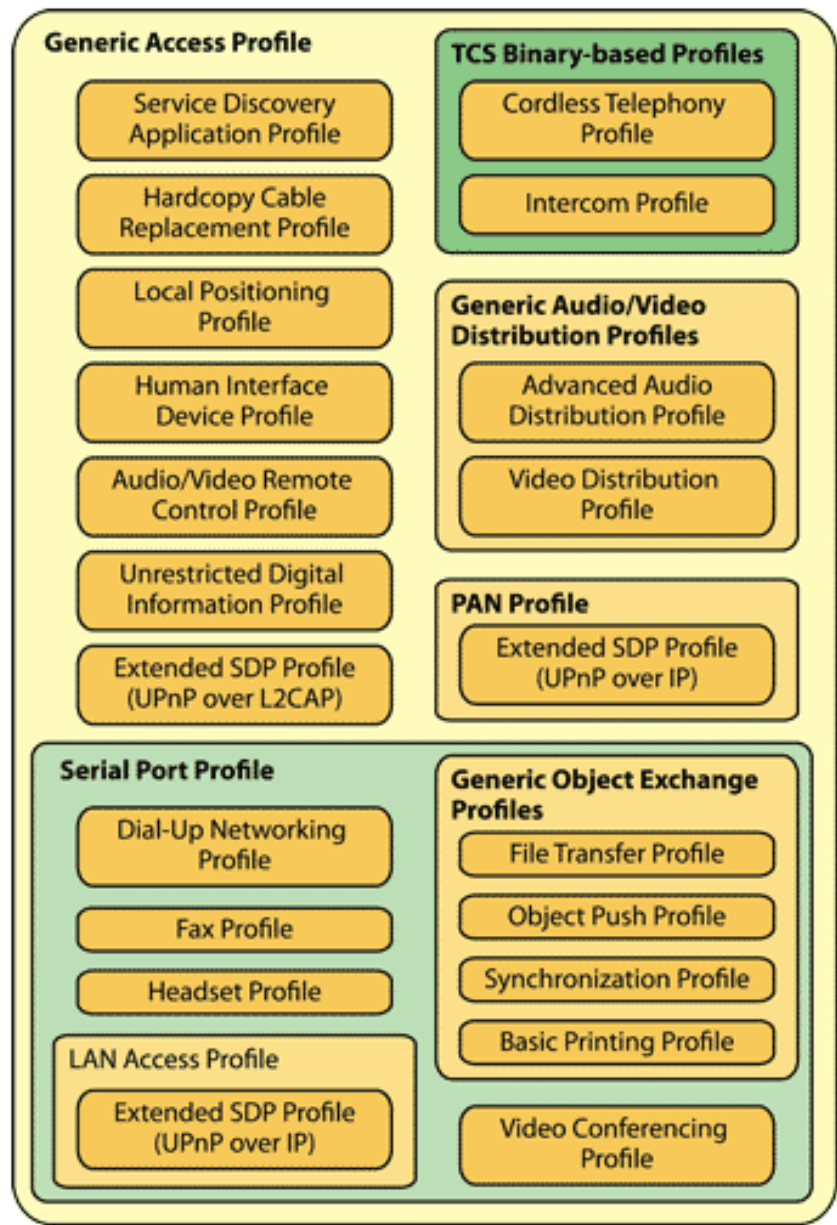
В даний час існує велика кількість **прикладних профілів, що забезпечують роботу різних функцій**.

Деякі з них, які набули найбільшого поширення в ПК, периферійних пристроях та сучасних гаджетах:

A2DP (Advanced Audio Distribution Profile) – забезпечує передачу двоканального (стереофонічного) аудіопотоку від джерела сигналу (ПК, плеєра, мобільного телефону) до бездротової стереогарнітури або іншого пристрою, що відтворює. Для стиснення потоку, що передається, може використовуватися стандартний кодек SBC (Sub Band Codec) або інший, визначений виробником пристрою;

BIP (Basic Imaging Profile) – забезпечує можливість передачі, прийому та перегляду зображень. Наприклад, дозволяє передавати цифрові фотографії із цифрової камери з пам'яті мобільного телефону. Передбачена можливість зміни розмірів і форматів зображень, що передаються, з урахуванням специфіки підключених пристроїв;

8.9 Класи і профілі Bluetooth



HID (Human Interface Device Profile) – описує протоколи та способи підключення бездротових пристроїв введення (мишей, клавіатур, джойстиків, пультів дистанційного керування тощо) до ПК. Профіль HID підтримується в ряді моделей мобільних телефонів та КПК, що дозволяє застосовувати їх як бездротові пульти для управління графічним інтерфейсом ОС або окремими додатками на ПК;

DUN (Dial-up Networking Profile) – цей профіль, що базується на SPP, забезпечує підключення ПК або іншого пристрою до Інтернету за допомогою мобільного телефону, що виконує в даному випадку функцію зовнішнього модему;

PAN (Personal Area Networking Profile) – дозволяє об'єднати два або кілька пристроїв у локальну мережу. У такий спосіб можна підключити кілька ПК до одного, який має доступ до Інтернету. Крім того, цей профіль забезпечує віддалений доступ до ПК, що виконує функції провідного пристрою;

HFP (Hands-Free Profile) – забезпечує підключення автомобільних пристроїв hands-free до мобільного телефону для голосового зв'язку.

Тема 9. Розподілені системи на основі CAN інтерфейсу

9.1 Загальна характеристика CAN.

9.2 Фізичний рівень CAN:

- ✓ топологія CAN шини;
- ✓ електричне з'єднання пристроїв з CAN інтерфейсом;
- ✓ структура типового трансівера CAN;
- ✓ типи синхронізації CAN шини;
- ✓ електричні характеристики станів CAN шини;
- ✓ лінійне кодування сигналів в CAN шині.

9.3 Канальний рівень CAN:

- ✓ адресація та доступ до шини CAN;
- ✓ достовірність передачі;
- ✓ типи фреймів шини CAN;
- ✓ пауза між кадрами і фільтрування повідомлень;
- ✓ валідація повідомлень і типи помилок CAN.

9.1 Загальна характеристика CAN

CAN (Controller Area Network) являє собою **комплекс стандартів для побудови розподілених промислових мереж**, який використовує послідовну передачу даних в реальному часі з дуже високим ступенем надійності і захищеності. **Центральне місце в CAN займає протокол канального рівня моделі OSI**. CAN був розроблений для автомобільної промисловості, але в даний час швидко впроваджується в область промислової автоматизації. Початок розвитку CAN було покладено компанією Bosch (1983), перші мікросхеми CAN контролерів були випущені Intel і Philips (1987), в даний час контролери та трансівери CAN випускаються багатьма фірмами (Analog Devices, Atmel, Dallas Semiconductor, Intel, National Semiconductor, та інші).

В даний час CAN підтримується 11-ю стандартами ISO, в тому числі ISO - Diagnostics. **CAN охоплює два рівня моделі OSI: фізичний і канальний**. Стандарт не передбачає ніякого протоколу прикладного рівня моделі OSI. Тому для його втілення в життя різні фірми розробили кілька таких протоколів: CANopen (CiA), SDS (Honeywell Micro Switch Division), CAN Kingdom (фірми Kvaser), DeviceNet (фірми Allen-Bradley, став Європейським стандартом) і ряд інших.

CAN характеризується наступними основними властивостями:

- 1) для кожного повідомлення (а не пристрою) встановлюється свій пріоритет;**
- 2) гарантована величина паузи між двома актами обміну;**
- 3) гнучкість конфігурації і можливість модернізації системи;**
- 4) широкомовний прийом повідомлень з синхронізацією часу;**
- 5) несуперечливість даних на рівні всієї системи;**
- 6) допустимість кількох провідних пристроїв в мережі;**
- 7) здатність до виявлення помилок і сигналізації про їх наявність;**

9.1 Загальна характеристика CAN

8) автоматичний повтор передачі повідомлень, доставлених з помилкою, відразу, як тільки мережа стане вільною;

9) автоматичне розпізнавання збоїв і відмов з можливістю автоматичного відключення модулів, які несправні.

Таблиця – CAN відповідно до моделі OSI

№	Назва рівня	Підрівні CAN	Примітка
7	Прикладний		Стандартом CAN не встановлений. Визначений стандартами CANopen, DeviceNet, SDS, CAN Kingdom та ін.
6	Представлення	Ні	Ні
5	Сеансовий	Ні	Ні
4	Транспортний	Ні	Ні
3	Мережевий	Ні	Ні
2	Канальний (передачі даних)	LLC	Підтвердження фільтрації, інформування про перевантаження, управління відновленням даних
		MAC	Формування пакетів даних, кодування, управління доступом, виявлення помилок, сигналізація про помилки, підтвердження прийому, перетворення з послідовної форми до паралельної і зворотно
1	Фізичний	Фізичний	Забезпечення надійної передачі на рівні байтів (кодування, контрольна сума, часові діаграми, синхронізація). Вимоги до лінії передачі

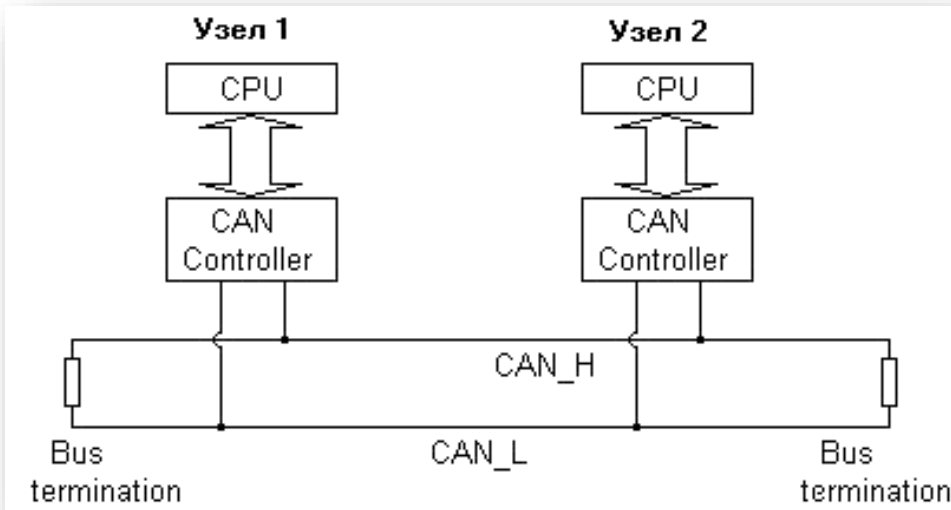
Примітка: **MAC** - Medium Access Control - «управління доступом до каналу»; **LLC** - Logical Link Control - «управління логічними зв'язками»

До недоліків можна віднести порівняно високу вартість CAN-пристроїв, відсутність єдиного протоколу прикладного рівня, а також надмірну складність і заплутаність протоколів канального і прикладного рівнів, викладених в стандартах організації CAN in Automation (CiA).

9.2 Фізичний рівень CAN

Кабель витой пари в мережі CAN повинен мати загальний (третій) провід; на обох кінцях крученої пари повинні бути узгоджувальні резистори, опір яких дорівнює хвильовому опору кабелю.

Максимальна довжина кабелю – 1 км. Для збільшення довжини, кількості вузлів або гальванічної розв'язки можуть бути використані повторювачі інтерфейсу, мережеві мости і шлюзи.



Топологія CAN шини

Управління виконується шляхом заряду ємностей затворів вихідних транзисторів від джерел струму, при цьому величина струму задається зовнішнім резистором. **Збільшення тривалості фронту дозволяє знизити вимоги до узгодження лінії на низьких частотах, збільшити довжину відводів та послабити випромінювання електромагнітних завад.**

Виводи "земля" всіх передавачів мережі повинні бути з'єднані (якщо інтерфейси гальванічно не ізольовані). При цьому різниця потенціалів між виводами заземлення не повинна перевищувати 2 В. Електрична ізоляція рекомендується при довжині лінії більше 200 м, але не є обов'язковою вимогою стандарту.

Вита пара може бути в екрані або без, в залежності від електромагнітної обстановки. **Топологія мережі повинна бути шинної, максимальна довжина відводу від шини при швидкості передачі 1 Мбіт/с не повинна перевищувати 30 см.**

Основні вимоги до лінії передачі і її характеристикам близькі до RS-485, проте в передавачах CAN є **режим управління тривалістю фронтів імпульсів.**

9.2 Фізичний рівень CAN

Для електричного з'єднання пристроїв з CAN інтерфейсом стандарт передбачає два варіанти.

Перший варіант полягає в застосуванні Т-образних розгалужувачів, які складаються з трьох 9-штирькових роз'ємів D-sub, розташованих в одному корпусі, однойменні контакти яких з'єднані між собою. Розгалужувачі мають один роз'єм зі штирями і два – з гніздами. **Другий варіант** вимагає наявності в кожному CAN-пристрої двох роз'ємів. Для включення пристрою в мережу кабель розрізають і на його кінцях встановлюють відповідні частини роз'ємів. Пристрій включається буквально в розрив лінії передачі. Такий підхід дозволяє нарощувати кількість пристроїв і змінювати топологію мережі шляхом додавання в розрив кабелю нових пристроїв і кабелю з роз'ємами на кінцях. Один з роз'ємів повинен бути зі штирями, другий - з гніздами. **Підключення пристроїв до шини без роз'ємів не допускається.** Узгоджувальний резистор повинен розташовуватися всередині роз'єму, який підключається до кінця кабелю. Для приєднання модулів до CAN-шини повинен використовуватися 9-контактний роз'єм типу D-Sub. На модулі встановлюється роз'єм з гніздами, на кабелі, що з'єднує - зі штирями.

Стандарт встановлює наступні швидкості обміну: 1 Мбіт/с, 800 кбіт/с, 500 кбіт/с, 250 кбіт/с, 125 кбіт/с, 50 кбіт/с, 20 кбіт/с.

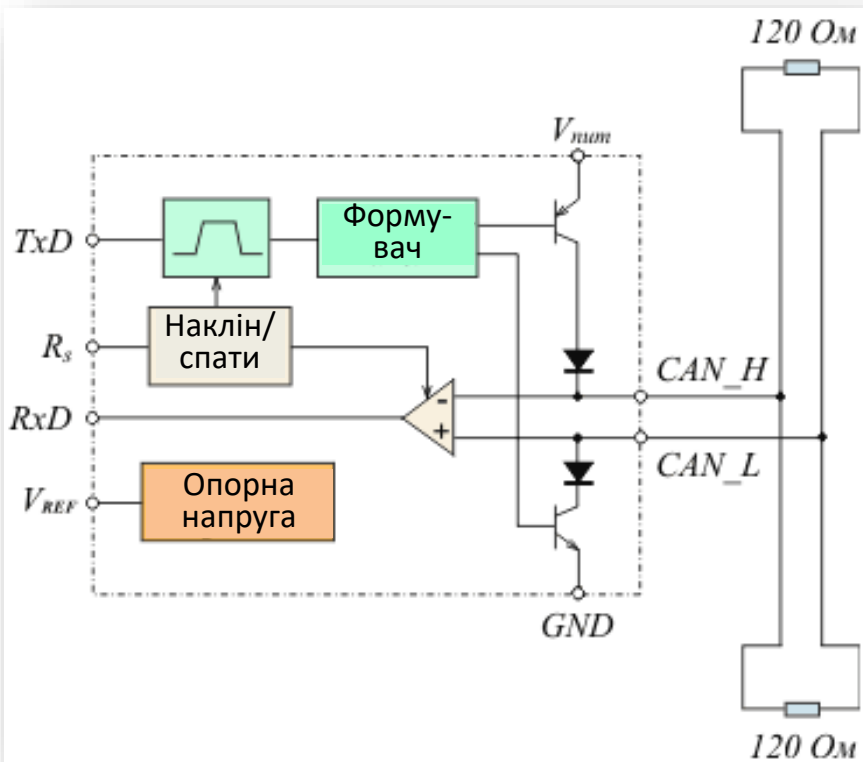
Цоколювка роз'ємів D-Sub для CAN

Контакт	Сигнал	Примітка
1	-	Зарезервований
2	CAN_L	Сигнал лінії
3	CAN_GND	«Земля»
4	-	Зарезервований
5	(CAN_SHLD)	Екран кабеля (не обов'язково)
6	(GND)	«Земля» (не обов'язково)
7	CAN_H	Сигнал лінії
8	-	Зарезервований
9	(CAN_V+)	Зовнішнє живлення (не обов'язково, для живлення передавачів з гальванічною ізоляцією)

Примітка: В кожному модулі контакти 3 і 6 повинні бути з'єднані

9.2 Фізичний рівень CAN

Типова структура трансівера CAN (на прикладі IC LT1795 фірми Linear Technology) приведена на рисунку. При подачі рівня лог. 0 на вхід TxD (вхід є інвертуючим) обидва транзистора вихідного каскаду передавача відкриваються і через навантаження (два резистора по 120 Ом) тече струм, що створює в лінії стан, що відповідає лог. 1. При цьому потенціал виводу CAN_H завжди буде вище, ніж виводу CAN_L. При логічній одиниці на вході передавача його вихід переходить в високоомний стан і диференціальне напруження на лінії стає рівною нулю. Наявність термінальних резисторів в CAN необхідно не тільки для узгодження лінії (як у випадку RS-485), але ще і для створення шляху протікання струму.



