

Лекція 11

Інтерфейс I²C

1 Загальна характеристика інтерфейсу I²C

Інтерфейс I²C (InterIC, або ІІС) є двопровідним послідовним синхронним інтерфейсом, який розроблено фірмою “Philips Corporation”, і призначено для зв'язку між інтегральними мікросхемами або модулями. Існує ціла група I²C–сумісних пристроїв для різних додатків: цифро–аналогові та аналого–цифрові перетворювачі, мікросхеми пам'яті, мікроконтролери, що містять модуль I²C, і т. ін.

Шина інтерфейсу I²C складається із двох ліній:

- двонаправленої лінії даних (SDA);
- лінії тактових (синхро) імпульсів (SCL).

На рисунку 1 наведена структурна схема типової мережі, що використовує для обміну даними інтерфейс (шину) I²C.

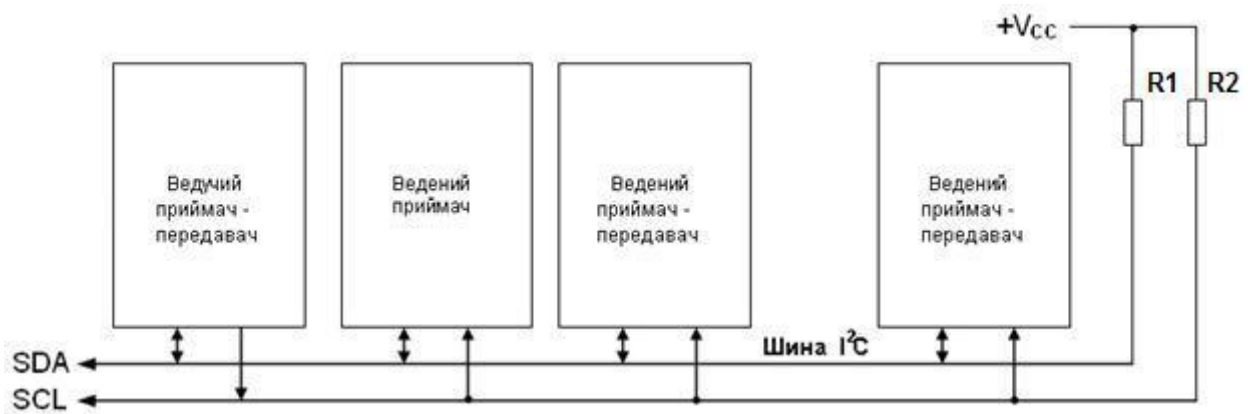


Рисунок 1 – Структурна схема мережі з інтерфейсом I²C

Лінії шини SDA і SCL з'єднані з додатним полюсом напруги живлення (+Vcc) через підтягуючі резистори R1 та R2.

Передавач генерує і передає повідомлення, а приймач його приймає.

Один із двох пристроїв, які беруть участь у обміні, є ведучим (master), а інший – веденим (slave). Ведучий пристрій керує роботою шини та формує тактові сигнали (синхросигнали) SCL.

Кожен пристрій, який використовує для обміну інтерфейс I²C, має свою адресу. Коли ведучий пристрій бажає ініціювати обмін даними, він передає на лінію SDA адресу пристрою, з яким буде виконуватися обмін (передача/прийм).

Всі ведені пристрої слідкують за адресою, яка виставляється на шину, і порівнюють її із власною адресою. Після адреси ведучий передає біт напрямку R/\overline{W} , який визначає, чи буде ведучий читати дані від веденого ($R/\overline{W} = 1$), чи буде передавати дані веденому ($R/\overline{W} = 0$). Ведений приймач після одержання адреси або даних видає на шину SDA біт підтвердження (логічний нуль). Інтерфейс I²C може використовувати два формати адреси:

- 7-ми бітну адресу;
- 10-ти бітну адресу.

1 багатьох мікроконтролерах сімейства ARM використовується 7-ми бітна адреса.

На рисунку 2 наведений формат 7-ми бітної адреси.

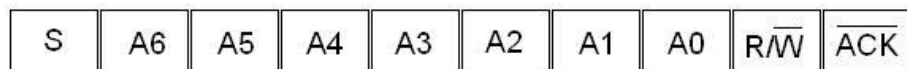


Рисунок 2 – Формат 7-бітної адреси

(S – старт, R/\overline{W} – біт «читання/запис», \overline{ACK} – підтвердження)

На рисунку 3 наведені часові діаграми, які відображають стани на шині (рисунку 3, а), а також пояснюють формування сигналу підтвердження (рисунку 3, б).

Наведені діаграми відображають наступні коректні стани сигналів на шині під час обміну даними:

- шина не зайнята: на обох лініях одиниці ($SDA = SCL = 1$);
- початок обміну даними: зміна сигналу на лінії даних SDA з одиниці в нуль при одиничному значенні сигналу на лінії SCL, що визначає умову початку обміну (умова «СТАРТ» – S);
- припинення передачі: зміна сигналу на лінії даних з нуля в одиницю при одиничному значенні сигналу на лінії SCL, що визначає умову закінчення обміну (умова «СТОП» – P);

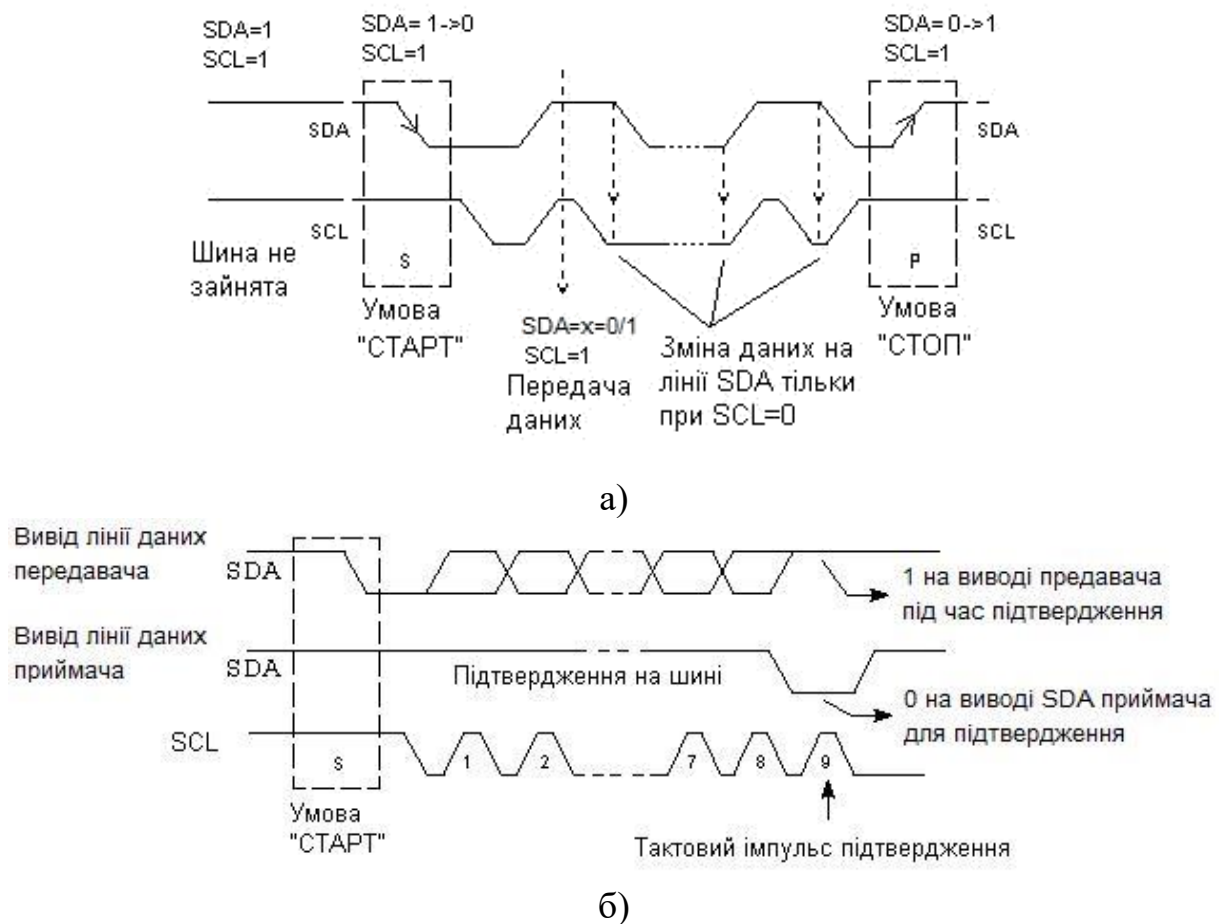


Рисунок 3 – Стани на шині: старт – S, стоп – P, а) – передача даних; б) – формування сигналу підтвердження

- коректність даних: при одиничному значенні сигналу на лінії SCL стан лінії даних не повинен змінюватися, щоб не сформувати невірну умову СТАРТ або СТОП. Дані можна змінювати, якщо на лінії SCL присутній низький рівень сигналу (логічний нуль). На один біт інформації на лінії SDA міститься один тактовий імпульс на лінії SCL. Кожний цикл обміну даними починається умовою «СТАРТ» і закінчується умовою «СТОП». Кількість інформаційних бітів даних, що передаються між цими станами, необмежена. Дані передаються байтами. Приймач підтверджує одержання чергового байта, посылаючи біт підтвердження (логічний нуль) після прийому кожного байта;
- біт підтвердження: передається після прийому кожного байта даних або адреси. Активний передавач (ведений або ведучий) після передачі чергового байта формує на лінії SDA сигнал високого рівня. Ведучий

пристрій (приймач або передавач) формує на лінії SCL тактовий імпульс, а приймач (ведучий або ведений) видає на лінію даних SDA сигнал підтвердження низького рівня. Приймач, що генерує біт підтвердження, підключає лінію SDA до низького рівня і утримує її в цьому стані доти, доки тактовий імпульс лінії SCL не переключиться у стан низького рівня. Для припинення обміну приймач повинен залишити останній прийнятий байт без підтвердження, що автоматично викликає формування активним передавачем умови «СТОП».

