3 ПОСЛІДОВНИЙ СИНХРОННИЙ ІНТЕРФЕЙС TWI (I^2C)

3.1 Характеристика інтерфейсу TWI (I^2C) на апаратному рівні

3.1.1 Загальна характеристика інтерфейсу I^2C

Інтерфейс I^2C (InterIC, або IIC) є двопровідним послідовним синхронним інтерфейсом, розробленим фірмою “Philips Corporation”, і призначений для зв'язку між інтегральними мікросхемами або модулями. Існує ціла група I^2C-сумісних пристроїв для різних додатків: цифро-аналогові та аналого-цифрові перетворювачі, мікросхеми пам'яті, мікроконтролери, що містять модуль I^2C і т.ін [1…7].

Шина інтерфейсу I^2C складається із двох ліній:

– двонаправленої лінії даних (SDA);

– лінії тактових (синхро)імпульсів (SCL).

На рисунку 3.1 наведена структурна схема типової мережі, що використовує для обміну даними інтерфейс (шину) I^2C.

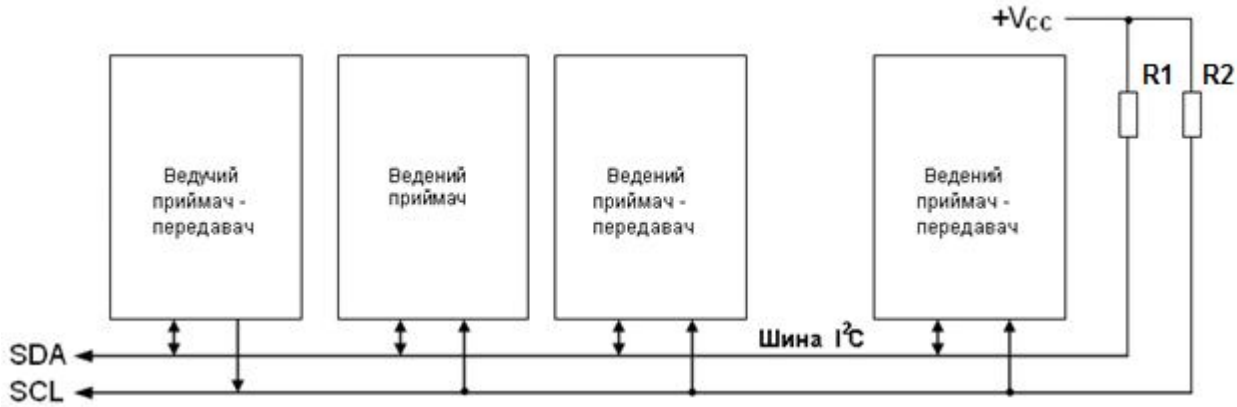


Рисунок 3.1 – Структурна схема мережі з інтерфейсом I^2C

Лінії SDA і SCL шини з'єднані з додатним полюсом напруги живлення (+*V*cc) через підтягуючі резистори, *R*1 та *R*2.

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.

Кожний пристрій, який використовує для обміну інтерфейс I^2C, має свою адресу. Коли ведучий пристрій бажає ініціювати обмін даними, він передає на лінію SDA адресу пристрою, з яким буде виконуватися обмін (передача/прийом). Всі ведені пристрої слідкують за адресою, яка виставляється на шину, і порівнюють її із власною адресою. Після адреси ведучий передає біт напряму $R/\overline{W}$, 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 I^2C може використовувати два формати адреси:

– 7-ми бітну адресу;

– 10-ти бітну адресу.

У AVR-мікроконтролерах сімейства Mega використовується лише 7-ми бітна адреса

На рисунку 3.2 наведений формат 7-ми бітної адреси.

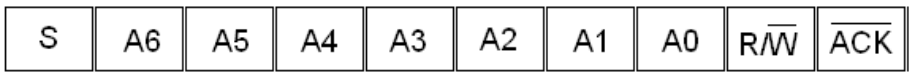
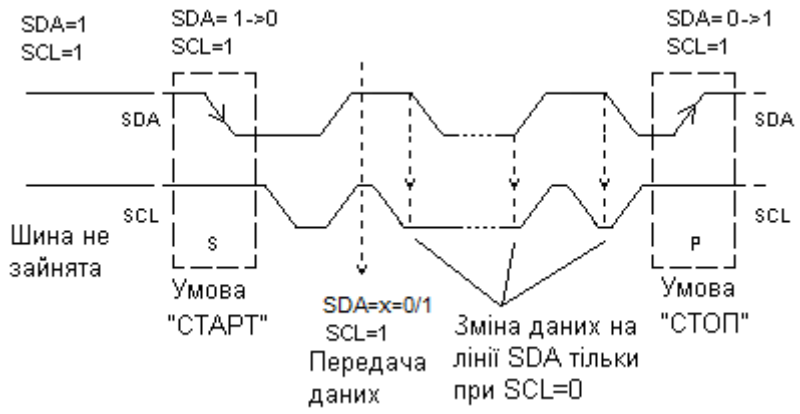