CAN передавач має дуже важливу властивість: якщо один з передавачів встановлює в мережі лог. 0, а другий – лог. 1, то цей стан не є аварійним, як в мережі на основі RS-485, оскільки наскрізного струму не виникає. У разі CAN лінія залишається в стані лог. 1. Інакше кажучи, лог. 1 завжди домінує над лог. 0. Тому в стандарті CAN використовується поняття "домінантний стан", стан лінії для позначення стану з струмом, і поняття "рецесивний стан" як протилежний домінантному.

Це властивість CAN забезпечує можливість отримання доступу до лінії, посылаючи в лінію логічні рівні, порівнюючи їх з тим рівнем, який фактично встановлюється в ній: якщо передавач посилає в лінію рецесивний стан, а в ній при цьому залишається домінантний, значить лінія зайнята.

9.2 Фізичний рівень CAN



Доступ отримує той вузол мережі, який може надати їй доміантний рівень сигналу. Вузли з рецесивним рівнем залишають лінію і чекають наступного випадку. Цей метод доступу справедливий і при використанні оптоволоконного каналу або бездротової мережі – в цих випадках наявність світла або електромагнітної хвилі завжди буде домінувати над їх відсутністю.

Вивід Vref дозволяє встановити граничну напругу для входу TxD і рівень синфазної напруги в лінії, коли вона знаходиться в рецесивному стані. Зазвичай Vref=2,5 В.

Щоб встановити рівень синфазної напруги на лінії, термінальні опори ділять на два по 60 Ом, з'єднують їх послідовно, а до точки з'єднання підключають вивід Vref. При симетричній формі імпульсів CAN_H і CAN_L щодо рецесивного стану зменшується рівень випромінюваних перешкод, оскільки збільшення струмів в кожному з проводів крученої пари при перемиканні логічних рівнів виявляються рівними за величиною, але зворотними по знаку і тому компенсують один одного.

Вивід Rs має кілька призначень. Якщо на ньому встановлено стан лог. 1, трансівер переходить в сплячий режим, при якому він споживає дуже малий струм від джерела живлення, а на виході встановлюється високоомний (рецесивний) стан. "Розбудити" його можна сигналом, що надходять в приймач з лінії. Підключення цього виводу до "землі" через опір дозволяє встановити потрібну тривалість фронтів імпульсів передавача. Деякі трансівери мають два режими: резервний і сплячий, які відрізняються рівнем споживаного струму і способом переходу в активний режим.

Режим зниженого енергоспоживання передбачений стандартом для економії заряду акумуляторних батарей в припаркованому автомобілі.

9.2 Фізичний рівень CAN

Якщо сигнал TxD є домінуючим надто довго (більше 1 мс), генератор імпульсу “timeout” (на схемі прямокутником з імпульсом) тимчасово відключає передавач, оскільки в іншому випадку модуль може бути назавжди блокований засобами канального рівня як такий, що відмовив.

Стандартом передбачена можливість підключення до CAN мережі будь-якої кількості пристроїв, проте практично воно обмежується здатністю навантаження передавачів (100...200) або затримкою в повторювачах.

У CAN-трансівері є генератор синхроімпульсів з частотою 16 МГц $\pm 0,1\%$. Ширина одного біта програмно встановлюється величиною від 8 до 25 імпульсів синхрогенератора, зазвичай 8 імпульсів при швидкості передачі 1 Мбіт/с та 16 імпульсів при 20 кбіт/с. Синхронізація всіх вузлів мережі відбувається протягом першого такту синхронізації. Процедура обробки бітів у приймачі забезпечує програмовану затримку імпульсів синхронізації, необхідну для компенсації часу затримки проходження сигналу лінії зв'язку і зсуву фази внаслідок дрейфу частоти тактового генератора.

Розрізняють два типи синхронізації: жорстку синхронізацію за допомогою стартового біта на початку повідомлення та ресинхронізацію під час передачі повідомлення. За допомогою ресинхронізації можна підлаштувати інтервал часу від початку синхронізації до моменту, коли вимірюється логічний рівень прийнятого імпульсу даних. Інтервал підстроювання може бути змінено на 1...4 такти.

Для визначення логічного стану шини рівні сигналів, що приймаються, вимірюються на відстані 6-ти тактів синхрогенератора від переднього фронту імпульсу (біта) при швидкості 1 Мбіт/с і на відстані 14-ти тактів при швидкості 20 кбіт/с (для порівняння вкажемо, що у стандартних UART відліки беруться посередині імпульсу). Кількість відліків може бути 1 чи 3 (встановлюється програмно). **CAN використовує синхронну передачу бітів. Це підвищує пропускну здатність каналу зв'язку, але потребує ускладненого процесу синхронізації.**

9.2 Фізичний рівень CAN

Параметр	Позначення	Од. вимірювання	Мін.	Ном.	Макс.	Умова
Для рецесивного стану шини						
Потенціали на виході передавача	CAN_H	В	2,0	2,5	3	Без навантаження
	CAN_L	В	2,0	2,5	3	
Диференційна напруга на виході передавача	Vdiff	мВ	-500	0	50	Без навантаження
Диференційна напруга на вході приймача	Vdiff	В	-1	-	0,5	Без навантаження
Для домінантного стану шини						
Потенціали на виході передавача	CAN_H	В	2,75	3,5	4,5	З навантаженням
	CAN_L	В	0,5	1,5	2,25	
Диференційна напруга на виході передавача	Vdiff	В	1,5	2	3	З навантаженням
Диференційна напруга на вході приймача	Vdiff	В	-0,9	-	5	З навантаженням

Напруга живлення пристроїв у мережі CAN повинна становити від 18 до 30 В. Вихідна напруга на 9-му контакті роз'єму (зовнішня позитивна напруга живлення) повинна бути від +7 до +13 В при струмі споживання модуля не більше 100 мА. Не допускається, щоб модулі були джерелами струму.

9.2 Фізичний рівень CAN

CAN використовує NRZ кодування (Non Return-to-Zero), при якому лог. 0 відповідає низький рівень напруги в лінії (рецесивний стан), лог. 1 – високий рівень (домінантний стан).

Такий спосіб має наступний недолік: у випадку, коли через лінію передачі транспортується байт, який містить всі одиниці (пауз між ними при кодуванні NRZ немає), приймач не може відрізнити цей байт від паузи.

Для усунення цієї проблеми використовується так званий **біт-стафінг**. Після кожної послідовності з 5-ти однакових символів поспіль вставляється протилежний їм символ. Наприклад, після 5 одиниць поспіль вставляється логічний нуль. Приймач, виявивши 5 однакових символів поспіль, видаляє символ, який є бітом стафінгу.

Іншими властивостями CAN-трансіверів, які передбачені у стандарті, є:

- ✓ захист від короткого замикання проводів інтерфейсу між собою, джерело живлення або землю. З цих вимог автоматично впливає захист від зміни полярності підключення приймача та передавача до лінії, обривів та передавлення кабелю;
- ✓ захист від електростатичних розрядів;
- ✓ ослаблення синфазного сигналу в лінії;
- ✓ захист від перегріву вихідних каскадів.

9.3 Канальний рівень CAN

Канальний рівень CAN складається з двох підрівнів: LLC та MAC.

Адресація та доступ до шини.

У мережі CAN жоден з вузлів немає адреси. Натомість повідомлення посилаються "всім", але містять ідентифікатор, який описує зміст даних, що посилаються. **Відповідно до цього значення будь-який вузол мережі може прийняти це повідомлення, якщо воно необхідне пристрою для функціонування.** Повідомлення приймається вузлом, якщо його ідентифікатор проходить через фільтр повідомлень, що є у кожному вузлу.

У мережі CAN гарантується, що повідомлення буде прийнято будь-яким з вузлів в один і той же час або не буде прийнято жодним з них. Це досягається завдяки широкомовній передачі та використаним методом підтвердження прийому повідомлень.

Коли мережа вільна, будь-який вузол може розпочати надсилання повідомлення. Але **кожне повідомлення має пріоритет при отриманні доступу до шини.** Тому передачу може здійснити лише один пристрій – той, який містить повідомлення із найвищим пріоритетом.

Боротьба доступу до шини відбувається так. Якщо два або більше пристроїв виявили, що лінія вільна і почали передачу повідомлень одночасно, то **конфлікт, що виник, вирішується шляхом побитового порівняння ідентифікатора повідомлення, що передається, зі станом лінії.** У процесі арбітражу (врегулювання конфлікту) кожен пристрій порівнює логічний рівень біта, що передається, з логічним рівнем на шині. Якщо рівні однакові, пристрої продовжують передавати наступний біт ідентифікатора. Якщо приймач пристрою показує, що на шині домінуючий рівень, а передавач в цей же час передає рецесивний рівень, пристрій відразу припиняє передачу цього повідомлення. Такий механізм арбітражу гарантує, що ні інформації, ні часу не буде втрачено.

9.3 Канальний рівень CAN

Достовірність передачі.

Для досягнення максимальної надійності (достовірності) передачі даних протокол передбачає спеціальні методи виявлення помилок, сигналізації про помилки та самоконтролю, які втілені у кожному вузлі мережі.

Для виявлення помилок вжито наступних заходів:

- ✓ передавач порівнює кожен біт на шині з переданим бітом для підтвердження правильності передачі на рівні бітів;
- ✓ виконується контроль циклічним надлишковим кодом (CRC – Cyclic Redundancy Check);
- ✓ використовується біт-стафінг;
- ✓ використовується перевірка кожного переданого кадру.

Механізм виявлення помилок характеризується такими властивостями:

- ✓ виявляються усі глобальні помилки;
- ✓ виявляються всі помилки, що вносяться передавачем;
- ✓ у повідомленні виявляються до 5 випадково розподілених помилок;
- ✓ у повідомленнях виявляється пакет наступних один за одним помилок завдовжки до 15 біт;
- ✓ виявляються помилки парності.

Імовірність наявності у повідомленні невиявлених помилок становить менше $4,7 \times 10^{-11}$.

Повідомлення з виявленими помилками позначаються прапором у тому вузлі, де їх було виявлено. Такі повідомлення відхиляються та автоматично надсилаються повторно. Час від моменту виявлення помилки до початку повторної передачі дорівнює тривалості 31 біта, якщо не виникають нові помилки. CAN здатний розрізняти збої та відмови. Якщо виникла відмова, то, відключається від мережі той пристрій, який відмовив. Усі приймачі мережі перевіряють цілісність (несуперечність) отриманих повідомлень, підтверджують (квітують) цілісні повідомлення та позначають прапором суперечливі повідомлення.

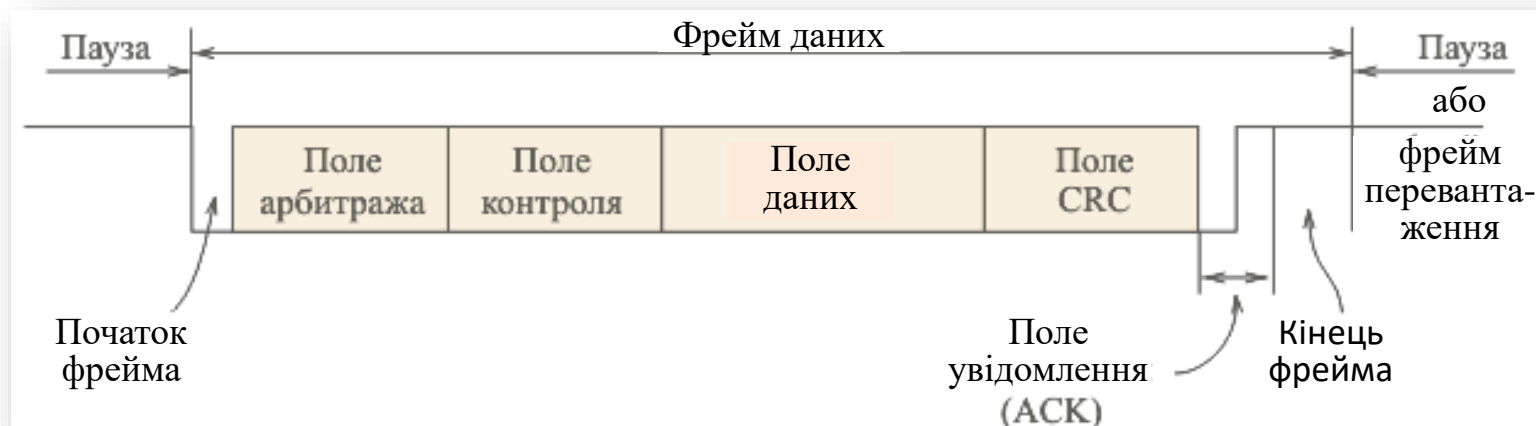
9.3 Канальний рівень CAN

Повідомлення CAN передаються за допомогою **фреймів (блоків даних)**. Використовується два різних формати фреймів, які відрізняються довжиною поля ідентифікатора: стандартний фрейм з ідентифікатором довжиною 11 біт і розширений фрейм з довжиною ідентифікатора 29 біт.

Існує 4 різних типи фреймів:

- ✓ DATA FRAME – "фрейм даних" – переносить дані від передавача до приймача;
- ✓ REMOTE FRAME – "дистанційний фрейм" (фрейм виклику) – передається одним із пристроїв для того, щоб отримати від іншого пристрою дані у форматі DATA FRAME з тим же ідентифікатором, що і в REMOTE FRAME;
- ✓ ERROR FRAME – "фрейм помилок" – передається будь-яким пристроєм, що виявив помилку на шині;
- ✓ OVERLOAD FRAME – "фрейм перевантаження" – використовується для запиту додаткової затримки між попередніми та наступними даними.

Фрейм даних складається з наступних полів: початок фрейму (Start Of Frame), поле арбітражу (Arbitration Field), поле контролю (Control Field), поле даних (Data Field), поле циклічного надлишкового коду (CRC Field), поле повідомлення про прийом (ACKnowledgement Field) та поле кінця фрейму (End Of Frame). Поле даних може мати нульову довжину.



9.3 Канальний рівень CAN

Простір між кадрами представлений рецесивним станом шини (що відповідає високому рівню, оскільки CAN-передавачі інвертують логічні рівні). Тільки при рецесивному стані шини пристрій може розпочати передачу фрейму. **Початок фрейму** кодується одним доміантним бітом. Всі пристрої мережі одночасно синхронізують свої приймачі на передньому фронті імпульсу цього біта.

Формат поля арбітражу відрізняється для стандартного та розширеного формату фрейму. У стандартному кадрі поле арбітражу складається з ідентифікатора довжиною 11 біт та RTR-біта (Remote Transmission Request - "запит дистанційної передачі"). У розширеному форматі поле арбітражу має ідентифікатор довжиною 29 біт, SRR-біт (Substitute Remote Request - "замінний RTR-біт"), IDE-біт (Identifier Extension Bit - "біт ідентифікації розширеного формату") та RTR-біт. Поле ідентифікатора у розширеному форматі складається з базового ідентифікатора та розширеного ідентифікатора. Базовий ідентифікатор визначає пріоритет розширеного кадру. RTR-біт служить для того, щоб відрізнити фрейм даних від фрейм виклику. IDE-біт служить для розрізнення стандартного та розширеного формату фреймів.

Поле контролю включає код, який вказує довжину даних в полі даних, IDE-біт і один (у стандартному форматі) або два (у розширеному) зарезервованих біта.

Поле даних складається з даних, які мають бути передані фреймом даних. Він може мати довжину від 0 до 8 байт по 8 біт кожен. Дані передаються молодшим розрядом уперед.

Поле CRC містить циклічний надлишковий код, що служить для виявлення помилок у всіх попередніх полях кадру, включаючи біт початку кадру. Поле CRC закінчується CRC-розділювачем (рецесивний стан) завдовжки 1 біт. Стандарт CAN встановлює алгоритм обчислення CRC. Біти стафіну перед обчисленням видаляються.