Можливі чотири режими (типу) обміну даними для інтерфейсу I²C.

1. Ведучий передавач: на вихід SDA передавача виводяться дані, а на лінію SCL видаються синхроімпульси. Перший переданий байт містить адресу веденого приймача (7 біт) і біт напрямку обміну даними $R/\overline{W} = 0$, що говорить про те, що буде проводитися запис (передача). Дані передаються послідовно по 8 біт. Після передачі чергового байта (адреси або даних), ведучий передавач очікує від веденого приймача біт підтвердження \overline{ACK} . Для завдання початку і кінця сеансу обміну даними ведучий передавач формує умови «СТАРТ» і «СТОП».

2. Ведучий приймач: спочатку сеансу обміну ведучий приймач передає на лінію SDA адресу веденого передавача (7 біт) і біт напрямку обміну $R/\overline{W} = 1$, що говорить про те, що ведучий буде здійснювати прийом. Ведучий приймач формує імпульси синхронізації, які передаються лінією SCL. Після прийому адреси ведений передавач виставляє на лінію SDA сигнал підтвердження \overline{ACK} , а потім передає дані. Дані від веденого передавача передаються послідовно по 8 біт лінією SDA. Після прийому чергового байта ведучий приймач виставляє на лінію SDA сигнал підтвердження \overline{ACK} . Умови СТАРТ і СТОП формуються ведучим пристроєм для вказівки початку і кінця сеансу обміну послідовними даними.

3. Ведений приймач: ведучий передавач видає на лінію SDA адресу веденого приймача й біт напрямку $R/\overline{W} = 0$, що говорить про те, що буде виконуватися запис (передача). На лінії SCL ведучий передавач видає синхроімпульси. Після одержання адреси ведений приймач передає сигнал

підтвердження \overline{ACK} , після чого ведучий передавач послідовно передає на лінію SDA дані. Ведений приймач після прийому чергового байта даних передає сигнал підтвердження \overline{ACK} , що надходить до ведучого передавача лінією SDA. Умови СТАРТ і СТОП формуються ведучим передавачем.

4. Ведений передавач: перший байт на шині SDA (адреса) приймається і обробляється веденим передавачем так само, як і в режимі веденого приймача. При цьому біт напрямку $R/\overline{W} = 1$, що говорить про те, що ведучий буде здійснювати прийом. Дані послідовно передаються лінією SDA від веденого передавача у той час, як синхроімпульси передаються лінією SCL від ведучого приймача. Після передачі кожного байта ведений передавач аналізує наявність на лінії SDA біта підтвердження \overline{ACK} , який передає ведучий приймач. Умови СТАРТ і СТОП формує ведучий приймач.

У підпорядкованому (веденому) режимі апаратні засоби інтерфейсу I²C здійснюють пошук своєї власної підпорядкованої адреси або адреси загального виклику. Якщо отримується одна із цих адрес, відбувається запит на переривання і виконуються відповідні дії. Якщо модуль I²C хоче захопити шину і стати ведучим, то він чекає, поки шина звільниться ($SDA = SCL = 1$). Можливе функціонування в якості веденого при цьому не переривається.

Два і більше пристроїв в I²C можуть спробувати стати ведучими і одночасно згенерувати умову «СТАРТ». У цьому випадку здійснюється арбітраж шини в моменти, коли шина SCL перебуває у високому стані. Якщо один ведучий передає на лінію даних низький рівень, а інший – високий, то останній відключається від лінії, тому що стан шини SDA (низький) не відповідає високому стану внутрішньої шини даних пристрою, який бажає стати ведучим.

Якщо арбітраж шини загублений у головному (ведучому) режимі, то відповідний пристрій I²C переключається у підпорядкований режим і може розпізнавати свою власну підпорядковану адресу.

На рисунку 4 наведені часові діаграми, які відображають обмін даними шиною I²C.

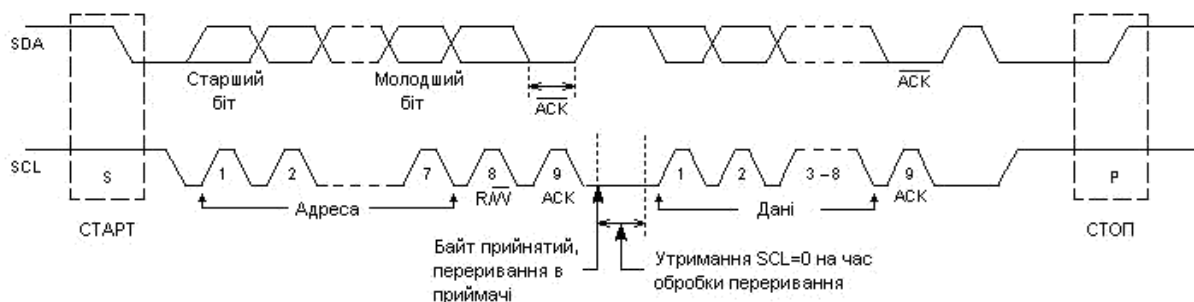


Рисунок 4 – Приклад обміну даними шиною I²C

2.2 Типові особливості інтерфейсу I²C мікроконтролерів сімейства

LPC2300

Інтерфейс I²C мікроконтролерів сімейства LPC2300 має такі особливості:

- шинний інтерфейс стандарту I²C з можливістю конфігурування як ведучого (Master), веденого (Slave) або ведучого/веденого;
- програмоване тактування, що дозволяє керувати швидкістю обміну;
- двонаправлена передача даних між ведучим і веденим;
- в режимі Multi-Master можливе одночасне підключення до шини I²C декількох ведучих пристроїв, при цьому інтерфейс I²C автоматично здійснює арбітраж з метою передачі послідовних даних без спотворень;
- шина I²C може використовуватися для випробувальних і діагностичних цілей;
- основним функціональним призначенням інтерфейсу I²C є зв'язок із зовнішніми периферійними пристроями з послідовним доступом, наприклад:
 - пристроями пам'яті з послідовним доступом;
 - рідкокристалічними індикаторами (РКІ) з вбудованими контролерами;
 - таймером–календарем реального часу;
 - програмованими логічними матрицями (ПЛМ) і т. ін.

Для ініціалізації інтерфейсу I²C в програмі повинні бути присутніми наступні рядки:

```
VICVectCntl1 = 0x00000029; // Обираємо слот для даного
// переривання
VICVectAddr1 = (unsigned)I2CISR; // Передаємо адресу обробника в
```

```

// модуль VIC
VICIntEnable = 0x00000200; // Дозволяємо переривання
PINSEL0      = 0x50;      // Підключаємо виводи
// мікроконтролера до
// модуля
// I2C
I2SCLH       = 0x08;      // Встановлюємо частоту
// тактового сигналу
// 57.6 кГц
I2SCLL       = 0x08;

```

Щоб модуль I²C реагував на всі події, що виникають на шині, він повинен бути відповідним чином сконфігурований. При цьому ми отримуємо периферійний пристрій, керований, в основному, за перериваннями. Отже, перше, що ми повинні зробити, це сконфігурувати векторний контролер переривань VIC (vectored interrupt controller) для роботи за перериванням від модуля I²C. Потім необхідно сконфігурувати блок керування виводами, щоб підключити лінії даних і тактового сигналу модуля I²C до виводів мікроконтролера. На завершення ми повинні задати швидкість передачі, програмуючи регістри I2SCLH і I2SCLL. В обох регістрах для зберігання використовуються тільки молодші 16 біт. Частота передачі бітів визначається за формулою:

$$\text{Bit Freq} = \text{Pclk} / (2 * (\text{I2SCLH} + \text{I2SCLL})), \text{ де}$$

Pclk — частота периферійного тактового сигналу.

У наведеному вище прикладі ініціалізації інтерфейсу схема ФАПЧ відключена, а частота резонатора дорівнює 14,7456 МГц. Тому частота передачі дорівнює:

$$\text{Bit Freq} = (14745600/8) / (2 * (8+8)) = 57600 \text{ біт/сек.}$$

Після конфігурування мікроконтролер може здійснювати обмін даними з іншими пристроями, зчитуючи і видаючи на шину дані в якості ведучого або приймаючи дані та відповідаючи на запити іншого ведучого пристрою.

2.3 Режими роботи інтерфейсу I²C мікроконтролерів сімейства L3C2300

2.3.1 Режим ведучого передавача

У режимі ведучого передавача дані передаються від ведучого до веденого. Для ініціалізації цього режиму, в регістр керування I²C (I2CONSET) необхідно заздалегідь записати значення, які наведено в таблиці 1.