Рисунок 3.2 – Формат 7-бітної адреси

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

На рисунку 3.3 наведені часові діаграми, які відображають стани на шині (рисунок 3.3, а), а також пояснюють формування сигналу підтвердження (рисунок 3.3, б).



а)

б)

Рисунок 3.3 – Стани на шині: старт – S, стоп – P

а) - передача даних; б) - формування сигналу підтвердження

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);

– 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, переданих між цими станами, необмежена. Дані передаються побайтно. Приймач підтверджує одержання чергового байта, посилаючи 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 I^2C.

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 напряму 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 задання 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 напряму 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 вказання початку і кінця сеансу обміну послідовними даними.

3) Ведений приймач: ведучий передавач видає на лінію SDA адресу веденого приймача й біт напряму $R/\overline{W}=0$, 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 напряму $R/\overline{W}=1$, 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 I^2C здійснюють пошук своєї власної підпорядкованої адреси або адреси загального виклику. Якщо детектується одна із цих адрес, відбувається запит на переривання і виконуються відповідні дії. Якщо модуль I^2C хоче захопити шину й стати ведучим, то він чекає, поки шина звільниться (SDA = SCL = 1). Можливе функціонування в якості веденого при цьому не переривається.

Два і більше пристрої в I^2C 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 I^2C переключається у підпорядкований режим і може розпізнавати свою власну підпорядковану адресу.

На рисунку 3.4 наведені часові діаграми, які відображають обмін даними шиною I^2C.

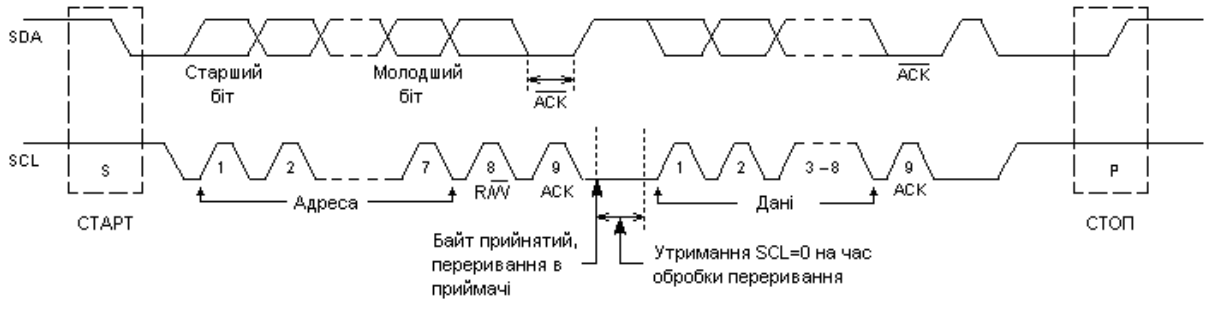


Рисунок 3.4 – Приклад обміну даними шиною I^2C

3.1.2 Архітектура інтерфейсу TWI мікроконтролерів сімейства AVR на апаратному рівні

3.1.2.1 Загальна характеристика

Модуль двопровідного послідовного інтерфейсу (Two Wire (Serial) Interface,) входить до складу практично всіх мікроконтролерів AVR сімейства Mega, за винятком моделей ATmega8515x, 162x. В моделях 165x, 325x, 3250x, 645x, 6450x функцію TWI може виконувати модуль USI, який розглядається в розділі CPC до лекції 12 [2].

Інтерфейс TWI являється повним аналогом базової версії інтерфейсу I^2C фірми «Philips». Інтерфейс TWI дозволяє об'єднати разом до 128 різних пристроїв за допомогою двонаправленої шини, яка складається всього із двох ліній: лінії тактового сигналу (SCL) і лінії даних (SDA). Єдиними додатковими елементами для реалізації шини є два підтягуючі резистори, по одному на кожну лінію (рисунок 3.5, резистори *R*1, *R*2).