Поле повідомлення має довжину 2 біти. Передавальний пристрій у цьому полі посилає два рецесивні біти. Пристрій відповідає доміантним бітом, якщо повідомлення прийнято без помилок. Другий біт цього поля завжди є рецесивним.

Кінець фрейму є послідовністю з семи рецесивних біт.

9.3 Канальний рівень CAN

Фрейм виклику виконує функцію запиту даних. Він аналогічний кадру даних, але відрізняється від нього лише відсутністю поля даних та іншими значеннями бітів.

Фрейм помилок використовується будь-яким приймаючим вузлом, щоб повідомити всім учасникам мережі про те, що повідомлення, що передається по мережі, містить помилку. Першим полем у кадрі помилок є прапор помилки. Повідомлення про помилку має найвищий у системі пріоритет, тому передається відразу після виявлення помилки та приймається всіма пристроями одночасно. Всі пристрої також одночасно видаляють із своєї пам'яті повідомлення, що містить помилку.

Фрейм перевантаження складається з двох полів: прапора перевантаження та поля роздільника. Існують такі умови, при настанні яких починається передача кадру перевантаження: перевантаження приймача, що вимагає збільшити паузу між кадрами, які він приймає; виявлення домінантного біта на місці першого і другого біта в полі перерви паузи між кадрами.

Пауза між кадрами.

Між фреймами даних, фреймом виклику та будь-якими іншими фреймами встановлюється пауза. На відміну від цього, перед фреймами перевантаження та помилок паузи немає, це прискорює їх доставку. Пауза містить поле перерви (3 біти) та поле простою (довільної довжини) і для пасивних до помилки пристроїв, які виконували передачу попереднього повідомлення, поле призупиненої передачі.

Фільтрування повідомлень.

Фільтрування повідомлень використовується для вибору з усіх повідомлень на шині лише тих, які відповідають масці, записаний у реєстр приймача. Маска може бути налаштована на відбір групи повідомлень та використовує ідентифікатор, що входить до складу поля арбітражу. Відібрані повідомлення розміщуються в буфер приймача.

9.3 Канальний рівень CAN

Валідація повідомлень.

Під валідацією розуміється встановлення факту, що повідомлення не містить помилок. Момент часу, коли встановлюється факт правильності повідомлення, відрізняється для передавача і приймача. Повідомлення вважається достовірно переданим, якщо не було помилок під час передачі від початку до кінця кадру. Якщо повідомлення містить помилку, воно автоматично повторюється відповідно до поточних пріоритетів. Повідомлення вважається достовірно прийнятим, якщо не було виявлено помилок під час його прийому. Якщо виявлено помилку, пристрій надсилає в шину прапор помилки.

У CAN розглядається 5 типів помилок:

- ✓ помилки передачі біта (контролюється рівень на шині і порівнює з переданим. Помилка виявляється під час передачі одного біта);
- ✓ помилка стафінгу (виявляється за відсутності біта стафінгу в 6-й позиції послідовності однакових бітів);
- ✓ CRC-помилка;
- ✓ помилка формату (виявляється, якщо заздалегідь фіксований формат фрейму поле з відомим значенням бітів містить неправильні біти);
- ✓ помилка повідомлення (виявляється трансівером, якщо він не знаходить домінуючий стан у полі повідомлення про отримання).

Пристрій, який виявить будь-яку з перелічених помилок, сигналізує про це за допомогою прапора помилки.

Тема 10. Розподілені системи на базі інтерфейсу USB

- 10.1 Загальний опис специфікацій інтерфейсу USB.
- 10.2 Фізичні пристрої на шині USB.
- 10.3 Топологія шини USB.
- 10.4 Типи передач в USB шині.
- 10.5 Поняття кінцевої точки інтерфейсу USB.
- 10.6 Структура кадрів і транзакцій інтерфейсу USB.
- 10.7 Типи пакетів інтерфейсу USB.
- 10.8 Фізичні властивості інтерфейсу USB.

10.1 Загальний опис специфікацій інтерфейсу USB

Шина або інтерфейс USB (Universal Serial Bus) – універсальна послідовна шина, призначена для підключення периферійних пристроїв. У свій час шина USB прийшла на зміну вже морально застарілим інтерфейсам COM, LPT, PS/2 та GamePort (для підключення джойстика).

Стандарт USB 1.0, який набув широкого поширення, був представлений у 1996 році. Він визначав **два швидкісні режими передачі**: режим Full-Speed з максимальною швидкістю передачі до 12 Мбіт/с і режим Low-Speed з максимальною швидкістю передачі до 1,5 Мбіт/с. Максимальна довжина кабелю в режимі Full-Speed не перевищувала 3 м, а в режимі Low-Speed – 5 м.

Для підключення периферійних пристроїв до шини USB 1.0 використовувався кабель, що складається із двох пар дротів. Одна кручена пара в диференціальному включенні використовувалася для прийому та передачі даних, а ще одна пара проводів - для живлення периферійного пристрою напругою +5 В. Завдяки вбудованим лініям живлення шина USB 1.0 дозволяла підключати периферійні пристрої без власного джерела живлення. При цьому максимальна сила струму, що споживається пристроєм по лініях живлення шини USB 1.0, не повинна була перевищувати 0,5 А. До одного контролера USB 1.0 шини можна було під'єднати до 127 пристроїв через ланцюжок концентраторів.

У 2000 році було опубліковано специфікацію USB 2.0, а наприкінці 2001 року цю версію було стандартизовано. **Сьогодні шина USB 2.0 є найпоширенішою**. Всі ноутбуки, материнські плати та периферійні пристрої (принтери, БФП, миші, клавіатури, джойстики, графічні планшети тощо) оснащуються інтерфейсом USB 2.0, а інтерфейс USB 1.1 (1.0) вже практично вийшов із вжитку. У той же час інтерфейс USB 2.0 є сумісним з інтерфейсом USB 1.1 (1.0).

10.1 Загальний опис специфікацій інтерфейсу USB

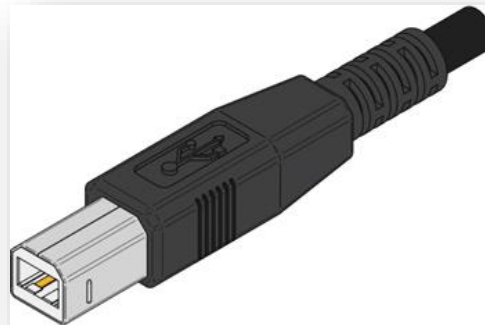
У специфікації USB 2.0 передбачено три швидкісні режими: крім режимів Low-Speed (до 1,5 Мбіт/с) і Full-Speed (до 12 Мбіт/с), є високошвидкісний режим Hi-Speed, що визначає максимальну швидкість передачі до 480 Мбіт/с з (60 Мбайт/с).

Як і у разі специфікацій USB 1.0 та USB 1.1, у специфікації USB 2.0 для підключення периферійних пристроїв використовується кабель, що складається з двох пар проводів: одна кручена пара проводів для прийому та передачі даних, а інша – для живлення периферійного пристрою. При цьому напруга живлення знову становить 5 В, а максимальна сила струму не повинна перевищувати 0,5 А. До одного контролера шини USB 2.0 можна приєднувати до 127 пристроїв через ланцюжок концентраторів.

У стандарті USB 2.0 передбачено три типи роз'ємів: роз'єм типу А, роз'єм типу В та роз'єм miniUSB типу В. Згодом, у січні 2007 року, були представлені роз'єми microUSB, проте вони призначені для мобільних телефонів та комунікаторів та не використовуються у комп'ютерах та периферійних пристроях.



Роз'єм USB типу А



Роз'єм USB типу В



Роз'єм miniUSB

10.1 Загальний опис специфікацій інтерфейсу USB

Основна проблема шини USB 2.0 полягає в тому, що вона є односпрямованою. Тобто дані передаються в обидві сторони (або одну, або в іншу, але не одночасно) по одній і тій же крученій парі. Незважаючи на максимальну швидкість передачі даних до 480 Мбіт/с, шина USB 2.0 має досить **великі затримки між запитом на передачу даних і власне початком передачі.** Тому насправді інтерфейс USB 2.0 не дозволяє передавати дані зі швидкістю більше 35 Мбайт/с (280 Мбіт/с). Звісно, така швидкість передачі явно недостатня. Підключення за інтерфейсом USB 2.0 зовнішніх жорстких дисків і флеш-пам'яті не дозволяє реалізувати весь їхній швидкісний потенціал. Наприклад, при підключенні зовнішнього жорсткого диска за інтерфейсом eSATA швидкість передачі даних обмежується швидкісними характеристиками самого диска і для сучасних дисків становить близько 90 Мбайт/с (при послідовних операціях читання та запису), а при підключенні того ж зовнішнього диска за інтерфейсом USB 2.0 передачі даних знижується до 33 Мбайт/с. Не тільки зовнішні накопичувачі, але і багато інших периферійних пристроїв потребують значно більшої пропускної здатності інтерфейсу, ніж може надати інтерфейс USB 2.0. Тому з'явився високошвидкісний стандарт USB 3.0.

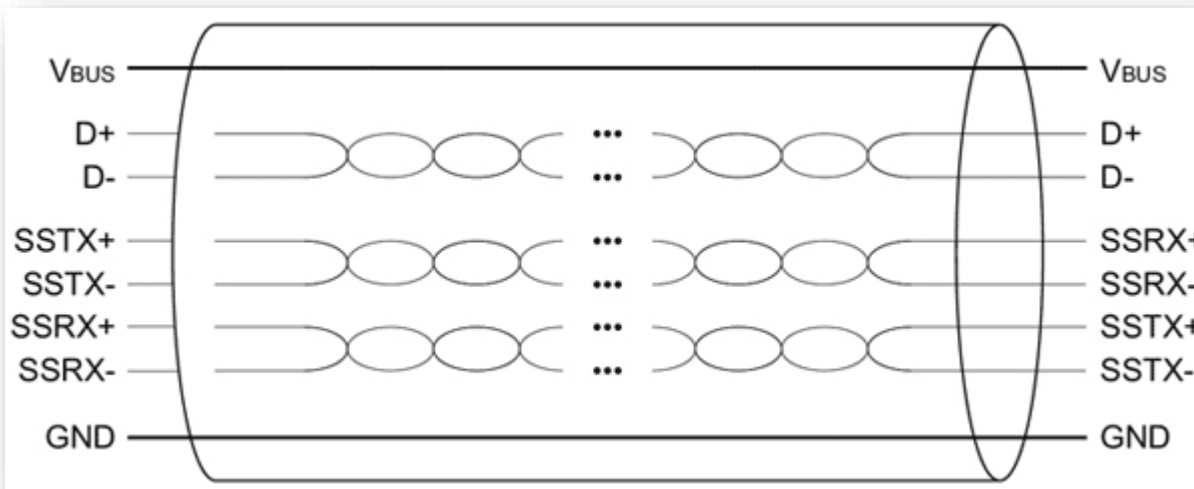
У роботі над новим стандартом брали участь компанії Hewlett-Packard Company, Intel Corporation, Microsoft Corporation, NEC Corporation, ST-NXP Wireless та Texas Instruments.

Головна перевага стандарту USB 3.0 у порівнянні зі стандартом USB 2.0, то це швидкість передачі даних. Специфікацією USB 3.0 передбачений режим SuperSpeed зі швидкістю передачі даних до 5 Гбіт/с (640 Мбайт/с), тобто більш ніж у 10 разів, що перевищує ту, що передбачена специфікацією USB 2.0. Реальна швидкість передачі даних буде нижчою, проте навіть з урахуванням цього для більшості периферійних пристроїв такої пропускної здатності шини USB 3.0 більш ніж достатньо. Висока швидкість передачі у шини USB 3.0 – це не єдина її відмінність від шини USB 2.0. Але **при всіх відмінностях зберігається зворотна сумісність USB 3.0 з USB 2.0.** Тобто в специфікації USB 3.0, окрім режиму SuperSpeed, залишені і режими Hi-Speed, Full-Speed та Low-Speed.

10.1 Загальний опис специфікацій інтерфейсу USB

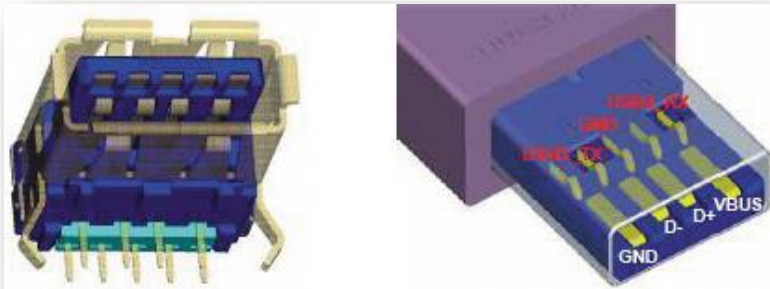
Роз'єми USB 3.0 сумісні з роз'ємами USB 2.0. Всі периферійні пристрої з інтерфейсом USB 2.0 можна підключати до USB 3.0.

Наступна важлива відмінність специфікації USB 3.0 від USB 2.0 полягає в тому, **що стандартом передбачено двонаправлену передачу даних з різних витих пар**. Таким чином, контактів у роз'ємах USB 3.0 більше. Збільшилася кількість проводів в USB-кабелі. Так, у стандарті USB 2.0 одна кручена пара використовувалася для прийому/передачі даних, а інша - для живлення, тобто в роз'ємах USB 2.0 було чотири контакти, а USB-кабель містив чотири дроти. **У специфікації USB 3.0 перша кручена пара застосовується для передачі даних, друга - для прийому даних, третя – для живлення, а для сумісності зі стандартом USB 2.0 передбачена четверта кручена пара, за якою здійснюється прийом/передача даних в режимі USB 2.0 (Hi-Speed, Full-Speed та Low-Speed)**. Крім того, в обов'язковому порядку є ще одна «земля» (GND_DRAIN) у вигляді оплетення двох кручених пар. Таким чином, у кабелі USB 3.0 не чотири дроти (як у кабелі USB 2.0), а вісім, а в роз'ємах USB 3.0 як мінімум дев'ять контактів (оплетки кручених пар підключаються до контакту GND_DRAIN).



Напруга живлення, що передається USB-шиною, залишилася без змін - +5 В. Кількість можливих роз'ємів USB 3.0 побільшало. Є роз'єми USB 3.0 наступних типів: типу А, типу В, Powered-В, Micro-В, Micro-А та Micro-АВ.

10.2 Фізичні пристрої на шині USB



Роз'єм USB 3.0 типу А, як штепсельний, так і гніздовий, за своєю формою та розмірами не відрізняється від роз'єму USB 2.0 типу А. Він повністю сумісний з роз'ємом USB 2.0 типу А, тобто в гніздовий роз'єм USB 3.0 типу А, крім штепсельного роз'єму USB 3.0 типу А, можна вставляти штепсельний роз'єм USB 2.0 типу А, і навпаки.

Фізичні пристрої на шині USB бувають трьох типів: **хост-контролер, хаб та кінцевий пристрій**.

Хост-контролер – це головний керуючий шиною USB. Саме він забезпечує зв'язок пристроїв, підключених до шини, з комп'ютером (з ОС та клієнтським ПЗ). Будь-які сеанси обміну даними може починати тільки хост-контролер, інші пристрої мовчать не активні, доки хост-контролер до них не звернеться.

Контролер взаємодіє з ОС через драйвер хост-контролера (HCD - host controller driver). Цей драйвер прив'язаний до конкретної моделі хост-контролера. Тільки він знає, які дані, в які регістри і в якому порядку завантажувати в хост-контролер, а також звідки які дані брати, щоб хост-контролер зробив те, чого від нього потребується.

З боку ОС шиною USB керує ще один драйвер – **USB D (Universal Serial Bus Driver)**, який вирішує загальні (неспецифічні для конкретного хост-контролера) питання: взаємодія з клієнтським ПЗ, нумерація пристроїв на шині, їх конфігурування, розподіл живлення та пропускної спроможності шини і так далі. Це, можна сказати, своєрідний диспетчер, який здійснює загальний контроль над шиною та її взаємодію із зовнішнім світом (з клієнтським ПЗ).

Для спілкування з підлеглими пристроями хост-контролер має спеціальні помічників – хаби (їх ще іноді називають концентраторами).

10.3 Топологія шини USB

Хаби – це пристрої, які дозволяють фізично підключити USB-пристрої до шини. Вони надають порти для підключення, ретранслюють трафік від хост-контролера до кінцевих пристроїв та назад, відстежують стан та фізично керують електроживленням портів. У хабів є один висхідний (upstream) порт – це той порт, який підключений до хост-контролера, і кілька низхідних (downstream) портів – це порти, до яких підключаються кінцеві пристрої. Хаби можна каскадувати, підключаючи до низхідного порту хаба ще один хаб. Найголовніший хаб, інтегрований з хост-контролером, називається кореневим хабом (він же – кореневий концентратор або root hub).