Таблиця 1 – Ініціалізація регістра I2CONSET для режиму ведучого

Номер біта	Назва біта	Значення біта
0	–	–
1	–	–
2	AA	0
3	SI	0
4	STO	0
5	STA	0
6	I2EN	1
7	–	–

Біт (прапорець) I2EN регістра I2CONSET повинен бути встановлений, щоб дозволити роботу інтерфейсу I²C. Біт (прапорець) AA, навпаки, слід скинути, щоб інтерфейс I²C мікроконтролера не видавав в лінію біт «підтвердження» (ACK) у разі, коли ведучим на шині є інший пристрій. Таким чином, функціонування в режимі веденого блокується, і інтерфейс I²C мікроконтролера буде функціонувати в режимі ведучого. Біти (прапорці) STA, STO і SI повинні бути скинуті. Біт SI скидається шляхом запису логічної 1 в біт SIC регістра I2CONCLR.

Перший байт (його передає ведучий), містить адресу веденого приймача (7 бітів) і біт напрямку подальшої передачі даних (R/\overline{W}). У режимі ведучого передавача цей біт повинен бути скинутий (містить значення логічного 0), що відповідає напрямку «запис». Таким чином, перший байт, що передається, містить адресу веденого і біт запису (W). Потім ведучий побайтно передає дані. Після передачі кожного з байтів даних, ведучий приймає від веденого біт

«підтвердження» (ACK). Умови START і STOP генеруються ведучим, щоб вказати початок і кінець послідовної передачі.

Коли вбудований програмований драйвер інтерфейсу I²C встановлює біт STA, відбувається перехід в режим ведучого передавача. Тепер, якщо шина вільна (тобто інші підключені до шини пристрої не встановлюють на ній напругу НИЗЬКОГО рівня), логічна схема I²C генерує і передає в лінію умову START. Після цього, мікроконтролер встановлює біт SI регістра I2CONSET і в регістрі стану інтерфейсу I²C (I2STAT) повертає код стану (результат операції), який дорівнює 08h. Цей код стану призначений для використання вектором і підпрограмою обробки переривання, яка спочатку повинна завантажити адресу веденого і біт запису в регістр даних інтерфейсу I²C (I2DAT), а потім скинути біт SI. Як уже згадувалося вище, біт SI скидається шляхом запису логічної 1 в біт SIC регістра I2CONCLR.

Біт SI встановлюється знову після того, як адреса веденого і біт R/\overline{W} передані веденому, і від нього отримано біт «підтвердження» (ACK). У регістрі стану I2STAT знову повертається відповідний код стану. Дії, які мають бути виконані програмою користувача у відповідності зі значенням цього коду, перераховані в таблиці 2.

Після переведення модуля I²C в режим ведучого, ми можемо почати передачу записуваних даних таким чином:

```
void I2CTransferByte(unsigned Addr, unsigned Data)
{
    I2ADB      = Addr           // Адреса та дані повинні
                               // зберігатися в глобальних
                               // змінних, щоб їх можна було
                               // використовувати у перериван-
                               ні
    I2DAT      = Data;
    I2CONCLR   = 0x000000FF;   // Скидаємо усі біти керування
                               // шиною I2C
    I2CONSET   = 0x00000040;   // Дозволяємо інтерфейс I2C
    I2CONSET   = 0x00000020;   // Формуємо стан СТАРТ
}
```

Таблиця 2 – Коди стану для режиму I²C ведучого передавача

Код стану(I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
08h	Умова START була передана в лінію	Завантажити адресу веденого + W	X	0	0	X	Адреса веденого + W буде передана в лінію; біт ACK буде прийнятий з лінії
10h	Умова повторний START була передана в лінію	Завантажити адресу веденого + W	X	0	0	X	Як і в попередньому випадку, адреса веденого + W буде передана в лінію
18h	Адреса веденого+ W була передана в лінію; біт ACK(підтвердження веденого) був прийнятий з лінії	Завантажити байт даних	0	0	0	X	Байт даних буде передано; біт ACK буде прийнято з лінії
		Немає операцій з I2DAT	1	0	0	X	Буде передана умова повторний START
		Немає операцій з I2DAT	0	1	0	X	Буде передана умова STOP; прапорець STO буде скинуто
		Немає операцій з I2DAT	1	1	0	X	Буде передана умова STOP, після якої буде передана умова START; прапорець STO буде скинуто

Продовження таблиці 2

Код стану(I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення					Наступна дія апаратної логіки інтерфейсу
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI	AA	
20h	Адреса веденого + W була передана; біт NOT ACK було прийнято	Завантажити байт даних	0	0	0	X	Байт даних буде передано; біт ACK буде прийнято з лінії
		Немає операцій з I2DAT	1	0	0	X	Буде передана умова повторний START
		Немає операцій з I2DAT	0	1	0	X	Буде передана умова STOP; прапорець STO буде скинуто
		Немає операцій з I2DAT	1	1	0	X	Буде передана умова STOP, після якої буде передана умова START; прапорець STO буде скинуто
28h	Байт даних з I2CDAT передано; біт ACK було прийнято	Завантажити байт даних	0	0	0	X	Байт даних буде передано; біт ACK буде прийнято з лінії
		Немає операцій з I2DAT	1	0	0	X	Буде передана умова повторний START
		Немає операцій з I2DAT	0	1	0	X	Буде передана умова STOP; прапорець STO буде скинуто
		Немає операцій з I2DAT	1	1	0	X	Буде передана умова STOP, після якої буде передана умова START; прапорець STO буде скинуто

Продовження таблиці 2

Код стану(I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення					Наступна дія апаратної логіки інтерфейсу
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI	AA	
30h	Байт даних з I2CDAT передано; біт NOT ACK було прийнято	Завантажити байт даних	0	0	0	X	Байт даних буде передано; біт ACK буде прийнято з лінії
		Немає операцій з I2DAT	1	0	0	X	Буде передана умова повторний START
		Немає операцій з I2DAT	0	1	0	X	Буде передана умова STOP; прапорець STO буде скинуто
		Немає операцій з I2DAT	1	1	0	X	Буде передана умова STOP, після якої буде передана умова START; прапорець STO буде скинуто
38h	Арбітраж втрачений при передачі адреси веденого + R/W або байта даних	Немає операцій з I2DAT	0	0	0	X	Шина I2C буде «відпущена» без звернення до адресованого веденого
		Немає операцій з I2DAT	1	0	0	X	Умова START буде передана в лінію

Адреса веденого пристрою і дані, які посилаються, зберігаються в глобальних змінних, щоб їх можна було використовувати в підпрограмі обробки переривання. У бітах 7...1 байта адреси знаходиться власне 7-бітна адреса веденого пристрою, а молодший біт вказує тип транзакції (1 – запис, 0 – читання). Формат обміну по шині I²C для режиму ведучого передавача проілюстровано на рисунку 5. Після того як на шині буде сформовано стан

СТАРТ, генерується переривання і з регістра стану модуля I²C можна зчитати код результату операції.