Шинні формувачі на лініях всіх TWI-сумісних пристроїв виконуються за схемою з відкритим колектором (стоком), що дозволяє реалізувати функцію «монтажне І». Відповідно, НИЗЬКИЙ рівень на лінії встановлюється тоді, коли один або більше пристроїв виставляють на лінію сигнал лог. 0, а ВИСОКИЙ рівень на лінії встановлюються тоді, коли всі пристрої, які підключені до неї, встановлюють свої виходи в одиничний або третій стан.

Шина TWI повністю статична і швидкість обміну може бути якою завгодно низькою. Максимально допустима кількість мікросхем, під’єднаних до однієї шини, обмежується максимальною ємністю шини 400 пФ.

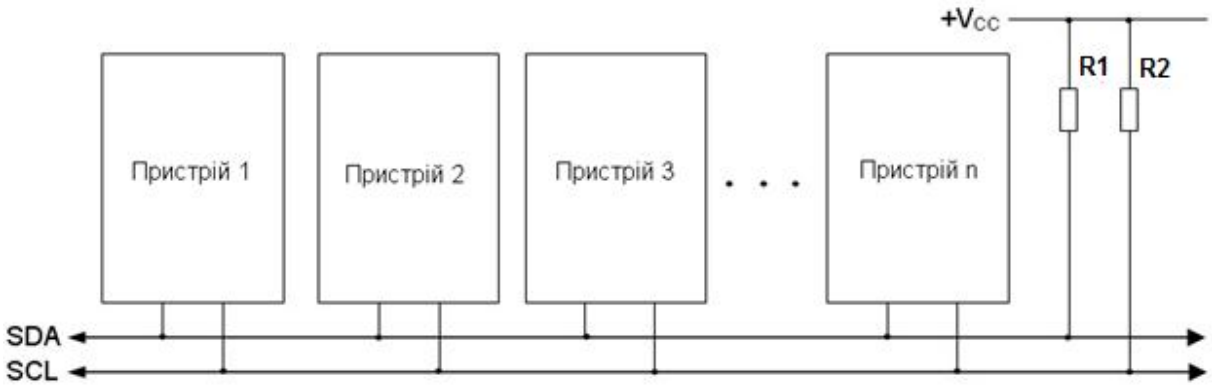


Рисунок 3.5 – З'єднання пристроїв за допомогою шини TWI

Основний режим роботи шини – 100 кбіт/с. В режимі із заниженою швидкістю швидкість передачі даних становить 10 кбіт/с. Зауважимо, що стандарт допускає призупинку тактування для роботи з повільними пристроями.

При наступному розгляді модуля TWI будуть використані терміни, опис яких наведено у таблиці 3.1.

Таблиця 3.1 – Терміни, що використовуються при описі модуля TWI

Термін	Опис
Ведучий (Master)	Пристрій, який ініціює і завершує передачу даних шиною, а також генерує тактовий сигнал SCL
Ведений (Slave)	Пристрій, який адресується ведучим
Передавач (Transmitter)	Пристрій, який видає дані на шину
Приймач (Reciever)	Пристрій, який зчитує дані з шини

3.1.2.2 Принципи обміну даними шиною TWI

3.1.2.2.1 Загальні відомості

Оскільки шина TWI являється послідовною, усі дані передаються нею (лінією SDA) порозрядно. Кожний переданий розряд супроводжується імпульсом на лінії тактового сигналу SCL. Причому сигнал на лінії SDA повинен бути стабільним увесь той час, поки на шині SCL присутній сигнал лог. 1 (рисунок 3.6). Єдиним винятком із цього правила є два особливих стани шини TWI: СТАРТ і СТОП.