У хаба є два основних завдання: 1) створити хост-контролеру ілюзію, що він безпосередньо розмовляє з підключеним до хабу пристроєм; 2) спостерігати за своїм сегментом шини (за девайсами, підключеними до низхідних портів), повідомляти «вгору» про всі зміни і, якщо треба, підключати та відключати живлення портів.

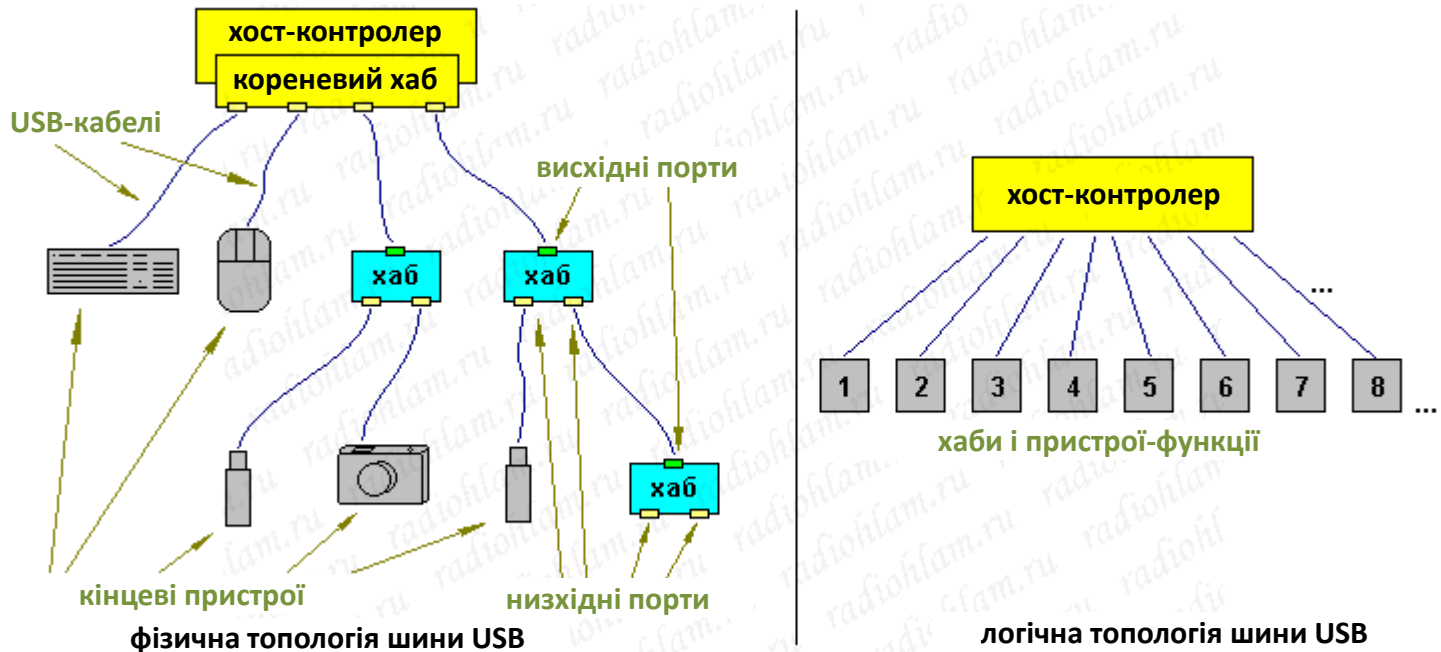
Кінцеві пристрої – це всі корисні пристрої, які підключаються до шини USB (флешки, принтери, мишки і т.д.)

Потрібно сказати, що **фізичні пристрої та логічні пристрої – це не завжди те саме**. Існують, наприклад, такі кінцеві пристрої (звані складовими – compound devices), які містять у собі хаб, до якого підключено ще кілька пристроїв. Незважаючи на те, що в цьому випадку хаб і всі підключені до нього пристрої заповнені в один корпус, з точки зору логіки шини це будуть різні пристрої.

Для кінцевих логічних пристроїв зазвичай використовують термін «функції». Таким чином, з погляду логіки шини, пристрої на ній можна розділити на хаби та функції. Кожен логічний пристрій на шині має унікальну адресу (1-127), що присвоюється хостом при підключенні.

10.3 Топологія шини USB

Виходячи з описаного вище, фізична топологія USB – дерево (хаби можна каскадувати), а логічна топологія – зірка, центром якої є хост-контролер.



Логічне пристрій є набір кінцевих точок (endpoints чи EP). Фізично, кінцеві точки – це різні буфери в логічному пристрої USB, через які відбувається обмін даними з хостом. Для різних завдань зручно мати різні буфери. Пристрій може виконувати паралельно кілька різних завдань. (мінімум два — відстежувати команди управління від хоста і виконувати завдання користувача.) У цих різних завдань можуть бути різні ступені важливості, вимоги до надійності, своєчасності та швидкості доставки даних, джерела та споживачі інформації, що пересилається, також можуть бути різні (джерелом і споживачем корисної інформації є клієнтський драйвер, в той же час будь-яка управляюча інформація йому не потрібна).

10.4 Типи передач в USB шині

Типи передач в USB шині:

- 1) ізохронні передачі (isochronous transfers). Вони призначені для передачі поточкових даних у реальному часі. Такі передачі гарантують час доставки, але не гарантують, що всі дані будуть доставлені. Якщо під час передачі відбувається помилка, дані просто губляться. Крім того, для передач такого типу має бути попередньо узгоджено, яку частину пропускної спроможності шини ця передача займатиме. Ізохронні передачі мають найвищий пріоритет та мають право зайняти до 90% пропускної спроможності каналу. Передачі цього типу використовуються, наприклад, для відеокамер або колонок;
- 2) переривання (interrupts). Цей тип призначений для спонтанних невеликих повідомлень, але з гарантованим часом обслуговування та гарантованою доставкою. Прикладом може бути USB клавіатура;
- 3) передача масивів даних (bulk data transfers). Для цього типу немає жодних гарантій за швидкістю, єдине в чому можна бути впевненим – що дані дійдуть у цілості та безпеці. Такі передачі мають найнижчий пріоритет, але їм нічого не треба узгоджувати, скільки залишиться вільною від інших типів передач ширини каналу, стільки вони й займуть. Не залишиться взагалі – чекатимуть, коли канал звільниться. Такі передачі можна використовувати для обміну даними з повільними пристроями, наприклад, з принтерами;
- 4) керуючі передачі (control transfers). Це передачі типу запит-відповідь. З їх допомогою передаються команди управління пристроями. Тут важлива як безпомилкова передача, так отримання відповіді результати виконання команди. Крім того, оскільки ці передачі є службовими, їм гарантовано 10% пропускної спроможності каналу.

10.5 Поняття кінцевої точки інтерфейсу USB

Для того, щоб відрізнити одну точку від іншої, кінцеві точки повинні мати унікальний номер. Крім номера, кожна кінцева точка має ще й напрямок. IN – якщо точка призначена для передачі даних хосту, OUT – якщо точка призначена для прийому даних від хоста. Точки з однаковими номерами, але з різними напрямками передачі даних – це різні з погляду логіки шини кінцеві точки.

Єдиний виняток – кінцева точка EP0. Вона має особливий статус. Вона є службовою та призначена для загального управління пристроєм (конфігурування, налаштування тощо). Крім того, ця кінцева точка двонаправлена і вона повинна обов'язково бути присутньою в будь-якому USB-пристрої.

Виходячи з усього вищеописаного, для ідентифікації якоїсь кінцевої точки на шині, потрібно знати адресу пристрою, до якого відноситься кінцева точка, її номер у пристрої та напрямок передачі даних через цю точку.

Крім того, введені такі поняття, як **інтерфейс, конфігурація та альтернативні установки**. Інтерфейс поєднує кінцеві точки, призначені для вирішення будь-якої однієї задачі. Набори інтерфейсів, що використовуються одночасно, називаються конфігураціями. Альтернативні установки дозволяють включати або відключати якісь кінцеві точки, які входять у конфігурацію, залежно від способу вирішення завдань, для яких призначена ця конфігурація. Самих конфігурацій та альтернативних установок кожної з цих конфігурацій для одного логічного пристрою може існувати кілька, але в кожний момент часу тільки один з цих наборів може бути активним. Причому хост повинен знати, який саме набір активний і відповідно забезпечувати зв'язок з кінцевими точками, що входять в цей набір. Інші кінцеві точки, що не входять до активного набору, не будуть доступні для зв'язку.

10.5 Поняття кінцевої точки інтерфейсу USB

Для зв'язку клієнтського ПЗ з кожною активною кінцевою точкою хост створює **комунікаційний канал (communication pipe)**. Клієнтське ПЗ, яке хоче поспілкуватися з кінцевою точкою, має відправити до відповідного каналу **пакет запиту введення/виведення (IRP – input/output request packet)** та чекати на повідомлення про завершення його обробки. У IRP вказується лише адреса буфера, куди треба складати або звідки брати дані та довжина передачі. Все інше зроблять хост і драйвери, що обслуговують його (USB D і HCD).

Залежно від типу передач, що використовуються в каналі, комунікаційні канали поділяються на два типи: потокові (streaming pipes) та канали повідомлень (message pipes).

Комунікаційний канал до точки EPO є службовим і називається основним каналом повідомлень (default pipe, control pipe 0). Власником основних каналів повідомлень всіх підключених пристроїв є драйвер USB D, оскільки, як ми вже говорили, через EPO здійснюється конфігурування та налаштування пристрою.

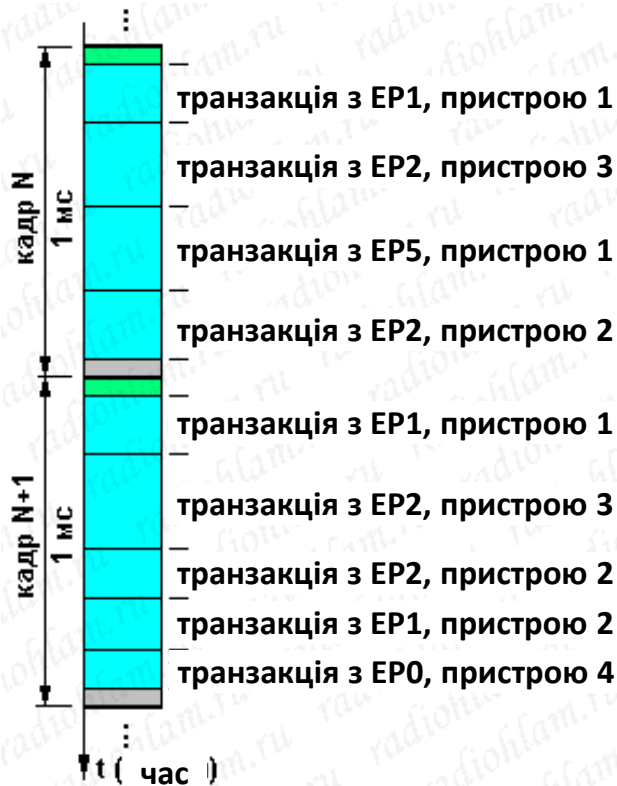
Таким чином, якщо потрібно з клієнтського ПЗ відправити якісь дані до кінцевої точки нашого пристрою. Ми надсилаємо IRP до каналу, який USB D встановив з нашою точкою і повідомляємо адресу буфера, де лежать дані, які нам потрібно надіслати, і розмір блоку даних, що пересилається. З даними ними починає працювати USB D.

USB D нарізує наш блок даних на фрагменти, які можна передати за одну транзакцію і, відповідно до пріоритету обраного нами для даної кінцевої точки типу передачі, для кожного фрагменту планує, коли він може його відправити (тобто планує транзакції). Далі, у запланований час USB D ці фрагменти посилає кінцевій точці. І таким чином усю посилку отримає кінцева точка.

Транзакція – це один сеанс зв'язку із пристроєм. Оскільки до шини у нас підключено багато пристроїв, то хост фізично не може постійно обмінюватися даними з усіма цими пристроями. Натомість він організовує цикли (фрейми, кадри), у кожному з яких здійснює кілька сеансів зв'язку з різними пристроями. Ось ці сеанси зв'язку і називають транзакціями.

10.6 Структура кадрів і транзакцій інтерфейсу USB

Кадри слідуєть один за одним з періодичністю 1 кадр у мс. В одному кадрі не обов'язково повинні бути присутні сеанси зв'язку з усіма пристроями на шині або всі фрагменти інформації, призначені для одного пристрою. Розклад транзакцій планується USBД з урахуванням пріоритету обраних типів передач і з якимись кінцевими точками хост може не здійснювати транзакцій кілька фреймів поспіль, навіть за наявності запиту на обмін даними з цими точками (принтер може і зачекати, а ось передача музики в USB колонки чекати ніяк не може).



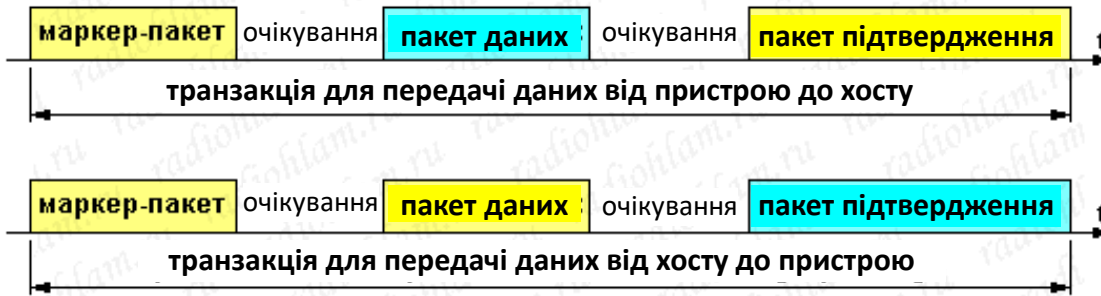
Структура кадрів і транзакцій

Зв'язок з типами передач та пропускної спроможності каналу: здатність зайняти 90% пропускної спроможності каналу для ізохронних передач означає, що у кожному кадрі 90% часу може бути відведено транзакціям тільки цього типу передач, а 10% пропускної спроможності каналу, гарантованих керуючим передачам, означають, що у кожному кадрі 10% часу гарантовано можуть зайняти транзакції керуючих передач.

Початок кожного кадру позначається посилкою **спеціального маркер-пакета SOF (start of frame)**, до складу якого входять 11 молодших біт номеру кадру. Маркер-пакет використовується для синхронізації ізохронних точок та хабів. У режимі HS кожен кадр ділиться на 8 мікрокадрів по 125 мкс, кожен з яких починається з посилки маркер-пакета SOF (при цьому в SOF всіх мікрокадрів, що належать до одного кадру, передається однаковий номер). **Інтервал часу наприкінці кожного кадру називається EOF.** EOF – це час тиші. До цього часу мають бути завершені всі транзакції. Якщо в цей час хаб виявить, що в якийсь спадний порт йому поступають дані, то він цей порт просто відключить і повідомить про це хосту.

10.7 Типи пакетів інтерфейсу USB

Транзакції складаються з пакетів. Будь-які сеанси обміну даними можуть розпочинатися лише з ініціативи хоста. Транзакція починається хостом з відправки маркер-пакета транзакції, в якому вказується адреса пристрою, адреса кінцевої точки, з якою хост хоче «поспілкуватися», а також **напрямок передачі даних**. Отримавши такий пакет, пристрій, що адресується, готується до обміну. Далі, після невеликого тайму-ауту, слідує пакет даних від джерела (джерело визначається напрямком, зазначеним у маркер-пакеті). Для ізохронних передач транзакція на цьому закінчується (оскільки їм не потрібні жодні підтвердження доставки даних). Для решти типів передач в транзакцію входить ще третій пакет – пакет підтвердження або пакет квітування (handshake).



- пакети, що передаються хостом
- пакети, що передаються пристроєм

назва поля	Sync	PID	Check	Data + CRC5/16
Його довжина в бітах	8	4	4	8 x n
	Заголовок			тіло + контрольна сума

Загальна структура пакетів (для швидкостей передачі LS/HS)

Усього існує 4 типи пакетів: маркер-пакети (token), пакети даних (data), пакети підтвердження (handshake) та спеціальні пакети (special). Ці пакети мають певну структуру, яка залежить від типу пакета, хоча у всіх пакетів можна виділити і деякі загальні поля.