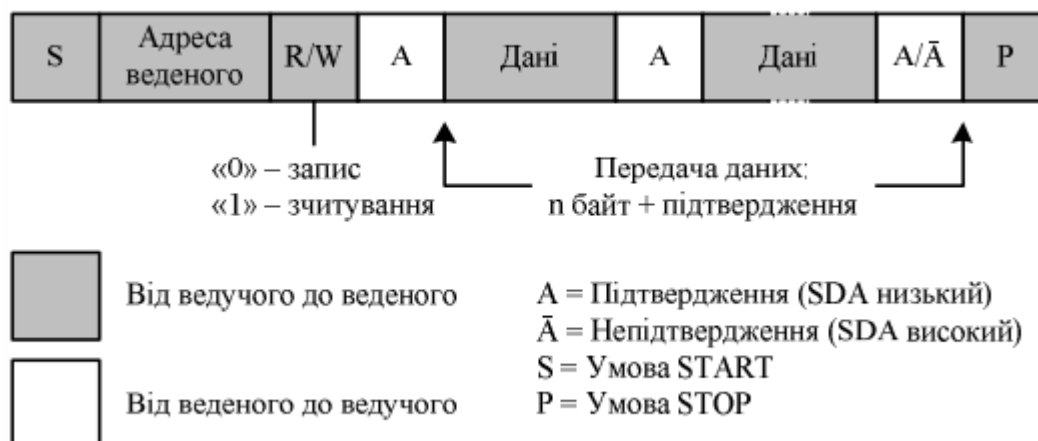


Рисунок 5– Формат обміну шиною I²C для режиму ведучого передавача

При успішному формуванні стану СТАРТ цей код буде дорівнювати 0x08. Зчитавши такий код, прикладна програма повинна завантажити в регістр I2CDATA адресу веденого пристрою з керуючим бітом $R/\bar{W} = 0$. Ці дані будуть видані на шину і підтвержені веденим. При отриманні підтвердження генерується нове переривання, а в регістрі стану буде знаходитися (при успішній передачі) число 0x18. Це означає, що ведений пристрій адресований і готовий до прийому даних. Тепер ми можемо записувати в регістр I2CDATA дані. При записі чергового байта в регістр, він передається по шині I²C і від веденого пристрою приймається підтвердження. Після прийому сигналу від веденого генерується переривання і в регістрі стану формується число 0x28 (при успішній передачі) або 0x20 (при неуспішній). В останньому випадку байт даних необхідно передати повторно. Таким чином, після передачі кожного байта і генерації переривання можна перевіряти код стану і передавати наступний байт. Після передачі всіх даних можна сформувати на шині стан СТОП, записавши відповідне значення в регістр керування модуля I²C, і транзакція буде завершена. Неважко побачити, що підпрограма обробки переривання модуля I²C являє собою кінцевий автомат, який при кожному перериванні перевіряє вміст регістра стану та виконує необхідні дії. Такий автомат легко реалізується за допомогою оператора switch().

```

void I2CISR (void) // Підпрограма обробки
{ // переривання I2C
    switch (I2STAT) // Зчитуємо код стану і
    { // виконуємо наступну дію
        case (0x08): // START
            I2CONCLR // Скидаємо прапорець
            стану // СТАРТ
            I2DAT = I2CAddress; // Посилаємо адресу і біт запису
            break;
        case (0x18): // SLA+W, ACK
            I2DAT // Пишемо дані в
            реєстр // передачі
            break;
        case (0x20): // SLA+W, NACK
            I2DAT // Повторно поси-
            лаємо адресу і // біт запису
            break;
        case (0x28): // Дані, ACK
            I2CONSET // Формуємо СТОП
            break;
        default:
            break;
    }
    I2CONCLR = 0x08; // Скидаємо прапорець
    // переривання I2C
    VICVectAddr // Скидаємо пере-
    ривання в VIC
}

```

У наведеному прикладі передається тільки один байт, однак цей приклад легко доопрацювати для передачі декількох байтів. Також можна додати додаткові оператори вибору для підтримки запитів даних від інших ведучих.

2.3.2 Режим ведучого приймача

У режимі ведучого приймача здійснюється прийом даних від веденого передавача. Передача ініціалізується тим же самим способом, що і в режимі ведучого передавача. Після генерації ведучим умови START, підпрограма обробки переривання повинна завантажити адресу веденого і біт напряму даних у реєстр даних I²C (I2DAT), а потім скинути прапорець SI в реєстрі I2CONSET.

Біт напрямку передачі даних (R/W) повинен бути встановлений в логічну 1, що відповідає зчитуванню даних. Таким чином, перший переданий байт повинен містити адресу веденого і біт зчитування (R).

Після передачі ведучим адреси веденого і біта напрямку даних і отримання від веденого біта «підтвердження» (ACK), мікроконтролер встановлює біт SI і повертає відповідний код стану в регістрі стану I²C (I2STAT). Дії, які повинні бути зроблені для користувача програмним забезпеченням у відповідності зі значенням цього коду стану, наведені в таблиці 3.

Формат обміну шиною I²C для режиму ведучого приймача зображено на рисунку 6. На рисунку 7 показаний формат обміну шиною I²C при перемиканні інтерфейсу I²C мікроконтролера з режиму ведучого приймача в режим ведучого передавача після генерації умови повторний START.



Рисунок 6 – Формат обміну по шині I²C для режиму ведучого приймача



Рисунок 7 – Формат обміну шиною I²C для режиму ведучого приймача, а потім для режиму ведучого передавача

Таблиця 3 – Коди стану для режиму I²C ведучого приймача

Код стану(I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			ST A	STO	SI		AA
08h	Передана умова START	Завантажити адресу веденого +R	X	0	0	X	Адреса веденого + R буде передана в лінію; біт ACK буде прийнятий з лінії
10h	Передана умова повторний START	Завантажити адресу веденого +R	X	0	0	X	Як і в попередньому випадку, адреса веденого +R буде передана в лінію
		Завантажити адресу веденого +W	X	0	0	X	I2C перейде в режим ведучого передавача
38h	Арбітраж втрачено під час прийому біта NOT ACK	Немає операцій з I2DAT	0	0	0	X	Шина I2C буде «відпущена»; I2C перейде в режим веденого
		Немає операцій з I2DAT	1	0	0	X	Умова START буде передана в лінію, коли шина звільниться
40h	Адреса веденого +R була передана в лінію; біт ACK було прийнято	Немає операцій з I2DAT	0	0	0	0	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт NOT ACK
		Немає операцій з I2DAT	0	0	0	1	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт ACK

Продовження таблиці 3

Код стану(I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення					Наступна дія апаратної логіки інтерфейсу
		Операції з I2DAT	Операції з бітами I2CONSET				
			ST A	STO	SI	AA	
48h	Адреса веденого +R була передана в лінію; біт NOT ACK було прийнято	Немає операцій з I2DAT	1	0	0	X	Умова повторний START буде передана в лінію
		Немає операцій з I2DAT	0	1	0	X	Умова STOP буде передана в лінію; прапорець STO буде скинуто
		Немає операцій з I2DAT	1	1	0	X	Умова STOP буде передана в лінію, після якої буде передана умова START; прапорець STO буде скинуто
50h	Байт даних було прийнято; біт ACK було передано в лінію	Прочитати байт даних	0	0	0	0	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт
		Прочитати байт даних	0	0	0	1	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт ACK
58h	Байт даних було прийнято; біт NOT ACK було передано в лінію	Прочитати байт даних	1	0	0	X	Умова повторний START буде передана в лінію
		Прочитати байт даних	0	1	0	X	Умова STOP буде передана в лінію; прапорець STO буде скинуто
		Прочитати байт даних	1	1	0	X	Умова STOP буде передана в лінію, після якої буде передана умова START; прапорець STO буде скинуто

Для запуску транзакції читання можна використовувати ту ж функцію I2CISR(), додавши в обробник переривання нові варіанти оператора switch():

```
case (0x40):          // SLA+R, ACK
I2CONSET  = 0x04;    // Дозволяємо підтвердження для байтів
                                // даних
break;

case (0x48.):        // SLA+R, NACK
I2CONSET  = 0x20;    // Знову формуємо стан СТАРТ
break;

case (0x50):        // Дані прийняті, ACK
message   = I2DAT;
I2CONSET  = 0x10;    // Формуємо стан СТОП
Lock      = 0;      // Індикуємо закінчення активності на шині
break;

case (0x58);        // Дані прийняті, NACK
I2CONSET  = 0x20;    // Знову формуємо стан СТАРТ
break;
```

2.3.3 Режим веденого приймача

У режимі веденого (Slave) приймача мікроконтролер одержує байти даних від ведучого (Master) передавача. Перехід в режим веденого приймача здійснюється наступним чином: для користувача програма ініціалізує регістр адреси веденого (I2ADR), записуючи в нього задане значення адреси, а також ініціалізує регістр керування I²C (I2CONSET), записуючи в нього значення, які вказані в таблиці 4.

Біт I2EN (прапорець дозволу) регістра I2CONSET повинен бути встановлений, щоб дозволити роботу інтерфейсу I²C в мікроконтролері. Встановлення в логічну 1 біта AA (прапорець підтвердження) вказує інтерфейсу I²C мікроконтролера, що він повинен після отримання з лінії власної адреси веденого або адреси загального виклику видавати в лінію біт «підтвердження» (ACK). Біти (прапорці) STA, STO і SI повинні бути скинуті.

Після ініціалізації регістрів I2ADR і I2CONSET, інтерфейс I2C веденого очікує звернення від зовнішнього (ведучого) пристрою за його (веденого) індивідуальною адресою веденого або мережевою адресою загального виклику.

Таблиця 4 – Ініціалізація регістра I2CONSET для режиму веденого

Номер біта	Назва біта	Значення біта
0	–	–
1	–	–
2	AA	1
3	SI	0
4	STO	0
5	STA	0
6	I2EN	1
7	–	–

Характер цього звернення (запис або читання) ідентифікується веденим за значенням біта напрямку даних в посилці, що надійшла від ведучого. Якщо біт напрямку скинутий (НИЗЬКИЙ рівень відповідає режиму запису в лінію), значить, ведучий пристрій має намір передавати дані, а інтерфейс I²C мікроконтролера переходить в режим веденого приймача.

Після отримання адреси веденого і біта напрямку даних, мікроконтролер у режимі веденого приймача встановлює біт SI і повертає відповідний код стану в регістр стану (I2STAT). Дії, які повинні бути зроблені для користувача програмним забезпеченням у відповідності зі значенням цього коду стану, перераховані в таблиці 5.

Формат обміну шиною I²C для режиму веденого приймача показаний на рисунку 8.