Пакет можна умовно розділити на заголовок (2 байти), що має загальну для всіх пакетів структуру (Sync+PID+Check), та тіло, захищене контрольною сумою. Наявність, розмір і структура тіла, кількість біт контрольної суми залежить від типу пакета. Будь-який пакет на LS/FS починається з 8 біт синхронізації – поле Sync. Для синхронізації використовується бітова послідовність 10000000 (для HS довжина поля синхронізації – 32 біти).

10.7 Типи пакетів інтерфейсу USB

Далі слідує 4 біти ідентифікатора пакета – PID і його інверсна копія – Check. PID визначає призначення пакета та, відповідно, його наступну структуру. У таблиці нижче представлений опис різних ідентифікаторів пакетів шини USB.

Ім'я	PID	Опис
Ідентифікатори маркер-пакетів:		
SOF	0101	Ідентифікатор маркер-пакета початку кадра. Пакет з таким PID містить в тілі 11 молодших біт номера кадра, захищених контрольною сумою CRC5.
SETUP	1101	Ідентифікатор маркер-пакета транзакції управління. Пакет з таким PID містить в тілі сьомібітну адресу пристрою, чотирьохбітний номер кінцевої точки, з яким хост бажає «поспілкуватися», і контрольну суму CRC5.
OUT	0001	Ідентифікатор маркер-пакета транзакції виведення. Пакет з таким PID містить в тілі сьомібітну адресу пристрою, чотирьохбітний номер кінцевої точки, якій хост буде посилати дані, і контрольну суму CRC5.
IN	1001	Ідентифікатор маркер-пакета транзакції введення. Пакет з таким PID містить в тілі сьомібітну адресу пристрою, чотирьохбітний номер кінцевої точки, з якої хост буде отримувати дані, і контрольну суму CRC5.
Ідентифікатори пакетів даних:		
Data0	0011	Ідентифікатор пакета даних. Пакет з таким PID містить в тілі n байт даних, захищених контрольною сумою CRC16.
Data1	1011	Ідентифікатор пакета даних. Пакет з таким PID містить в тілі n байт даних, захищених контрольною сумою CRC16.

10.7 Типи пакетів інтерфейсу USB

Data2	0111	Ідентифікатори додаткових типів пакетів, які використовуються в транзакціях з широкосмуговими ізохронними точками (для HS USB 2.0)
MData	1111	
Ідентифікатори пакетів підтвердження:		
ACK	0010	Пакет з таким PID складається тільки із заголовка (тіло пакета є пустим – няких даних і контрольної суми немає) і використовується для підтвердження безпомилкового прийому пакета даних.
NAK	1010	Пакет з таким PID складається тільки із заголовка і використовується для повідомлення хосту про неготовність кінцевої точки до обміну даними (індикація зайнятості).
STALL	1110	Пакет з таким PID складається тільки із заголовка і використовується для повідомлення хосту про необхідність його втручання для розв'язання проблеми.
NYET	0110	Пакет з таким PID складається тільки із заголовка і використовується для підтвердження безпомилкового прийому і повідомлення про відсутність місця для прийому наступного пакета максимального розміру (в USB 2.0).
Ідентифікаторы спеціальних пакетов:		
PING	0100	Пакет з таким PID є пробним маркером управління потоком (USB 2.0). Таким маркером хост як би попередньо запитує пристрій, готовий чи він приймати дані, замість того, щоб відразу ці дані посилати (тому що, якщо пристрій не готовий і надішле NAK, то все відправлення прийдеться повторювати заново).

10.7 Типи пакетів інтерфейсу USB

У всіх полях пакетів, крім поля CRC, дані передаються молодшим бітом вперед.

Усі пакети складаються з цілого числа байт (розрядність полів, що входять у пакет, спеціально так підбрано, щоб сума розрядів усіх цих полів була кратна восьми).

Всі пакети закінчуються спеціальним сигналом EOP, для LS/FS цей сигнал триває 2 бітові інтервали, для HS – таким спеціальним сигналом є передача певної бітової послідовності.

Дані, що передаються по шині, **кодуються за методом NRZI з технікою вставки біт (bit stuffing)**.

Це означає, що при передачі нульового біта стан сигналу на шині змінюється на протилежне, а при передаванні одиниці стан сигналу не змінюється. Вставка біт використовується для того, щоб не втратити синхронізацію за монотонного одиничного сигналу. Суть цієї вставки зводиться до того, що після кожних 6 поспіль одиниць в дані, що передаються, вставляється нульовий біт, незалежно від того, яке значення має біт, що йде за цією групою одиниць.

Сигнал неспроможний дійти від джерела до приймача миттєво. По-перше, затримку вносять кабелі, по-друге, затримку вносять хаби (вони повинні прийняти сигнал від джерела та ретранслювати його приймачеві), по-третє, треба враховувати, що хабів у ланцюжку від хоста до пристрою може бути декілька. Крім того, приймач повинен мати деякий час, щоб «осмислити» прийнятий пакет, вирішити чи є в ньому помилки, кому він призначений і т.д. Таким чином, пакет у відповідь джерело не може отримати миттєво, тому що приймачеві потрібен час для формування відповіді, і для всієї транспортної мережі потрібен час щоб «доставити» відповідь. З іншого боку, відповідь не можна чекати нескінченно довго, раптом вона взагалі не прийде. Тому в пристроях USB нормується, по-перше, **"максимальний час обороту по шині"** – це час, за який дані повинні дістатися приймача і повернутися назад до джерела в найгіршому випадку – при проходженні послідовно через 5 хабів (час на те, щоб "доставити"), і, по-друге, **"максимальна затримка відповіді"** - максимальний час від кінця побаченого EOP до початку передачі пакету у відповідь (час на те, щоб "подумати").

10.7 Типи пакетів інтерфейсу USB

У пристроях при очікуванні пакета запускаються спеціальні таймери, які відраховують інтервал, достатній на формування відповіді та її доставки, і якщо відповіді цей час не отримано – це сприймається як помилка.

Для FS інтервал очікування становить 16-18 бітових інтервалів, для HS - 736-816 бітових інтервалів. Максимальний час, за який пристрої повинні все обміркувати і почати надсилати відповідь, становить 7,5 бітових інтервалів на FS і 192 бітових інтервалів для HS.

Тривалість бітового інтервалу для швидкості LS становить приблизно 667 нс (1,5 Мбіт/с), для FS приблизно 83 нс, для HS приблизно 2 нс.

4 ідентифікатори пакетів даних. При передачах типу bulk (масиви), control (керуючі) та interrupt (переривання) приймач після отримання пакета даних повинен надіслати назад до джерела пакет підтвердження. Цей пакет підтвердження (як і сам пакет даних) може зіпсуватися. Якщо джерело не отримує підтвердження успішної передачі пакета даних, то в наступній транзакції він повторить надсилання цього пакета. Щоб приймач зрозумів, що він ці дані вже отримувач, пакети даних надсилають з PID, що чергується.

Парні пакети посилають з PID Data0, а непарні – з PID Data1. Таким чином, отримавши двічі пакет даних з одним і тим же PID, приймач розуміє, що це не якісь нові дані, а просто повторне відправлення попереднього пакета, тому що джерело попередньої транзакції не побачило пакет підтвердження. Спеціальний біт в кінцевій точці, який вказує, пакет даних, з яким PID ми чекаємо, називається Toggle Bit.

PID Data2 і MData дозволяють розрізнити пакети даних усередині мікрокадру для широкосмгових ізохронних точок (USB2.0).

10.8 Фізичні властивості інтерфейсу USB

Для зв'язку по USB використовуються 4 дроти: +5В, D+, D- та GND. Ці дроти мають стандартне колірне маркування: червоний провід – це +5В, чорний – GND, зелений – D+, білий – D-.

Для передачі бітів використовується диференціальний сигнал між проводами D+ та D-.

Провід +5В і GND використовуються для живлення пристрою, а також для індикації деяких спеціальних станів (разом з D+ та D-).

На лініях D+ та D- високий рівень відповідає напрузі +3,3 (від 2,7 до 3,6). **Диференціальний сигнал, у якому різниця між D+ і D- більше 200 мВ при рівні напруги лінії D+ > 2В називається Diff1.** Диференціальний сигнал, у якому різниця між D- і D+ більше 200 мВ при рівні напруги лінії D- > 2В називається Diff0. Стан, коли на обох сигнальних лініях є низький рівень відносно GND (D+ < 0,8В і D- < 0,8В) називається SE0 (single-ended zero).

Для ідентифікації швидкості роботи пристрою використовується резистор на 1,5 ком, який підтягує одну з сигнальних ліній до високого сигнального рівня (+3,3В). Для LS-пристроїв підтягується лінія D-, для FS/HS-пристроїв підтягується лінія D+.

Сигнали Diff0 та Diff1 кодують стани шини, які називаються J (data J state) та K (data K state).

Для LS стан J відповідає сигналу Diff0, а стан K – сигналу Diff1. Для FS/HS навпаки J відповідає сигналу Diff1, K - сигналу Diff0. Стан спокою шини (bus idle) для LS/FS відповідає тривалому стану J, а HS – стану SE0. Ознакою початку передачі є перехід зі стану спокою в стан K.

Оскільки першим завжди передається нульовий біт (будь-яка передача починається з передачі синхропослідовності), то цей початковий стан K відповідає нульовому біту, подальше значення бітів визначається відповідно до NRZI кодування.

Для позначення кінця пакета (EOP) на LS/FS використовується сигнал SE0, тривалістю 2 бітові інтервали. На HS сигналом EOP служить передача спеціальної послідовності 11111110 (молодшим бітом уперед, не використовуючи техніку bit stuffing). Приймач відстежує порушення техніки вставки біт (сигнал на шині не змінюється протягом семи бітових інтервалів) і вважає це ознакою кінця пакета.

Тема 11. Комунікаційні інтерфейси ARM процесорів

11.1 Характеристики та архітектура ARM процесорів.

11.2 Процесори лінійки Cortex.

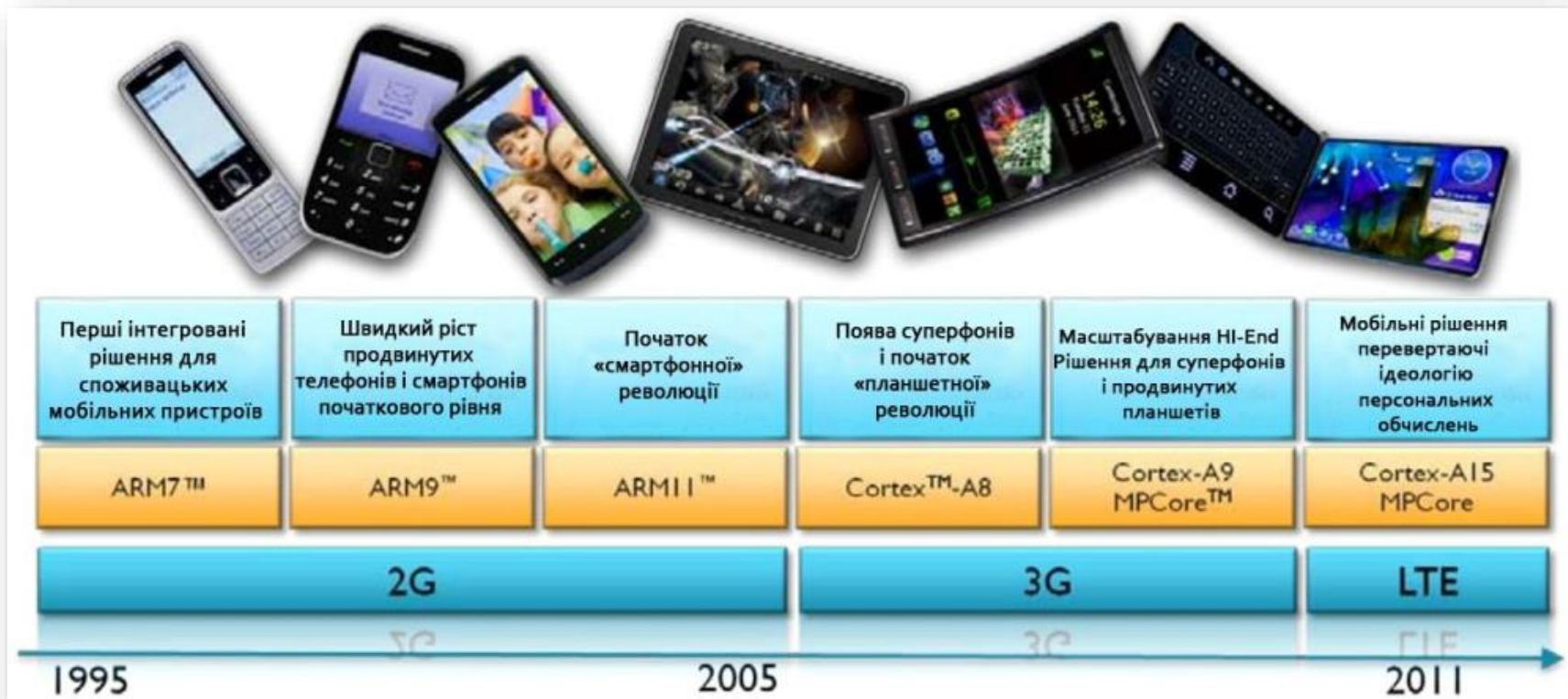
11.3 Архітектура та характеристики мікроконтролера STM32F407VG.

11.4 Вбудовані інтерфейси комунікації.

11.5 Робота з аналоговими сигналами.

11.1 Характеристики та архітектура ARM процесорів

Процесори ARM – нинішній безумовний лідер мікропроцесорного ринку мобільних і інтегрованих рішень. На рисунку зображено узагальнений шлях розвитку мобільних рішень (апаратів стільникового зв'язку, кишенькових обчислювальних пристроїв, тощо). Поряд із самим розвитком технічної бази мобільних рішень проілюстровано розвиток безпосередньо поколінь зв'язку – 2G → 3G → LTE.



Еволюція мобільних рішень

11.1 Характеристики та архітектура ARM процесорів

Необхідність в процесорі підвищеної потужності, здатному працювати з графічним інтерфейсом користувача, призвела на початку 80-х років минулого сторіччя британську компанію AcornComputers до думки про необхідність відмови від готових, але малопотужних рішень, пропонованих партнерами MOS Technology і Motorola, і запуску розробки нового власного процесора. Учасники проекту BBC Micro створили для Acorn мікропроцесор на архітектурі RISC, яка є альтернативою популярної архітектури CISC.

RISC-архітектура пропонувала оптимізацію обчислювального процесу за рахунок реалізації складних функцій не за допомогою єдиної комплексної команди, як це робилося в CISC, а за допомогою набору більш простих команд.

При такому підході арифметико-логічний пристрій істотно спрощується, що дозволяє додати в схему процесора більшу кількість регістрів. **Збільшення кількості регістрів знижує необхідність частого звернення до повільною оперативної пам'яті.**

У фірмі ARM обрали вектор розвитку мікропроцесорів спрямований у бік технологій ASIC і ASSP.

Технологія ASSP (Application-Specific Standard Products) – розробка простих, але в той же час універсальних по застосуванню компонентів, наприклад, апаратних декодерів звуку і відео.

Технологія ASIC (Application-Specific Integrated Circuit) на противагу ASSP – створення інтегральних мікросхем, що спеціалізуються на вирішенні деякого обмеженого кола завдань. До ASIC-рішень можна віднести роутери, мобільні телефони й ігрові консолі.

11.2 Процесори лінійки Cortex

Завдяки поєднанню технологій **ASIC** та **ASSP** можна за дуже короткий термін **створити новий пристрій довільної конфігурації**, адаптований саме для використання під конкретно визначені задачі.

Єдину архітектуру ARMv7 для всієї лінійки Cortex компанія ARM розділяє варіанти процесорних ядер Cortex по областях їх застосування.

В результаті процесори лінійки Cortex розділені на три класи:

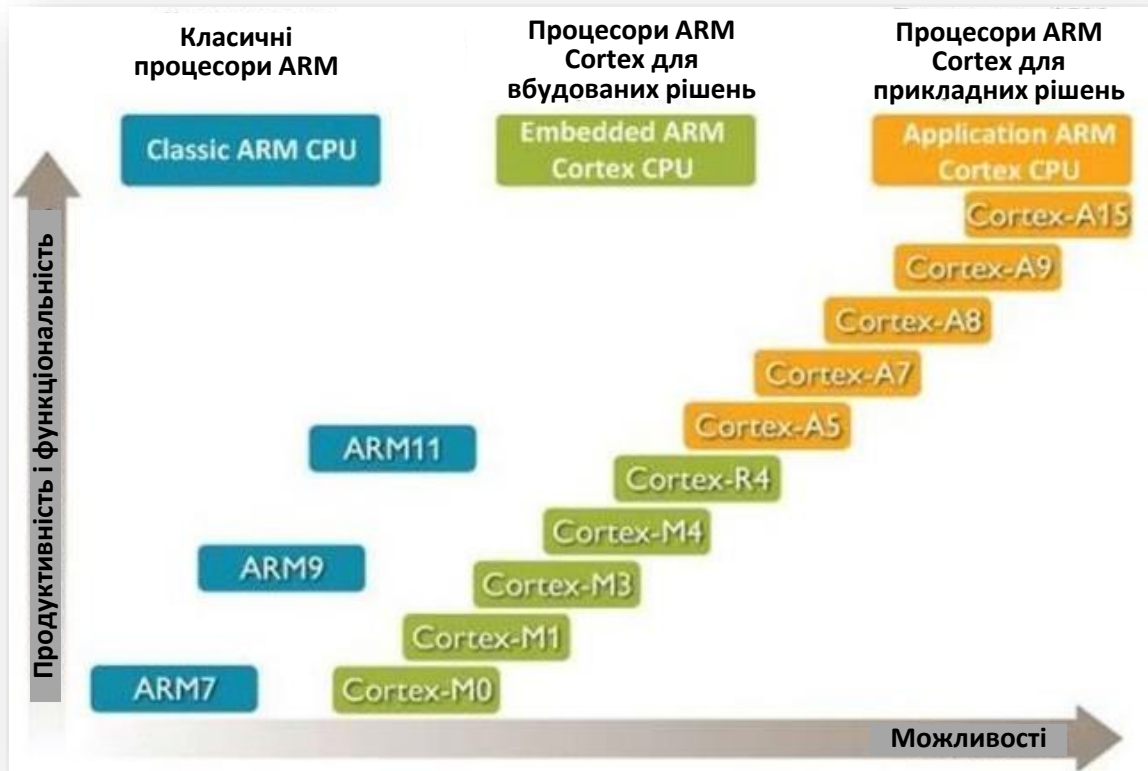
1. **Cortex-A** (від Application) – **прикладний**, серія процесорів, що надають широкий спектр рішень для пристроїв, що **виконують складні обчислювальні завдання**, такі як хостинг платформи операційної системи (ОС) мобільних пристроїв (rich OS) (найпоширенішими є Linux і його похідні Android) і підтримка декількох програмних додатків. Ядрами Cortex-A оснащені процесори більшості мобільних пристроїв, таких як смартфони та планшети. У даному сегменті ринку знаходяться кілька виробників інтегральних схем: від тих, хто продають каталог компонентів (TI або Freescale), до тих, хто виробляє процесори для інших ліцензіатів. Серед найбільш поширених ядер у цьому сегменті можна виділити 32-розрядні процесори Cortex-A7 та Cortex-A9, а також найновіші високопродуктивні 64-розрядні ядра Cortex-A53 та Cortex-A57.

11.2 Процесори лінійки Cortex

2. **Cortex-M** (від eMbedded) – **вбудований**, являє собою лінійку **масштабованих, сумісних, енергоефективних і простих у використанні** процесорів, призначених для недорогого ринку вбудованих пристроїв. Сімейство Cortex-M оптимізовано для чутливих до вартості та енергоспоживання мікроконтролерів, що підходять для таких додатків, як Інтернет речей (Internet of Things, IoT), зв'язок, керування двигуном, інтелектуальний облік, пристрої взаємодії з людиною (human interface devices, HID), автомобільні та промислові системи управління, побутова техніка, споживчі товари та медичні інструменти. У цьому сегменті ринку знаходяться багато виробників інтегральних схем, які виробляють процесори Cortex-M: ST Microelectronics є одним із них.

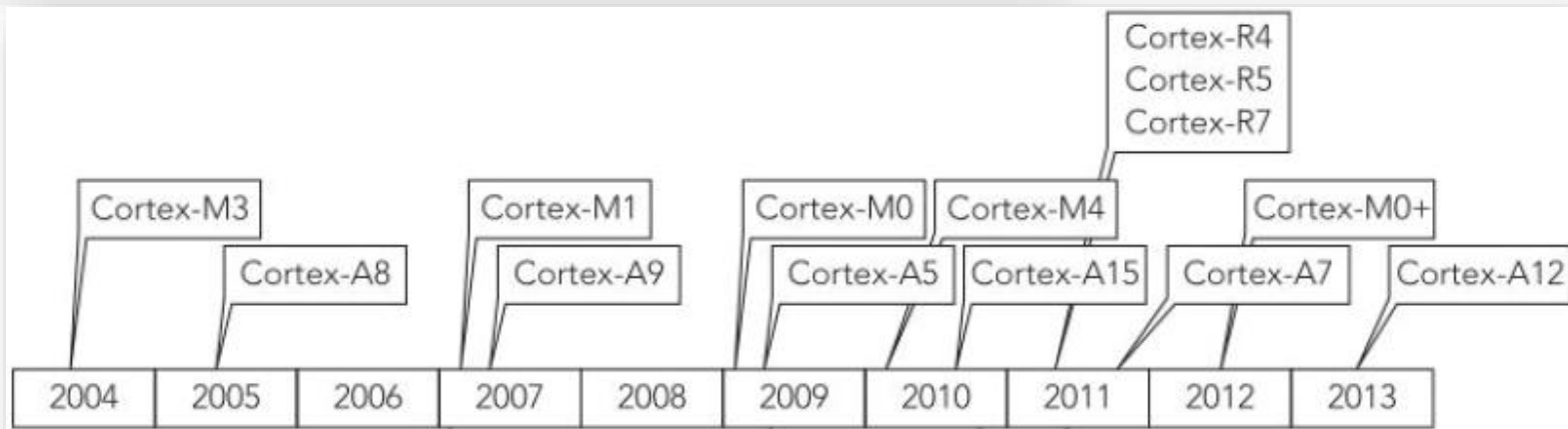
3. **Cortex-M** (від Realtime) – **реального часу**, являє собою серію процесорів, що пропонують **високопродуктивні обчислювальні рішення для вбудованих систем**, де необхідні надійність, висока доступність, стійкість до відмов, ремонтпридатність і детермінований відгук в реальному часі. Процесори серії Cortex-R забезпечують швидку та детерміновану обробку та високу продуктивність при одночасному вирішенні складних завдань у режимі реального часу. Вони об'єднують ці функції у корпусі, оптимізованому за продуктивністю, енергоспоживанням та займаною площею, що робить їх вірним вибором у надійних системах, вимогливих до відмовостійкості.

11.2 Процесори лінійки Cortex



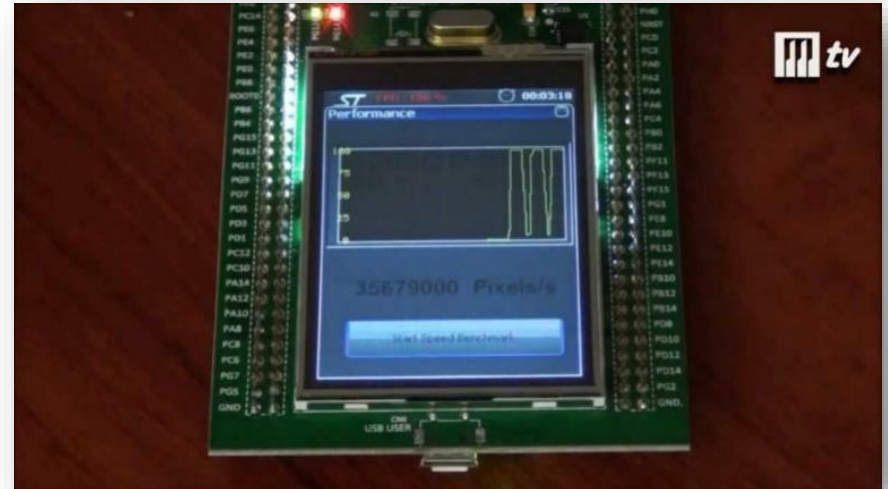
Поділ процесорів ARM на класи

Графік виходу ядер сімейства Cortex



11.3 Архітектура та характеристики мікроконтролера STM32F407VG

Кожна наступна лінійка процесорів ARM підтримує технологічні рішення попередників і включає в себе нові технології.



Зовнішній вигляд STM32F407VG (зліва) та STM32F407VG з TFT-драйвером (зправа)

Серія мікроконтролерів STM32F4 є розширенням платформи STM32 заснованим на останній версії ядра ARM Cortex-M4. У цій серії з'явилися нові можливості у сфері обробки сигналів і більш швидкі за часом виконання операції.

11.3 Архітектура та характеристики мікроконтролера STM32F407VG

Основні характеристики цього сімейства пристроїв:

Блок ядра (ARMCortex-M4 168 MHz):

- ✓ Ядро ARM 32-bit Cortex-M4 CPU;
- ✓ Частота тактування 168 МГц, 210 DMIPS / 1.25 DMIPS/МГц (Dhrystone 2.1);
- ✓ Підтримка DSP-інструкцій;

Блок ART прискорювача (ART Accelerator);

Блок високопродуктивної АНВ-матриці шин (Multi-AHB bus matrix);

Блок пам'яті (Memory):

- ✓ До 1 МБайт Flash-пам'яті;
- ✓ До 196 кбайт SRAM-пам'яті;
- ✓ Контролер SDIO (карти SD, SDIO, MMC, CE-ATA);
- ✓ FSMC-контролер (Compact Flash, SRAM, PSRAM, NOR, NAND і LCD 8080/6800);

Системний блок (System):

- ✓ Напруга живлення 1,8 ... 3,6 (POR, PDR, PVD і BOR);
- ✓ Внутрішні RC-генератори на 16 МГц і 32 кГц (для RTC);
- ✓ Зовнішнє джерело тактування 4 ... 26 МГц і для RTC – 32,768 кГц;

11.3 Архітектура та характеристики мікроконтролера STM32F407VG

- ✓ Апаратне обчислення CRC, 96-бітний унікальний ID;
- ✓ Зовнішній осцилятор 32kHz +16 MHz;
- ✓ Xtal осцилятори 32 kHz;
- ✓ Регулятор POR/PDR/PVD;
- ✓ 51/82/114/140 I/Os;
- ✓ Clock control;

Блок роботи з аналоговими сигналами:

- ✓ Три 12-бітних АЦП на 24 вхідних каналу (швидкість до 7,2 мегасемплів, температурний датчик);
- ✓ Два 12-бітових ЦАП;

Блок DMA-контролера на 16 потоків з підтримкою пакетної передачі (16-channel DMA);

Блок управління (Control):

- ✓ 17 таймерів (16 і 32 розрядні);

- ✓ Два сторожових таймера (WDG і IWDG);

Блок комунікації (Connectivity):

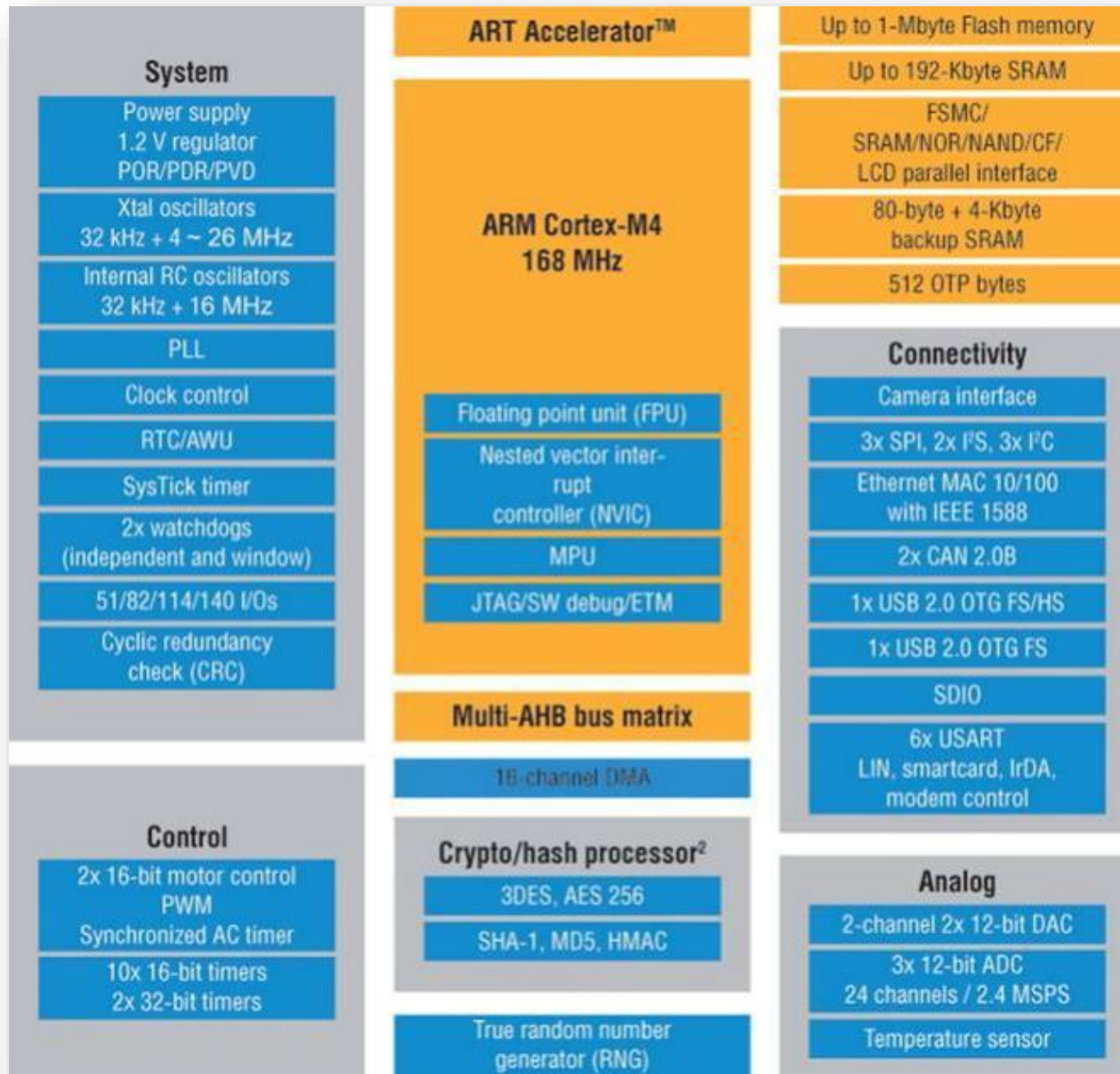
- ✓ CAN (2,0 В Active);
- ✓ USB 2.0 FS / HS OTG;
- ✓ 10/100 Ethernet MAC (IEEE 1588v2, MII /RMII);
- ✓ Інтерфейс цифрової камери (8/10/12/14-бітові режими);
- ✓ Апаратний генератор випадкових чисел;

Блоккриптопроцесора (Crypto/hash processor):

- ✓ Модуль шифрування AES 128, 192, 256, Triple DES, HASH (MD5, SHA-1), HMAC;

Розширений температурний діапазон – 40...105 °C.

11.3 Архітектура та характеристики мікроконтролера STM32F407VG



Структура мікроконтролерів STM32F4xx

11.3 Архітектура та характеристики мікроконтролера STM32F407VG

Коментар:

CPU (central processing unit) – процесор.

DMIPS (от Dhrystone MIPS, Dhrystone Million Instructions per Second) – число мільйонів інструкцій, що виконуються процесором за секунду під час штучного (синтетичного) тесту Dhrystone. Dhrystone видає результат у формі: кількість ітерацій на секунду. Часто цей результат призводять до DMIPS шляхом поділу на 1757 (результат Dhrystone/s для комп'ютера VAX 11/780, тобто номінальної машини з 1 DMIPS). DMIPS можна ділити на частоту процесора, щоб отримати DMIPS/MHz. Такі одиниці дозволяють порівнювати процесори з різною тактовою частотою.

DSP (digital signal processor) – цифровий процесор обробки сигналів.

Прискорювач ART™ Accelerator – це технологія пула кеша, зовнішнього по відношенню до ядра Cortex-M, який може обнулювати дію станів очікування. Прискорювач ART™ Accelerator спроектований таким чином, що він зберігає гарвардську архітектуру мікроконтролерів Cortex-M, надаючи окремі пули кеш-пам'яті I-Bus та D-Bus.

AHB (Advanced High-performance Bus – розширена високопродуктивна шина) – внутрішньосистемний інтерфейс призначений для об'єднання швидкодіючих високопродуктивних модулів, що працюють на високих тактових частотах.