Рисунок 8 – Формат обміну шиною I²C для режиму веденого приймача

Таблиця 5 – Коды стану для режиму I²C веденого приймача

Код стану(I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			ST A	STO	SI		AA
60h	Індивідуальна адреса веденого +W була отримана з лінії; біт АСК було повернено(передано в лінію)	Немає операцій з I2DAT	X	0	0	0	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт NOT АСК
		Немає операцій з I2DAT	X	0	0	1	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт АСК
68h	Арбітраж було втрачено під час отримання від ведучого адреси веденого +R/W; індивідуальна адреса веденого +W була отримана; біт АСК було повернуто	Немає операцій з I2DAT	X	0	0	0	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт NOT АСК
		Немає операцій з I2DAT	X	0	0	1	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт АСК
70h	Адреса загального виклику (00h) була отримана; біт АСК було повернуто -	Немає операцій з I2DAT	X	0	0	0	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт NOT АСК
		Немає операцій з I2DAT	X	0	0	1	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт АСК

Продовження таблиці 5

Код стану(I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			ST A	STO	SI		AA
78h	Арбітраж було втрачено під час отримання від ведучого адреси веденого +R/W; адреса загального виклику була отримана; біт ACK було повернуто	Немає операцій з I2DAT	X	0	0	0	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт NOT ACK
		Немає операцій з I2DAT	X	0	0	1	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт ACK
80h	Попередньо адресована індивідуальна адреса веденого була отримана; байт даних було отримано; біт ACK було повернуто	Прочитати байт даних	X	0	0	0	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт NOT ACK
		Прочитати байт даних	X	0	0	1	Буде прийнято байт даних з лінії і повернено (передано в лінію) біт ACK

Продовження таблиці 5

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення					Наступна дія апаратної логіки інтерфейсу
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI	AA	
88h	Попередньо адресована індивідуальна адреса веденого була отримана; байт даних було отримано; біт NOT ACK було повернуто	Прочитати байт даних	0	0	0	0	Перемикання в режим, в якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись
		Прочитати байт даних	X	0	0	1	Перемикання в режим, в якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись
		Прочитати байт даних	1	0	0	0	Перемикання в режим, в якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику не будуть розпізнаватись; умова START буде передана в лінію, коли шина звільниться

Продовження таблиці 5

Код стану(I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
		Прочитати байт даних	1	0	0	1	Перемикання в режим, в якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1; умова START буде передана в лінію, коли шина звільниться
90h	Попередньо адресована адреса загального виклику буде отримана; байт даних було отримано; біт АСК було повернуто	Прочитати байт даних	X	0	0	0	Байт даних буде отримано і буде повернуто біт NOT ACK
		Прочитати байт даних	X	0	0	1	Байт даних буде отримано і буде повернуто біт АСК

Продовження таблиці 5

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
98h	Попередньо адресована адреса загального виклику буде отримана; байт даних було отримано; біт NOT ACK було повернуто	Прочитати байт даних	0	0	0	0	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись
		Прочитати байт даних	0	0	0	1	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1
		Прочитати байт даних	1	0	0	0	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись; умова START буде передана в лінію, коли шина звільниться

Продовження таблиці 5

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення					Наступна дія апаратної логіки інтерфейсу
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI	AA	
		Прочитати байт даних	1	0	0	1	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1; умова START буде передана в лінію, коли шина
A0h	Під час роботи I2C в режимі веденого приймача або веденого передавача, була отримана умова STOP або умова повторний START	Немає операцій з I2DAT	0	0	0	0	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись

Продовження таблиці 5

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення					Наступна дія апаратної логіки інтерфейсу
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI	AA	
A0h	Під час роботи I2C в режимі веденого приймача або веденого передавача, була отримана умова STOP або умова повторний START	Немає операцій з I2DAT	0	0	0	1	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1
		Немає операцій з I2DAT	1	0	0	0	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись; умова START буде передана в лінію, коли шина

Продовження таблиці 5

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
		Немає операцій з I2DAT	1	0	0	1	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1; умова START буде передана в

2.3.4 Режим веденого передавача

У режимі веденого передавача перший байт приймається від ведучого і обробляється аналогічно режиму веденого приймача, за винятком того, що встановлено біт напрямку даних (ВИСОКИЙ рівень відповідає читанню даних з лінії) сигналізує про намір ведучого приймати дані від веденого. По лінії SDA в послідовному вигляді ведений передає ведучому дані, в той час як по лінії SCL від ведучого до веденого надходить послідовність синхроімпульсів. Умови START і STOP розпізнаються веденим, як початок і кінець послідовної передачі.

Як і в режимі веденого приймача, в режимі веденого передавача апаратні засоби інтерфейсу I²C виконують порівняння прийнятої від ведучого адреси веденого зі своєю індивідуальною адресою веденого і адресою загального виклику. Якщо хоча б одна із цих адрес збігається з отриманим від ведучого, генерується переривання мікроконтролера за цією подією.

У таблиці 6 перераховані коди станів, що повертаються мікроконтролером в регістрі стану (I2STAT) для режиму веденого передавача, і дії, які повинні бути зроблені програмою користувача у відповідності зі значенням цього коду.

Таблиця 6 – Коди стану для режиму I²C веденого передавача

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
A8h	Індивідуальна мережева адреса веденого +R була отримана з лінії; біт АСК було повернуто (передано в лінію)	Завантажити байт даних	X	0	0	0	Останній байт даних буде передано в лінію; біт АСК буде прийнято з лінії
		Завантажити байт даних	X	0	0	1	Останній байт даних буде передано і біт АСК буде прийнято з лінії
B0h	Арбітраж було втрачено під час проходження адреси веденого +R/W от ведучого; індивідуальна адреса веденого +R була отримана; біт АСК було повернуто	Завантажити байт даних	X	0	0	0	Останній байт даних буде передано в лінію; біт АСК буде прийнято з лінії
		Завантажити байт даних	X	0	0	1	Останній байт даних буде передано і біт АСК буде прийнято з лінії

Продовження таблиці 6

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення					Наступна дія апаратної логіки інтерфейсу
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI	AA	
B8h	Байт даних з регістра I2DAT було передано в лінію; біт АСК було прийнято з лінії	Завантажити байт даних	X	0	0	0	Останній байт даних буде передано в лінію; біт АСК буде прийнято з лінії
		Завантажити байт даних	X	0	0	1	Останній байт даних буде передано і біт АСК буде прийнято з лінії
C0h	Байт даних з регістра I2DAT було передано в лінію; біт NOT АСК було прийнято	Немає операцій з I2DAT	0	0	0	0	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись
		Немає операцій з I2DAT	0	0	0	1	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1

Продовження таблиці 6

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
		Немає операцій з I2DAT	1	0	0	0	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись; умова START буде передана в лінію, коли шина звільниться
		Немає операцій з I2DAT	1	0	0	1	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1; умова START буде передана в лінію, коли шина звільниться

Продовження таблиці 6

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
C8h	Останній байт з регістра I2DAT було передано в лінію (AA = 0); біт ACK було прийнято з лінії	Немає операцій з I2DAT	0	0	0	0	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись
		Немає операцій з I2DAT	0	0	0	1	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1
		Немає операцій з I2DAT	1	0	0	0	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого або адреса загального виклику не будуть розпізнаватись; умова START буде передана в лінію, коли шина звільниться

Продовження таблиці 6

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
		Немає операцій з I2DAT	1	0	0	1	Перемикання в режим, у якому попередня адресація веденого відсутня; індивідуальна мережева адреса веденого буде розпізнаватись; адреса загального виклику буде розпізнаватись, якщо I2ADR[0] = 1; умова START

Якщо від керуючої програми мікроконтролера надходить команда перевести мікроконтролер з режиму веденого в режим ведучого, апаратні засоби інтерфейсу I²C чекають звільнення шини, щоб не перервати транзакції, які можливо здійснюються веденим. Якщо в режимі ведучого втрачається шинний арбітраж, інтерфейс I²C мікроконтролера негайно перемикається в режим веденого і може виявити свою індивідуальну адресу веденого в тій же самій послідовній передачі.

У таблиці 7 наведені значення кодів стану інтерфейсу I²C, які повертаються в регістрі стану (I2STAT) у разі помилок шини або інших некоректних ситуацій.

Таблиця 7 – Коди інших станів інтерфейсу I²C

Код стану (I2STAT)	Поточний стан шини I ² C і апаратної логіки інтерфейсу	Відповідь користувальницького програмного забезпечення				Наступна дія апаратної логіки інтерфейсу	
		Операції з I2DAT	Операції з бітами I2CONSET				
			STA	STO	SI		AA
F8h	Невизначений стан, при якому ніяка інформація недоступна; SI = 0	Немає операцій з I2DAT	Немає операцій з I2CONSET				Очікування чи продовження поточної передачі
00h	Помилка на шині у режимі ведучого або вибір режиму веденого через некоректні умови START або STOP. Стан 00h може також настати у разі взаємного накладення спроб інтерфейсу I2C перейти в невизначений стан	Немає операцій з I2DAT	0	1	0	X	У режимах ведучого або адресованого веденого задіяні тільки внутрішні апаратні засоби I2C. У будь-якому випадку, шина I2C буде «відпущена» і інтерфейс I2C переключиться в режим веденого без попередньої адресації; біт STO буде скинуто

Формат обміну шиною I²C для режиму веденого передавача показано на рисунку 9.



Рисунок 9 – Формат обміну шиною I²C для режиму веденого передавача

2.3.5 Взаємодія прикладної програми з модулем I²C

Взаємодія прикладної програми з модулем I²C базується на використанні переривання від модуля, яке виникає після кожної події, що відбулася на шині (прийом байта, формування станів СТАРТ/СТОП і т. ін.). Відповідно, під час передачі даних шиною програма може виконувати інші задачі. Якщо переривання від модуля I²C з якихось причин використовувати не можна, його можна заборонити. У цьому випадку програма повинна буде постійно слідкувати за станом прапорця SI для реагування на події, які відбуваються на шині.

Встановлення прапорця SI регістра I2CONSET означає, що модуль I²C закінчив виконання чергової операції і очікує реакції програми.

У старших п'яти розрядах регістра I2STAT при цьому формується відповідне значення, яке характеризує поточний стан шини I²C. Відповідно, програма повинна проаналізувати це значення і задати подальше поведіння модуля I²C, маніпулюючи вмістом регістрів I2CONSET і I2DAT.