SRAM (static random access memory) – статическая память с произвольным доступом.

SDIO (SecureDigital Input/Output) – технологія, що дозволяє на базі формату SD створювати різні пристрої, наприклад, GPS-приймачі, Wi-Fi та Bluetooth-адаптери. Пристрої, створені за допомогою технології SDIO, підключаються до комп'ютера (смартфону, КПК тощо) через слот картки пам'яті SD.

11.3 Архітектура та характеристики мікроконтролера STM32F407VG

PSRAM (Pseudostatic RAM) – високошвидкісна псевдостатична пам'ять CMOS, призначена для недорогих портативних додатків. Для безперервного режиму роботи з асинхронною шиною пам'яті в PSRAM-компонентах застосовується явний механізм саморегенерації. Прихована регенерація не потребує додаткового обслуговування системним контролером пам'яті і не істотно впливає на продуктивність операцій запису/читання пристрою.

POR (Power On Reset) – внутрішній супервізор живлення, що зберігає стан скидання мікроконтролера до тих пір, поки напруга живлення не досягне фіксованого, налаштованого на підприємстві порогового рівня.

PDR (Power Down Reset) – внутрішній супервізор живлення, що виконує скидання мікроконтролера при знятті живлення.

BOR (Brownout Reset) – внутрішній супервізор живлення, що зберігає стан скидання мікроконтролера до тих пір, поки напруга живлення не досягне заданого, налаштованого користувачем порогового рівня.

PVD (Programmable Voltage Detector) – внутрішній програмований детектор живлення для моніторингу, якщо ця напруга вище або нижче заданих рівнів, то встановлюється біт в регістрі управління живленням.

RTC (Real Time Clock) – внутрішній годинник реального часу мікроконтролера.

DMA (Direct Memory Access) – прямий доступ до пам'яті, що використовується для забезпечення високошвидкісної передачі даних між периферійними пристроями та пам'яттю, а також для копіювання даних з однієї області пам'яті в іншу. Процес пересилання даних за допомогою DMA не вимагає будь-яких дій від процесора і звільняє ресурси для виконання інших операцій.

11.4 Вбудовані інтерфейси комунікації

Блок комунікації (дивіться структуру мікроконтролерів STM32F4xx) складається із наступних інтерфейсів:

Ethernet.

Даний блок присутній не в усіх продуктах сімейства, а лише в контролерах STM32F407 /STM32F417. Блок виконаний за стандартом IEEE802.3. Можлива передача даних зі швидкістю 10/100 Мбіт/с. Доступна синхронізація годин для чого протокол IEEE1588 v2 реалізований апаратно. PHY-трансивер з'єднується безпосередньо з портом MII або RMI.

USB (Universal Serial Bus).

Мікроконтролер має два роздільних блоку USB. Перший – USB OTG full-speed, є повністю апаратною реалізацією і сумісний зі стандартами USB 2.0, а також OTG 1.0. Працює на швидкості до 12 Мбіт/с. Підтримується робота в режимі Host/Device/OTG. Присутній SRP (Session Request Protocol) і HNP (Host Negotiation Protocol). Другий – USB OTG high-speed працює в режимі Host/Device/OTG з високою швидкістю 480 Мбіт/с, для чого необхідний блок передатчика, що працює на високій швидкості через спеціальний ULPI-інтерфейс.

SDIO (Secure Digital Input/Output).

Інтерфейс дозволяє працювати з картами SD/SDIO/MMC-картами пам'яті, дисковими контролерами CE-ATA. У восьмибітному режимі несуча частота обміну даними складає 48 MHz. Контролер відповідає таким стандартам: SD Memory Card 2.0, MultiMediaCard System 4.2 (робота в режимах 1/4/8 біт), SD I/O Card 2.0, CE-ATA 1.1.

11.4 Вбудовані інтерфейси комунікації

SPI (Serial Peripheral Interface).

Пристрій включає три блоки SPI, кожен з яких працює в режимі Master (Multimaster) або в режимі Slave, передаючи дані у напівдуплексному, дуплексному або сімплексному режимах. Підтримується апаратний розрахунок контрольних сум CRC для підвищення надійності каналу зв'язку: так CRC може бути переданий останнім байтом слова в режимі Tx, присутня автоперевірка правильності CRC останнього байта. Блок пристрою SPI1 працює на швидкостях до 37,5 Мбіт/с. Інші обмежені максимальною швидкістю в 21 Мбіт/с.

USART (Universal Synchronous Asynchronous Receiver Transmitter).

У мікроконтролер вбудовано чотири блоки USART і два UART. Блоки USART1 і USART6 допускають високошвидкісний обмін даними на швидкості до 10,5 Мбіт/с. Інші підтримують швидкість не більше 5,25 Мбіт/с. Вбудована підтримка передачі даних згідно стандарту NRZ (Non Return to Zero). Обмін даними здійснюється з використанням 8- або 9-бітних блоків, один або два біти яких виділені як стоп-біти і біти перевірки парності. USART можна конфігурувати на режим SPI, блок USART при цьому виступає в ролі ведучого пристрою SPI. Використовуючи блок USART можна організувати підключення до інтерфейсу LIN, яка знайшла застосування в автомобільній промисловості, або налаштувати на енкодинг/декодинг ІЧ-сигналу IrDA. Можлива робота з модемами по лініях управління RTS і CTS. Підтримується робота зі смарт-картками.

11.4 Вбудовані інтерфейси комунікації

I2C (Inter-Integrated Circuit).

На борту мікроконтролер містить три блоки I2C, що підтримують роботу в режимі Master/Slave (ведучий або ведений), а також у режимі Multimaster (режим в якому на шині присутні кілька Master-пристроїв, які поділяють спільні ресурси Slave, або по черзі змінюють свій стан з Master на Slave і назад). У складі пристрою є модуль діагностики та виправлення пакетних помилок PEC. Використовується 7-бітний і 10-бітний режим адресації. Підтримуються загальноприйняті для протоколу швидкості обміну даними до 100 кГц в простому режимі і 400 кГц в режимі швидкого обміну даними. Модулі можуть бути сконфігуровані на розширені протоколи SMBus 2.0 і PMBus.

I2S (Inter-Integrated Sound).

У мікроконтролері присутні два мультиплексованих блоку I2S з вбудованим SPI. Обидва модулі можуть бути сконфігурованими на роботу в режимі Master або Slave. Дані передаються по 16, 24 або 32 біта дуплексно або сімплексно. Серед підтримуваних протоколів: Phillips I2S, PCM, MSB і LSB з вирівнюванням даних. Інтерфейс I2S був розроблений для обміну звуковими даними в цифровому форматі. Відтепер для тактування присутній окремий PLL, який робить можливим генерацію частот аудіосемплів від 8 до 192 кГц з похибкою не більше 0,01%.

CAN (Controller Area Network).

На борту знаходиться два CAN-модуля, що працюють за стандартами 2.0A і 2.0B, швидкість роботи досягає 1 Мбіт/с. Модулі працюють зі стандартними, а також з розширеними кадрами. Модуль CAN містить три буфера передачі, трьохкаскадний FIFO-стек і 28 банків фільтрів повідомлень (розподілені і масштабуються).

11.4 Вбудовані інтерфейси комунікації

DCMI (Digital Camera Interface).

Присутній в мікроконтролерах STM32F407 і STM32F417. За допомогою даного інтерфейсу можна організувати пряме підключення до камери або CMOS-матриці. Можлива внутрішня і зовнішня синхронізація, робота в безперервному режимі, автообрізка зайвих частин зображення. Серед підтримуваних форматів 8/10/12/14-бітове прогресивне відео, YCbCr 4:2:2 і RGB 565, JPEG.

FSMC (Flexible Static Memory Controller).

Блок використовується для підключення рідкокристалічних дисплеїв, в якості зовнішньої пам'яті безпосередньо. Блок міститься лише в 100-, 144- або 176-пінових корпусах. Можливе сполучення з підключеною синхронної або асинхронної пам'яттю або PCMCIA-пристроями. В основному блок призначений для видачі даних МК у відповідному підключеним пристроям вигляді без зайвих витрат процесорного часу на перекодування даних. Таким чином кожній зовнішній пристрій має власний адрес пулу, власні сигнали для управління. Подавши необхідний сигнал вибору мікросхеми можна отримати доступ до того чи іншого зовнішнього пристрою (одночасне використання не припустимо). Можливе підключення таких типів пам'яті як NAND Flash, Compact Flash, NOR Flash, SRAM і PSRAM. Інтерфейс налаштований для роботи з LCD-контролерами Motorola 6800 і Intel 8080, однак може бути легко використаний для сполучення з контролерами інших виробників.

11.5 Робота з аналоговими сигналами

Блок роботи з аналоговими сигналами (дивіться структуру мікроконтролерів STM32F4xx) містить **три АЦП і два одноканальних ЦАП**. АЦП має хорошу роздільну здатність 12 біт і високу швидкість перетворення – 2,4 мегасемпла в звичайному режимі і 7,2 мегасемпла – в потрібному режимі. Максимально доступне число аналогових каналів – 24. Як і в більшості сучасних МК, присутній генератор опорного напруги. **Гнучка система налаштувань вбудованого аналогового мультиплексора дозволяє задавати будь-які послідовності перетворення аналогових каналів (за винятком одночасного перетворення одного каналу на декількох АЦП).**

Налаштування АЦП дозволяють виробляти одноразові і циклічні вимірювання. Для проведення перетворення на максимальних швидкостях необхідно дотримуватися діапазону напруги живлення 2,4 ... 3,6 В. При зниженні напруги до 1,8 (1,7) В швидкість перетворення знижується до 1,2 мегасемплов. **Для контролю внутрішньої температури мікроконтролера вбудований температурний датчик.** На його виході формується напруга в залежності від навколишньої температури. Вихід датчика через мультиплексор підключається до АЦП. Використовуючи температурний датчик, можна вимірювати температуру від -40 до 125 °С з точністю $\pm 1,5$ °С.

ЦАП має роздільну здатність 12 біт, перетворення можливо в 8/12-бітовому форматі з вирівнюванням цього результату по лівому або правому краях. Так як ЦАП містить два канали, тобто можливість формування стереосигнала. Доступна функція автоматичної генерації шумового сигналу з мінливою амплітудою або трикутного сигналу.

Тема 12. Периферія мікроконтролера STM32F407V. Інтерфейси SPI, USART, SDIO та SAI

12.1 Інтерфейс SPI.

12.2 Інтерфейс UART/USART.

12.3 Інтерфейс SDIO.

12.4 Інтерфейс SAI.

12.1 Інтерфейс SPI

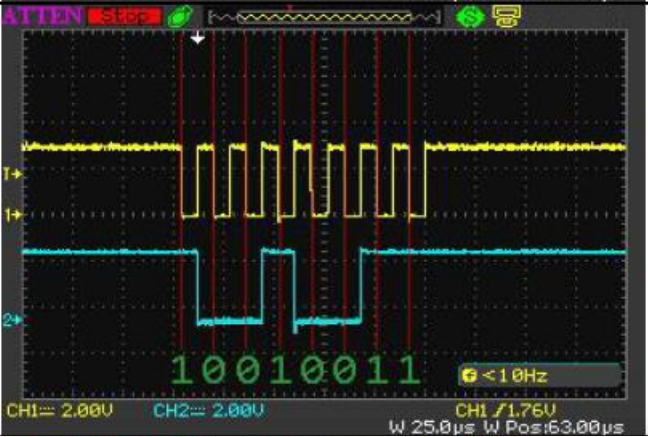
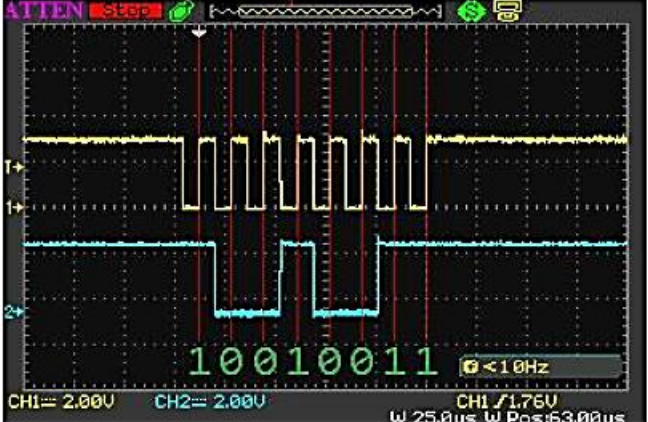
Послідовний периферійний інтерфейс (SPI) – послідовний синхронний стандарт передачі даних в режимі повного дуплексу.

Режими роботи SPI Режим	CPOL	CPHA	Осцилограма	Опис режиму
0	0	0		Вибірка по передьому наростаючому фронті
1	0	1		Вибірка по задньому спадаючому фронті

Режими роботи SPI:

Номер режиму складається з двох біт - CPOL і CPHA. Біт CPOL визначає, в якому стані буде перебувати вивід SCL в той час, коли нічого не передається. Якщо CPOL = 0, то в режимі простою на виводі низький логічний рівень. Це означає, що передній фронтом буде вважатися перехід з 0 в 1. Якщо CPOL = 1, то в режимі простою на виводі високий логічний рівень. Це означає, що передній фронтом буде вважатися перехід з 1 в 0 (а заднім фронтом, відповідно, навпаки з 0 в 1).

12.1 Інтерфейс SPI

2	1	0		Вибірка по передньому спадаючому фронту
3	1	1		Вибірка по задньому наростаючому фронту

Біт CPHA визначає по якому фронту потрібно робити вибірку 0 – по передньому фронту, 1 – по задньому фронту. Важливий параметр – порядок проходження біт. Зазвичай, першим передається старший біт, але іноді буває навпаки, якщо цього не врахувати, то можливі помилки при роботі інтерфейсу. Кількість біт може змінюватися, зазвичай це 8 біт.

Далі розглянемо SPI на платі STM32F4DISCOVERY. Для спрощення роботи зі всією периферією мікроконтролера (таймери, UART, SPI і т.д.) компанія ST розробила бібліотеку **stdperiph_lib**. З її використанням код стає більш зрозумілим і простішим для читання, також поліпшується перенесення коду з одного STM32 контролера на інший.

12.1 Інтерфейс SPI

RXONLY – якщо використовується 2-х провідний режим (див. біт VIDIMODE) то установка цього біта забороняє передачу, SPI модуль працює тільки на прийом.

Особливість виводу NSS. Якщо SPI модуль налаштований в режимі Slave, то він може отримувати сигнал з вивода NSS або ж програмно. Якщо біт SSM скинутий, то сигнал SS буде зчитуватися з вивода NSS, а якщо він встановлений, то стан вивода NSS ігнорується. У такому випадку для керування сигналом SS переходить на біт SSI. Біт встановлений - є сигнал SS, інакше немає. Якщо ж SPI модуль працює в режимі майстра, то вивід NSS потрібно підтягнути до живлення або включити програмне управління (SSM = 1) і встановити біт SSI. В іншому випадку - SPI модуль вирішить, що з'явився новий Master і сам стане Slave.

LSBFIRST – задає порядок передачі біт: 0 - спочатку передається старший біт. 1 - спочатку передається молодший біт.

SPE – вмикає/вимикає SPI модуль.

BR2, BR1, BR0 – задають швидкість прийому / передачі (частоту SCK). Частота тактирування модуля SPI ділиться на число, яке задається комбінацією цих трьох біт. Змінювати стан цих біт можна тільки коли SPI модуль вимкнений (біт SPE = 0).

MSTR – якщо біт встановлений – SPI модуль є Master, інакше Slave.

CPOL – полярність сигналу SCK. **CPHA** – фаза сигналу SCK.

Регістр SPI_CR2 – регістр керування 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	FRF	Res.	SSOE	TXDMAEN	RXDMAEN
								rw	rw	rw	rw		rw	rw	rw

TXEIE – дозволяє переривання, коли буфер передачі порожній.

RXNEIE – дозволяє переривання, коли буфер заповнений даними.

ERRIE – дозволяє переривання у разі виникнення помилки. Їх три, щоб розібратися яка виникла, потрібно дивитися стан біт в регістрі статусу.

SSOE – якщо цей біт виставлений, то SPI модуль сам керує виводом NSS. Тобто перед початком передачі виставляє нуль на цьому виводі, а після завершення – виставляє одиницю.

12.1 Інтерфейс SPI

TXDMAEN – дозволяє / забороняє запит DMA по завершенню передачі.

RXDMAEN – дозволяє / забороняє запит DMA по завершенню прийому.

Статусний регістр SPI_SR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSID E	TXE	RXNE
							r	r	r	r	rc_w0	r	r	r	r

FRE – Frame error flag, він використовується, коли SPI модуль працює в режимі «TI mode» (біт FRF = 1).

BSY – якщо цей біт встановлений, то модуль SPI зараз зайнятий передачею даних.

OVR – біт виставляється в тому випадку, якщо в SPI модуль надійшли нові дані і замінили старі, що не були прочитані.

MODF – виставляється в тому випадку якщо Master раптово перестав бути Master. Таке можливо, коли вивід Master NSS налаштована як вхід і на неї надійшов сигнал низького рівня.

CRCERR – помилка контрольної суми. **UDR** – флаг не використовується в режимі SPI.

TXE – передача даних закінчилась. **RXNE** – прийом даних завершено.

Регістр SPI_DR – регістр даних.

Представляє собою 16-ти бітний регістр даних, який розбитий на два. Один для передачі, а інший для прийому, але робота з ними здійснюється через один регістр SPI_DR. Якщо щось в нього записати, то запис даних проводиться в регістр для передачі. Якщо прочитати, то дані зчитуються з регістра для прийому даних.

Регістр SPI_CRCPR.

У цей регістр записують деяке число, яке повинно вплинути на розрахунок контрольної суми. За замовчуванням записано число 7.

12.2 Інтерфейс UART/USART

Регістр SPI_TXCRCR.

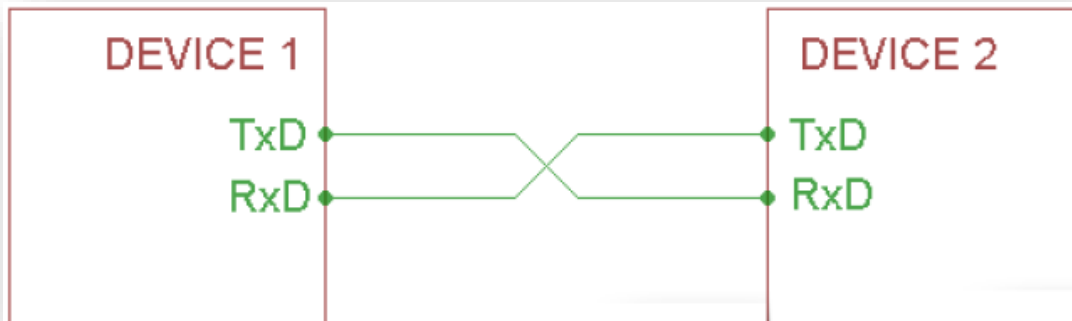
У цей регістр записується контрольна сума, яка була розрахована для переданих даних.

Регістр SPI_RXCRCR.

У цей регістр записується контрольна сума, яка була розрахована для прийнятих даних.

UART – вузол обчислювальних пристроїв, призначений для організації зв'язку з іншими цифровими пристроями. Перетворює передані дані в послідовний вид так, щоб було можливо передати їх по цифровій лінії інших аналогічних пристроїв. Метод перетворення добре стандартизований і широко застосовувався в комп'ютерній техніці, оскільки часто виникає потреба під'єднати пристрій до комп'ютера для обміну даними з ним.

Для обміну даними два пристрої повинні бути з'єднані за інтерфейсом UART:



Передавальний вивід одного пристрою з'єднується з приймаючим виводом іншого, і навпаки. Обмін даними можливий одночасно в обидві сторони. Коли передача не здійснюється, на виході передавача завжди присутня логічна одиниця.

12.2 Інтерфейс UART/USART

Регістр SPI_TXCRCR.

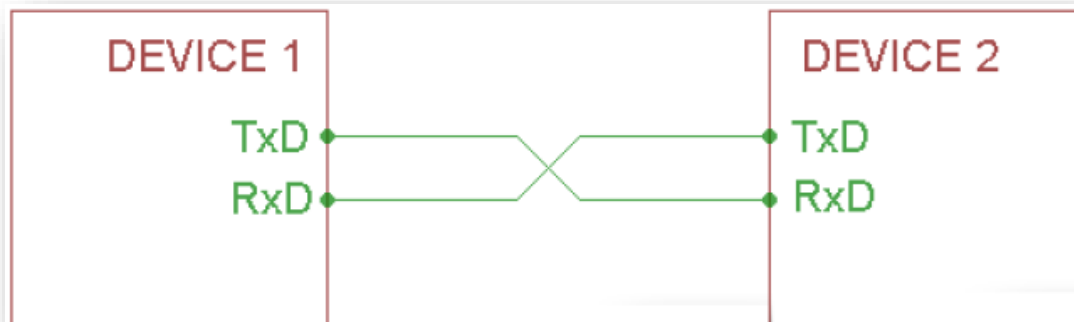
У цей регістр записується контрольна сума, яка була розрахована для переданих даних.

Регістр SPI_RXCRCR.

У цей регістр записується контрольна сума, яка була розрахована для прийнятих даних.

UART – вузол обчислювальних пристроїв, призначений для організації зв'язку з іншими цифровими пристроями. Перетворює передані дані в послідовний вид так, щоб було можливо передати їх по цифровій лінії інших аналогічних пристроїв. Метод перетворення добре стандартизований і широко застосовувався в комп'ютерній техніці, оскільки часто виникає потреба під'єднати пристрій до комп'ютера для обміну даними з ним.

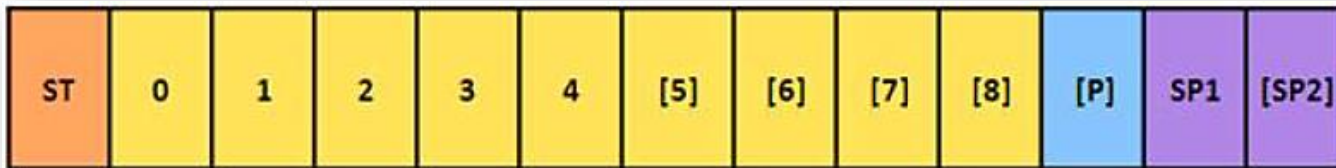
Для обміну даними два пристрої повинні бути з'єднані за інтерфейсом UART:



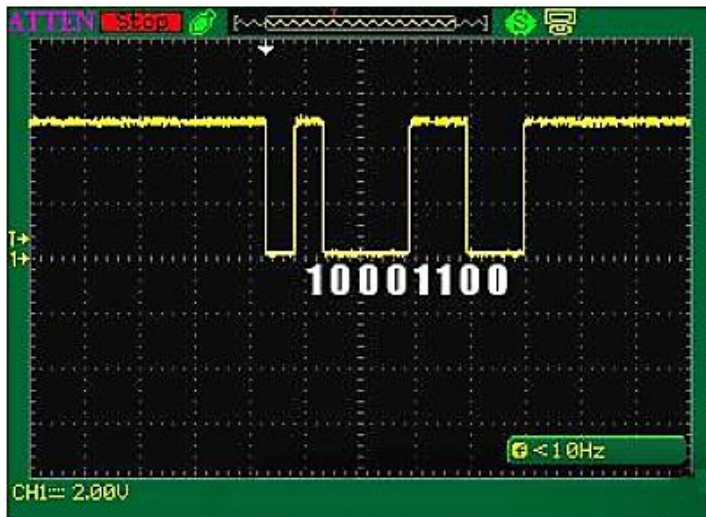
Передавальний вивід одного пристрою з'єднується з приймаючим виводом іншого, і навпаки. Обмін даними можливий одночасно в обидві сторони. Коли передача не здійснюється, на виході передавача завжди присутня логічна одиниця.

12.2 Інтерфейс UART/USART

Перед початком передачі даних передавач встановлює на виході логічний нуль. Це називається стартовим бітом, після якого починається передача біт даних, яких зазвичай вісім, але може бути від 5 до 9. Потім слідує біт перевірки парності, якщо вона використовується. Цей біт призначений для запобігання обробки некоректних даних після прийому. Потім, після відправлення всіх біт йде стоповий біт, зазвичай один, можливі два. Стоповий біт завжди логічна одиниця.



Формат передачі: ST – стартовий біт, 0...8 – біти даних, P – біт парності, SP1 і SP2 – стопові біти



Осцилограма передачі байту даних

Для успішної передачі даних, на обох пристроях, UART повинен бути налаштований з однаковими параметрами. Ще до початку передачі потрібно задати: швидкість, кількість стопових біт, кількість біт даних, наявність перевірки парності. Швидкість передачі даних довільна, але як правило існує певний набір стандартних швидкостей: 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600 біт/сек і т.д. Чим нижче швидкість передачі даних – тим надійніше така передача.

12.2 Інтерфейс UART/USART

У ньому використовуються молодші 8 або 9 біт (залежно від біта M в регістрі USART_CR1). Щоб відправити у UART дані записуємо їх в цей регістр. Щоб прочитати дані читаємо цей регістр. При цьому, щоб не виникало ситуацій, зчитування при передачі, або запису при прийомі використовується регістр USART_SR-статусний регістр.

Регістр USART_SR – статусний регістр.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
Reserved						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

RXNE – цей біт встановлюється, коли в UART щось прийшло. Якщо не використати з USART_DR дані, то вони перезапишуться новими.

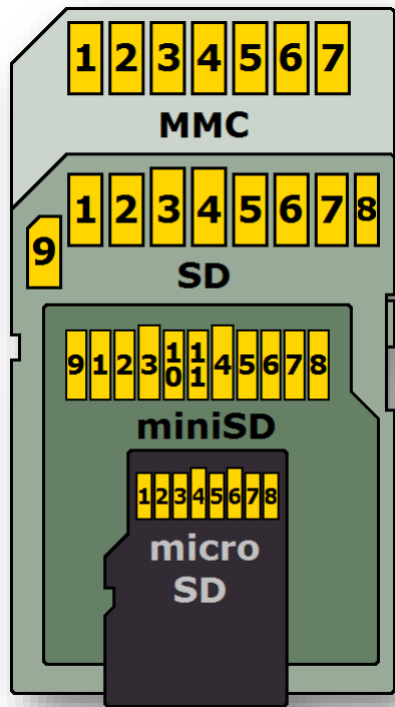
TC – якщо цей біт встановлений в одиницю, то це означає що передача даних завершена і можна записувати в регістр (USART_DR).

12.3 Інтерфейс SDIO

SDIO – це інтерфейс для передачі даних в/з карт пам'яті.

Працювати з картами флеш пам'яті можна за допомогою SPI інтерфейсу, але сучасні 32-бітні контролери, які мають модуль, спеціально призначений для роботи з картами пам'яті - SDIO, є значно дешевшими. Це також істотно спрощує і прискорює роботу.

Контакти і порівняння розмірів SD карт



Призначення виводів карт пам'яті

MMC Pin	SD Pin	miniSD Pin	microSD Pin	Ім'я	I/O	Logic	Опис
1	1	1	2	NC	.	.	Не використовується
2	2	2	3	CMD	I/O	PP, OD	Command, Response
3	3	3		VSS	S	S	Ground
4	4	4	4	VDD	S	S	Power
5	5	5	5	CLK	I	PP	Serial Clock
6	6	6	6	VSS	S	S	Ground
7	7	7	7	DAT0	I/O	PP	SD Serial Data 0
	8	8	8	NC	.	.	Не використовується (memory cards)
	9	9	8	nIRQ	O	OD	Interrupt (SDIO cards) (Negative Logic)
		10	1	NC	.	.	Не використовується
		11		NC	.	.	Зарезервовано
				NC	.	.	Зарезервовано

12.4 Інтерфейс SAI

З картою можна працювати по SDIO в режимі однієї та 4-х бітної шини даних. Жодної принципової різниці, крім кількості використовуваних провідників, це не має.

Як видно контактів, на флешці досить багато. З них, крім живлення та землі, в основному потрібні три:

- ✓ CLK - тактування карти.
- ✓ CMD - по цій лінії передаються команди.
- ✓ DAT0 - лінія даних (у випадку 4-х бітної їх буде 4).

З'єднання з мікроконтролером:

- ✓ PC8 --- SDIO_D0 (DAT0)
- ✓ PC12 --- SDIO_CK (CLK)
- ✓ PD2 --- SDIO_CMD (CMD).

Передача даних з/на карту пов'язана з обчисленням контрольних сум. Це відбувається при кожній передачі.

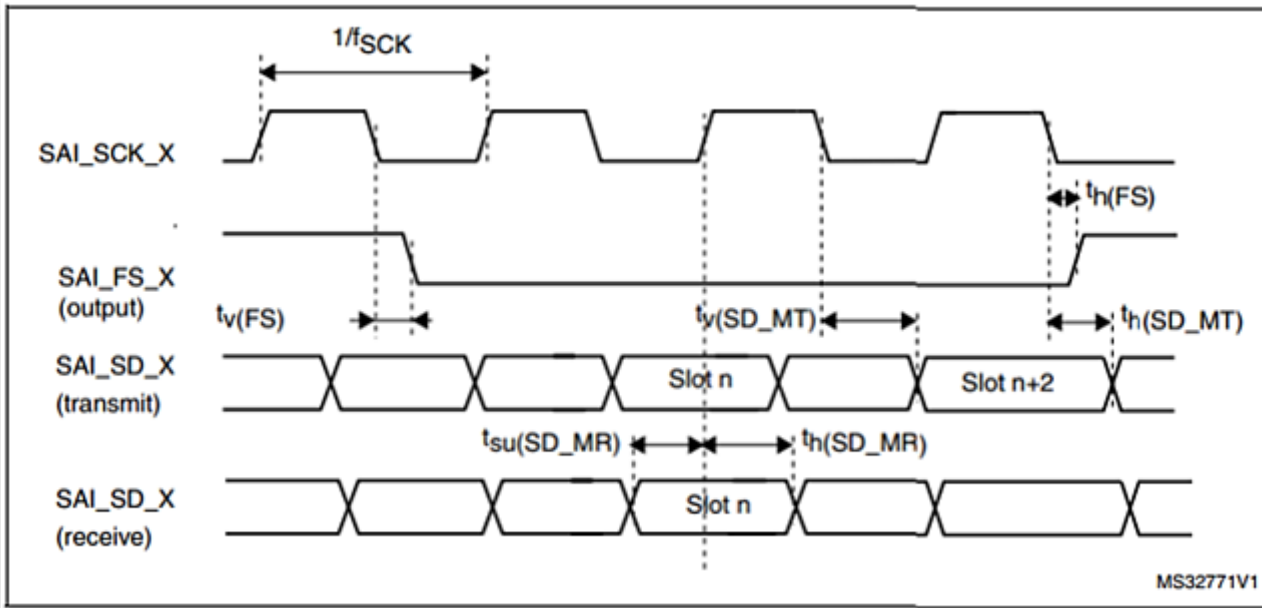
SAI (Serial audio interface) інтерфейс – інтерфейс передачі звукових сигналів. Складається з двох звукових підблоків, незалежних один від одного. Підтримує I2S, PCM/DSP і AC`97 протоколи. Може працювати в режимах Master/Slave. Звукові підблоки можуть як приймати, так і передавати, синхронно чи ні. Може використовуватися з DMA контролером, який може отримувати доступ до системної шини незалежно від центрального процесора.

12.4 Інтерфейс SAI

Характеристики послідовного звукового інтерфейсу

Symbol	Parameter	Conditions	Min	Max	Unit
f_{MCKL}	SAI Main clock output	-	256 x 8K	$256 \times F_s^{(2)}$	MHz
F_{SCK}	SAI clock frequency	Master data: 32 bits	-	$64 \times F_s$	MHz
		Slave data: 32 bits	-	$64 \times F_s$	
D_{SCK}	SAI clock frequency duty cycle	Slave receiver	30	70	%
$t_{V(FS)}$	FS valid time	Master mode	8	22	ns
$t_{su(FS)}$	FS setup time	Slave mode	2	-	
$t_{h(FS)}$	FS hold time	Master mode	8	-	
		Slave mode	0	-	
$t_{su(SD_MR)}$	Data input setup time	Master receiver	5	-	
$t_{su(SD_SR)}$		Slave receiver	3	-	
$t_{h(SD_MR)}$	Data input hold time	Master receiver	0	-	
$t_{h(SD_SR)}$		Slave receiver	0	-	
$t_{V(SD_ST)}$	Data output valid time	Slave transmitter (after enable edge)	-	22	
$t_{h(SD_ST)}$		Master transmitter (after enable edge)	-	20	
$t_{V(SD_MT)}$					
$t_{h(SD_MT)}$	Data output hold time	Master transmitter (after enable edge)	8	-	

12.4 Інтерфейс SAI



Часові діаграми роботи SAI Master

Часові діаграми роботи SAI Slave