На рисунку 10 показано взаємодію прикладної програми з модулем I²C на найпростішому прикладі передачі одного байта даних від ведучого до веденого.



Рисунок 10 – Приклад взаємодії програми з модулем I²C

Передача даних відбувається у наступній послідовності:

- першою операцією при передачі даних шиною I²C є формування стану СТАРТ. Для цього варто записати певне значення в регістр I2CONSET, відповідно до якого модуль I²C сформує на шині стан СТАРТ. При цьому для скидання прапорця SI розряд SIC регістра I2CONCLR повинен бути встановлений в «1». Формування стану СТАРТ почнеться відразу ж після скидання прапорця SI;
- після формування стану СТАРТ встановлюється прапорець SI. Число, яке перебуває у регістрі стану I2STAT, відображає результат виконання цієї операції;
- необхідно впевнитися в успішному формуванні стану СТАРТ, перевіривши вміст регістра I2STAT. Якщо код стану відповідає очікуваному, варто завантажити в регістр I2DAT вміст пакета SLA+W і сформувати в регістрі I2CONSET команду на передачу пакета (не забуваючи скинути при цьому прапорець SI). Передача адресного пакета почнеться відразу ж після скидання прапорця;
- по закінченні передачі адресного пакета встановлюється прапорець SI. Код стану, що перебуває в регістрі I2STAT, говорить про успішну передачу пакета, а також про одержання підтвердження від веденого пристрою;
- необхідно впевнитися в успішній передачі пакета та одержанні підтвердження, перевіривши вміст регістра I2STAT. Якщо код стану відповідає очікуваному, слід завантажити дані в регістр I2DAT, а потім сформувати в регістрі I2CONSET команду на передачу пакета (не забуваючи скинути при цьому прапорець SI). Передача пакета даних почнеться відразу ж після скидання прапорця;
- по закінченні передачі пакета даних встановлюється прапорець SI. Код стану, що перебуває в регістрі I2STAT, говорить про успішну передачу пакета, а також про одержання підтвердження від веденого пристрою;
- необхідно впевнитися в успішній передачі пакета та одержанні підтвердження, перевіривши вміст регістра I2STAT. Якщо код стану відповідає очікуваному, слід записати в регістр I2CONSET значення (не

забуваючи скинути при цьому прапорець SI), відповідно до якого модуль I²C сформує на шині стан СТОП або ПОВСТАРТ. Формування стану СТОП або ПОВСТАРТ почнеться відразу ж після скидання прапорця SI.

Зі сказаного видно, що будь-який етап взаємодії прикладної програми з модулем I²C складається із наступних трьох частин:

1) Після завершення модулем виконання чергової операції, він встановлює прапорець SI регістра I2CONSET, записує у регістр I2STAT код стану і очікує реакції програми. Доки прапорець встановлено в «1», на лінії SCL утримується НИЗЬКИЙ рівень.

2) Після встановлення прапорця SI користувач повинен занести в регістри модуля значення, які відповідають наступному етапу обміну.

3) Після оновлення вмісту регістрів модуля, користувач повинен сформувати в регістрі I2CONSET команду для виконання наступного етапу обміну. При завантаженні в регістр нового значення необхідно скинути прапорець SI записом логічної 1 в біт SIC регістра I2CONCLR. Після скидання прапорця модуль почне виконання операції, обумовленої вмістом регістра I2CONSET.

2.4 Опис виводів

Опис виводів інтерфейсу I²C мікроконтролера LPC2300 приведено в таблиці 8.

Таблиця 8 – Опис виводів інтерфейсу I²C

Назва виводу	Тип	Опис
SDA0, SDA1, SDA2	I/O	Послідовні дані. Вхід і вихід даних I ² C. Сполучена з цим ланцюгом лінія мікроконтролера, у відповідності зі специфікаціями I ² C, має вихід з відкритим колектором

Продовження таблиці 8

<i>Назва виводу</i>	<i>Тип</i>	<i>Опис</i>
SCL0, SCL1, SCL2	I/O	Послідовний синхросигнал. Вхід і вихід синхросигналу I ² C. Сполучена з цим ланцюгом лінія мікроконтролера, у відповідності зі специфікаціями I ² C, має вихід з відкритим колектором

2.5 Опис керуючих регістрів

Інтерфейс I²C мікроконтролерів сімейства LPC2300 включає в себе сім керуючих регістрів, які перераховано в таблиці 9.

Регістру керування відповідають два окремих регістри, які призначено для встановлення і скидання його керуючих бітів (I2CONSET і I2CONCLR). Швидкість передачі даних також визначається вмістом двох регістрів (I2CLH і I2CLL). Регістр стану I2STAT повертає коди стану, що відповідають різним подіям на шині. Регістр даних I2DAT використовується для запису кожного байта, який передається, або для зчитування кожного байта, який приймається. Коли мікроконтроллер працює в режимі веденого, його мережева адреса визначається вмістом регістра адреси I2ADDR.

Примітка. У мікроконтролерах LPC2300, що мають три інтерфейси I²C (I²C0, I²C1, I²C2), спеціальні регістри, що відносяться до інтерфейсу I²C0, мають адреси, що збігаються з адресами спеціальних регістрів інтерфейсу I²C в мікроконтролерах LPC2300 з одним інтерфейсом I²C.

2.5.1 Регістр керування встановленням

Регістр керування встановленням має адреси: для I²C0: I2CONSET або I2C0CONSET – 0xE001 C000, для I2C1: I2C1CONSET – 0xE005 C000, для I2C2: I2C2CONSET – 0xE008 0000.

Бітова структура регістра I2CONSET наведена в таблиці 10.

Таблиця 9 – Карта регістрів інтерфейсу I²C

Назва	Опис	Доступ	Значення після скидання ¹⁾	Адреса для I ² C0	Адреса для I ² C1	Адреса для I ² C2
I2CONSET	Регістр керування встановленням	Читання/запис	0	0xE001C000	0xE005C000	0xE0080000
I2STAT	Регістр стану	Тільки читання	0xF8	0xE001C004	0xE005C004	0xE0080004
I2DAT	Регістр даних	Читання/запис	0	0xE001C008	0xE005C008	0xE0080008
I2ADR	Регістр адреси веденого	Читання/запис	0	0xE001C00C	0xE005C00C	0xE008000C
I2SCLH	Регістр старшого напівслова швидкості передачі даних SCL	Читання/запис	0x04	0xE001C010	0xE005C010	0xE0080010
I2SCLL	Регістр молодшого напівслова швидкості передачі даних SCL	Читання/запис	0x04	0xE001C014	0xE005C014	0xE0080014
I2CONCLR	Регістр керування скиданням	Читання/запис	NA	0xE001C018	0xE005C018	0xE0080018

¹⁾ Значення відноситься до даних, що містяться тільки в використовуваних бітах, і не включає вміст зарезервованих бітів.

Таблиця 10 – Біти регістра керування I2CONSET

Біти I2CONSET	Назва	Опис	Значення після скидання
0	Зарезервовано	Користувальницьке програмне забезпечення не повинно проводити запис в зарезервовані біти. Читання зарезервованого біта повертає невизначене значення: NA (Not Applicable)	NA
1	Зарезервовано	Користувальницьке програмне забезпечення не повинно проводити запис в зарезервовані біти. Читання зарезервованого біта повертає невизначене значення: NA	NA
2	AA	Прапорець видачі біта «підтвердження»	0
3	SI	Прапорець переривання від I ² C	0
4	STO	Прапорець STOP	0
5	STA	Прапорець START	0
6	I2EN	Біт дозволу I ² C	0
7	Зарезервовано	Користувальницьке програмне забезпечення не повинно проводити запис в зарезервовані біти. Читання зарезервованого біта повертає невизначене значення: NA	NA

Нижче наведено пояснення окремих бітів регістра керування встановленням.

Біт AA – прапорець видачі біта «підтвердження». Якщо прапорець встановлений, мікроконтролер повертає в лінію SDA біт «підтвердження» (НИЗЬКИЙ рівень на лінії SDA), синхронізований відповідним «підтвердженню» тактовим імпульсом на лінії SCL. Це відбувається у наступних ситуаціях:

1. По шині SDA була прийнята адреса веденого, що співпадає з адресою, яка зберігається в регістрі адреси веденого I²C.
2. По шині SDA була прийнята адреса загального виклику, в той час як біт

загального виклику (GC) в регістрі I2ADR встановлений в 1.

3. По шині SDA був прийнятий байт даних, а інтерфейс I²C знаходиться в режимі ведучого приймача.

4. По шині SDA був прийнятий байт даних, а інтерфейс I²C знаходиться в режимі веденого приймача, до якого адресується звернення.

Біт AA може бути скинутий шляхом запису логічної одиниці в біт AAC регістра I2CONCLR. Коли біт AA скинутий, мікроконтролер повертає в лінію SDA біт «непідтвердження» (ВИСОКИЙ рівень на лінії SDA), синхронізований відповідним «підтвердженню» тактовим імпульсом на лінії SCL. Це відбувається у наступних ситуаціях:

1. По шині SDA був прийнятий байт даних, а інтерфейс I²C знаходиться в режимі ведучого приймача.

2. По шині SDA був прийнятий байт даних, а інтерфейс I²C знаходиться в режимі адресації веденого приймача.

Біт SI – прапорець переривання від I²C. Цей біт встановлюється, коли настає одне з 25 можливих активних станів інтерфейсу I²C. Якщо SI встановлено, низький рівень тактового сигналу на лінії SCL розтягується, а послідовний обмін буде призупинено. Коли рівень SCL високий, він не залежить від стану прапорця SI.

Біт SI може бути скинутий шляхом запису логічної 1 в біт SIC регістра I2CONCLR.

Біт STO – прапорець умови STOP. Встановлення цього біта змушує інтерфейс I²C генерувати на шині умову STOP в режимі ведучого або вийти зі стану помилки в режимі веденого. Коли біт STO встановлюється в режимі ведучого, на шині I²C генерується умова STOP. Коли апаратні засоби шини виявляють умову STOP, біт STO автоматично скидається.

У режимі веденого, встановлення цього біта виводить інтерфейс I²C зі стану помилки. У цьому випадку, умова STOP на шині не генерується, а апаратні засоби інтерфейсу I²C поведуться так, ніби умова STOP була отримана. Інтерфейс I²C перемикається в режим веденого приймача, до якого не адресувалося звернення.

Потім прапорець STO автоматично скидається апаратними засобами шини. **Біт STA – прапорець умови START.** Встановлення цього біта змушує інтерфейс I²C перейти в режим ведучого і згенерувати на шині I²C умову START або повторну умову START, якщо інтерфейс I²C вже знаходиться в режимі ведучого.

У разі, коли біт STA встановлюється в режимі веденого I²C, здійснюється перехід в режим ведучого, виконується перевірка шини і, якщо шина вільна, на ній генерується умова START. Якщо шина не вільна, то реалізуюча інтерфейс I²C логічна схема очікує умови STOP (яка звільнить шину), а потім після закінчення затримки, що дорівнює половині періоду внутрішнього тактового генератора, генерує умову START. Якщо інтерфейс I²C вже знаходиться в режимі ведучого, і прийняв або передав дані, то встановлення біта STA викличе генерацію умови «повторний START».

Біт STA може бути встановлений в будь-який час, включаючи моменти, коли інтерфейс I²C знаходиться в режимі веденого, до якого адресується звернення.

Біт STA може бути скинутий шляхом запису логічної 1 в біт STAC регістра I2CONCLR. Коли біт STA скинутий, умови START або повторний START не можуть бути згенеровані.

У ситуації, коли встановлені обидва біти STA і STO, на шині I²C буде згенеровано стан STOP, якщо інтерфейс знаходиться в режимі ведучого, а після цього буде згенеровано умову START. Якщо інтерфейс I²C знаходиться в режимі веденого, то буде згенерована внутрішня умова, відповідна умові STOP, однак умова STOP на шину I²C передана не буде.

I2EN – біт дозволу інтерфейсу I²C. Коли біт I2EN встановлений, робота інтерфейсу I²C мікроконтролера дозволена. Біт I2EN може бути скинутий шляхом запису логічної 1 в біт I2ENC регістра I2CONCLR. Коли біт I2EN скинутий, функція інтерфейсу I²C мікроконтролера заблокована.

2.5.2 Регістр керування скиданням

Регістр керування скиданням має адреси: для I²C0: I2CONCLR або I²C0 CONCLR – 0xE001 C018, для I2C1: I2C1CONCLR – 0xE005 C018, для I2C2:

I2C2CONCLR – 0xE008 0018.

Бітова структура регістра наведена в таблиці 11.

Таблиця 11 – Біти регістра керування скиданням

Біти I2CONCLR	Назва	Опис	Значення після скидання
0	Зарезервовано	Користувальницьке програмне забезпечення не має проводити запис в зарезервовані біти. Читання зарезервованого біта повертає невизначене значення: NA	NA
1	Зарезервовано	Користувальницьке програмне забезпечення не має проводити запис в зарезервовані біти. Читання зарезервованого біта повертає невизначене значення: NA	NA
2	AAC	Біт скидання прапорця видачі біта «підтвердження». Запис логічної 1 в цей біт скидає прапорець AA (очищає біт) в регістрі I2CONSET. Запис логічного 0 не призводить до яких-небудь наслідків	NA
3	SIC	Біт скидання прапорця переривання від I ² C. Запис логічної 1 в цей біт скидає прапорець SI в регістрі I2CONSET. Запис логічного 0 не призводить до яких-небудь наслідків	NA
4	Зарезервовано	Користувальницьке програмне забезпечення не має проводити запис в зарезервовані біти. Читання зарезервованого біта повертає невизначене значення: NA	NA
5	STAC	Біт скидання прапорця START. Запис логічної 1 в цей біт скидає прапорець STA в регістрі I2CONSET. Запис логічного 0 не призводить до яких-небудь наслідків	NA
6	I2ENC	Біт відключення інтерфейсу I ² C. Запис логічної 1 в цей біт скидає прапорець I2EN в регістрі I2CONSET. Запис логічного 0 не призводить до яких-небудь наслідків	NA
7	Зарезервовано	Користувальницьке програмне забезпечення не має проводити запис в зарезервовані біти. Читання зарезервованого біта повертає невизначене значення: NA	NA

2.5.3 Регістр стану

Регістр стану I²C має адреси: для I²C0: I2STAT або I2C0STAT – 0xE001

Цей регістр доступний тільки для читання. Він містить код стану інтерфейсу I²C. Молодші три біти цього регістра завжди дорівнюють 0. Всього наявні 26 можливих кодів стану, 25 з них активні. Код стану F8h означає, що стан інтерфейсу I²C не змінювався і біт (прапорець) SI регістра I2CONSET не встановлено (тобто його значення дорівнює 0). Кожен із 25 значень кодів стану, що залишились, відповідають певному стану інтерфейсу I²C. Коли настає будь-який з цих станів, автоматично встановлюється біт SI регістра I2CONSET.

Бітова структура регістра I2STAT наведена у таблиці 12.

Таблиця 12 – Біти регістра стану I²C

Біти I2STAT	Назва	Опис	Значення після скидання
2:0	Стан	Ці біти завжди дорівнюють 0	0
7:3	Стан	Біти (код) стану	0x1F

2.5.4 Регістр даних

Регістр даних має адреси: для I²C0: I2CDAT або I2C0DAT – 0xE001 C008, для I²C1: I2C1DAT – 0xE005 C008, для I²C2: I2C2DAT – 0xE008 0008.

Цей регістр містить дані, які будуть передані, або які були тільки що отримані через I²C. Центральний процесор може проводити запис і читання регістра даних у будь-який час, крім моменту, коли проводиться прийом або передача байта шляхом апаратного зсуву вмісту регістра. Звернення до регістра даних можливо тільки в тому випадку, коли встановлено біт (прапорець) SI регістра I2CONSET. Дані в регістрі I2DAT залишаються стійкими до тих пір, поки залишається встановленим біт SI. Дані в I2DAT завжди зсуваються справа наліво: першим передається MSB (біт 7), а після завершення прийому байта даних перший прийнятий біт цього байта даних розташовується в MSB регістра I2DAT. Бітова структура регістра I2DAT наведена в таблиці 13.

Таблиця 13– Біти регістра даних I²C

Біти I2DAT	Назва	Опис	Значення після скидання
7:0	Дані	Передані/прийняті біти даних	0

2.5.5 Регістр адреси веденого

Регістр адреси веденого має адреси: для I²C0: I2ADR або I2C0ADR – 0xE001 C00C, для I²C1: I2C1ADR – 0xE005 C00C, для I²C2: I2C2ADR – 0xE008 000C.

Цей доступний для читання і запису регістр використовується тільки тоді, коли інтерфейс I²C працює в режимі веденого. У режимі ведучого, вміст регістра I2ADR не має ніякого значення. Молодший біт (GC) регістра I2ADR є бітом дозволу розпізнавання логікою I²C адреси загального виклику (00h). Коли цей біт встановлено, адреса загального виклику (00h) буде розпізнаватися інтерфейсом I²C.

Бітова структура регістра I2ADR наведена в таблиці 14.

Таблиця 14– Біти регістра адреси веденого I²C

Біти I2ADR	Назва	Опис	Значення після скидання
0	GC	Біт дозволу розпізнавання загального виклику	0
7:1	Адреса	Адреса веденого в режимі веденого	0

2.5.6 Регістри програмування швидкості обміну I2SCL

Регістри програмування швидкості обміну I2SCL мають адреси: для I²C0: I2SCLH або I2C0SCLH – 0xE001 C010, для I²C1: I2C1SCLH – 0xE005 C010, для I²C2: I2C2SCLH – 0xE008 0010, для I²C0:I2SCLL або I2C0SCLL – 0xE001 C014, для I²C1:I2C1SCLL – 0xE005 C014, для I²C2:I2C2SCLL – 0xE008 0014.

Бітова структура регістрів I2SCLH, I2SCLL наведена відповідно у таблиці 15 і таблиці 16.

Таблиця 15– Біти регістра старшого доданку програмування швидкості обміну

Біти I2SCLH	Назва	Опис	Значення після скидання
15:0	Лічильні біти	Лічильні біти старшого доданку вибору періоду SCL	0x0004

Таблиця 16–Біти регістра молодшого доданку програмування швидкості обміну

Біти I2SCLL	Назва	Опис	Значення після скидання
15:0	Лічильні біти	Лічильні біти молодшого доданку вибору періоду SCL	0x0004

Програма–користувач може вибирати бажану швидкість передачі даних через інтерфейс I²C шляхом встановлення значень регістрів I2SCLH і I2SCLL. Вміст регістра I2SCLH являє собою один 16–бітний доданок числа, яке визначає тривалість (в циклах Pclk) періоду синхроімпульсів на лінії SCL інтерфейсу I²C, а вміст регістра I2SCLL – інший доданок. Частота синхроімпульсів SCL обчислюється у відповідності з наступною формулою:

$$\text{Частота передачі бітів} = \text{Pclk} / (2 * (\text{I2SCLH} + \text{I2SCLL})),$$

де (I2SCLH + I2SCLL) – значення суми вмісту регістрів I2SCLH та I2SCLL.

Значення I2SCLH і I2SCLL не обов'язково повинні бути однаковими. Вбудоване програмне забезпечення може задавати різну тривалість робочого циклу SCL, поміщаючи в I2SCLH і I2SCLL відповідні числа. Однак необхідно завжди стежити за тим, щоб частота передачі даних через інтерфейс I²C лежала в діапазоні 0...400 кГц. Крім того, на значення регістрів I2SCLH і I2SCLL накладається таке обмеження: кожне з чисел, що містяться в них, повинно бути більшим або рівним чотирьом (0004h).

Можливі значення частоти передачі бітів для інтерфейсу I²C, в залежності від вмісту регістрів I2SCLH, I2SCLL і значення дільника VPB наведені в таблицях 17, 18, 19.

Таблиця 17 – Вибір швидкості обміну через інтерфейс I²C для значення дільника VPB = 1

I2SCLL + I2SCLH	Частота передачі бітів [кГц] при P _{CLK} (МГц)			
	P _{CLK} =16 МГц	P _{CLK} = 20 МГц	P _{CLK} = 40 МГц	P _{CLK} = 60 МГц
8	–	–	–	–
10	–	–	–	–
25	–	–	–	–
50	320	400	–	–
75	213,333	266,667	–	–
100	160	200	400	–
160	100	125	250	375
200	80	100	200	300
320	50	62,5	125	187,5
400	40	50	100	150
510	31,373	39,216	78,431	117,647
800	20	25	50	75
1280	12,5	15,625	31,25	46,875

Таблиця 18 – Вибір швидкості обміну через інтерфейс I²C для значення дільника VPB = 2

I2SCLL + I2SCLH	Частота передачі бітів [кГц] при PCLK (МГц)			
	PCLK=16 МГц	PCLK = 20 МГц	PCLK = 40 МГц	PCLK = 60 МГц
8	–	–	–	–
10	–	–	–	–
25	320	400	–	–
50	160	200	400	–
75	106,667	133,333	266,667	400
100	80	100	200	300
160	50	62,5	125	187,5
200	40	50	100	150
320	25	31,25	62,5	93,75
400	20	25	50	75
510	15,686	19,608	39,216	58,824
800	10	12,5	25	37,5
1280	6,25	7,813	15,625	23,438

Таблиця 19 – Вибір швидкості обміну через інтерфейс I²C для значення дільника VPB = 4

I2SCLL+ I2SCLH	Частота передачі бітів [кГц] при P _{CLK} (МГц)			
	PCLK=16 МГц	PCLK= 20 МГц	PCLK= 40 МГц	PCLK = 60 МГц
8	500	–	–	–
10	400	–	–	–
25	160	200	400	–
50	80	100	200	300
75	53,333	66,667	133,333	200
100	40	50	100	150
160	25	31,25	62,5	93,75
200	20	25	50	75
320	12,5	15,625	31,25	46,875
400	10	12,5	25	37,5
510	7,843	9,804	19,608	29,412
800	5	6,25	12,5	18,75
1280	3,125	3,906	7,813	11,719

На рисунку 11 наведена спрощена структурна схема модуля I²C мікроконтролерів сімейства LPC2300.

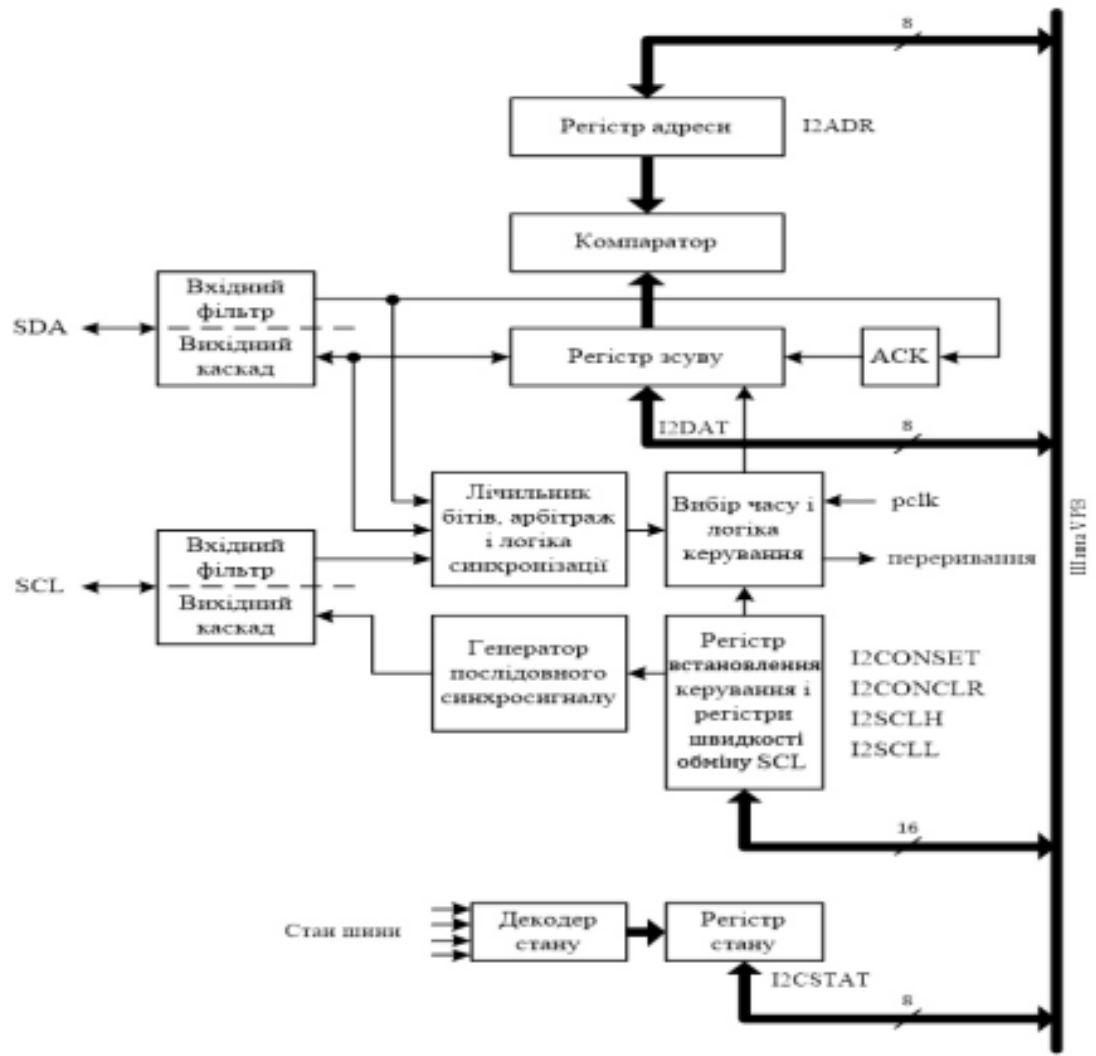


Рисунок 11 – Структурна схема модуля I²C

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Назвіть основні характерні особливості інтерфейсу I²C.
2. Наведіть приклади I²C–сумісних пристроїв.
3. З яких ліній складається шина інтерфейсу I²C?
4. Наведіть та опишіть структурну схему мережі з інтерфейсом I²C.
5. Назвіть коректні стани сигналів на шині під час обміну даними.
6. Назвіть та опишіть чотири режими (типи) обміну даними.
7. Наведіть та опишіть часові діаграми, які відображають обмін даними шиною I²C.
8. Опишіть типові особливості інтерфейсу I²C мікроконтролерів сімейства LPC2300
9. Опишіть роботу інтерфейсу I²C мікроконтролерів сімейства LPC2300 в режимі ведучого передавача.
10. Опишіть роботу інтерфейсу I²C мікроконтролерів сімейства LPC2300 в режимі ведучого приймача.
11. Опишіть роботу інтерфейсу I²C мікроконтролерів сімейства LPC2300 у режимі веденого приймача.
12. Опишіть роботу інтерфейсу I²C мікроконтролерів сімейства LPC2300 у режимі веденого передавача.
13. Опишіть взаємодію прикладної програми з модулем I²C.
14. Назвіть керуючі регістри модуля I²C.
15. Опишіть біти реєстр керування встановленням.
16. Опишіть біти реєстра стану.
17. Опишіть біти реєстра даних.
18. Опишіть біти реєстра адреси веденого.
19. Опишіть біти реєстрів програмування швидкості обміну.
20. Опишіть біти реєстра керування скиданням.
21. Опишіть структурну схему модуля I²C мікроконтролерів сімейства LPC2300.