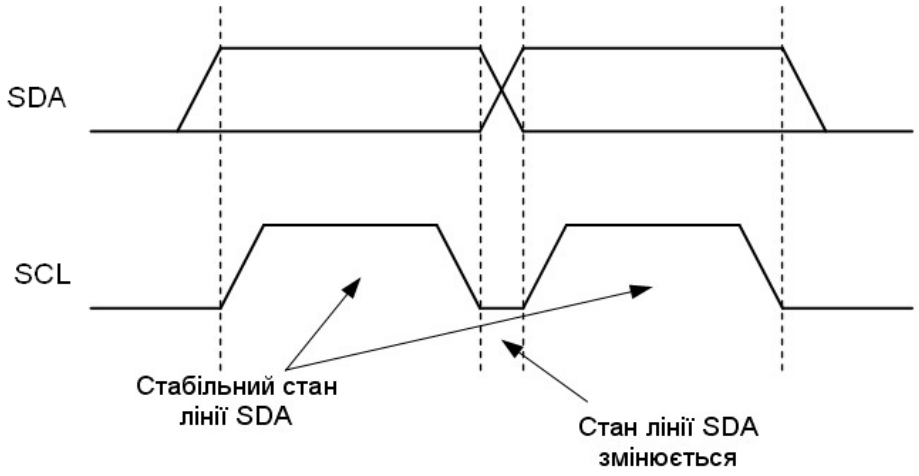


Рисунок 3.6 – Коректна видача даних на шину SDA

Стани СТАРТ і СТОП формуються ведучим на початку і наприкінці передачі даних відповідно. Між цими станами шина вважається зайнятою, й інші ведучі не повинні намагатися керувати нею.

Якщо ведучий хоче почати передачу нового блоку даних без втрати контролю над шиною, він може сформувати стан СТАРТ до формування стану СТОП. Сформований у такий спосіб стан називається «повторний СТАРТ» (ПОВСТАРТ), а шина вважається зайнятою до наступного формування стану СТОП. Оскільки така поведінка нічим не відрізняється від поведінки після формування стану СТАРТ, надалі обидва ці стани

(СТАРТ і ПОВСТАРТ) будуть позначатися як СТАРТ, крім тих випадків, де їх необхідно розрізняти.

Стани СТАРТ і СТОП формуються шляхом зміни рівня сигналу на лінії SDA при ВИСОКОМУ рівні на лінії SCL. Стану СТАРТ відповідає зміна рівня з «1» в «0», а стану СТОП – навпаки, з «0» в «1» (рисунок 3.7).

Зазначимо, що протокол інтерфейсу TWI дозволяє підключати до шини кілька ведучих пристроїв (режим Multi-Master). При цьому можуть виникати різні проблеми, однією з яких є розбіжність частот тактових сигналів, які генеруються різними ведучими.

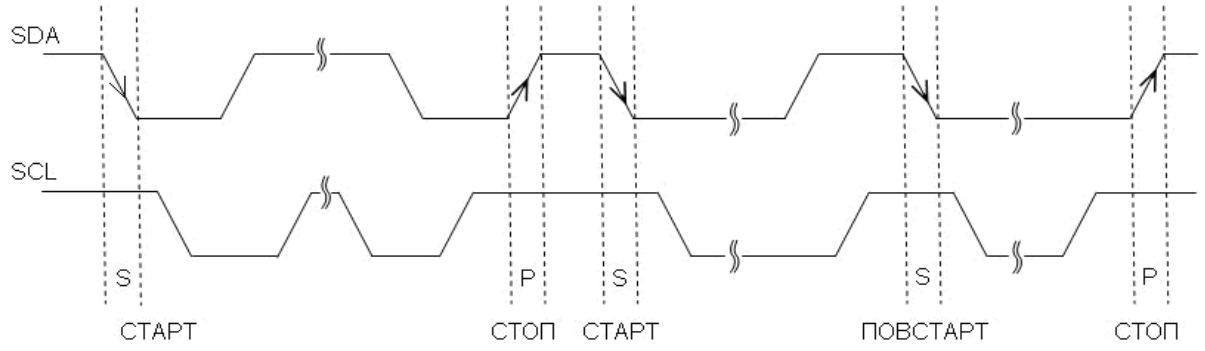


Рисунок 3.7 – Стани СТАРТ, СТОП і ПОВСТАРТ

Задача синхронізації вирішується дуже просто завдяки приєднанню всіх пристроїв до лінії SCL за схемою «монтажне І». У результаті тривалість тактових імпульсів на лінії SCL визначається тактовим сигналом з найменшою тривалістю імпульсів. А пауза між тактовими імпульсами, навпаки, визначається тактовим сигналом з найбільшою паузою між імпульсами. Крім цього, всі ведучі контролюють рівень, що присутній на лінії SCL, і визначають момент початку відліку імпульсу або паузи тактового сигналу за відповідною зміною сигналу (рисунок 3.8).

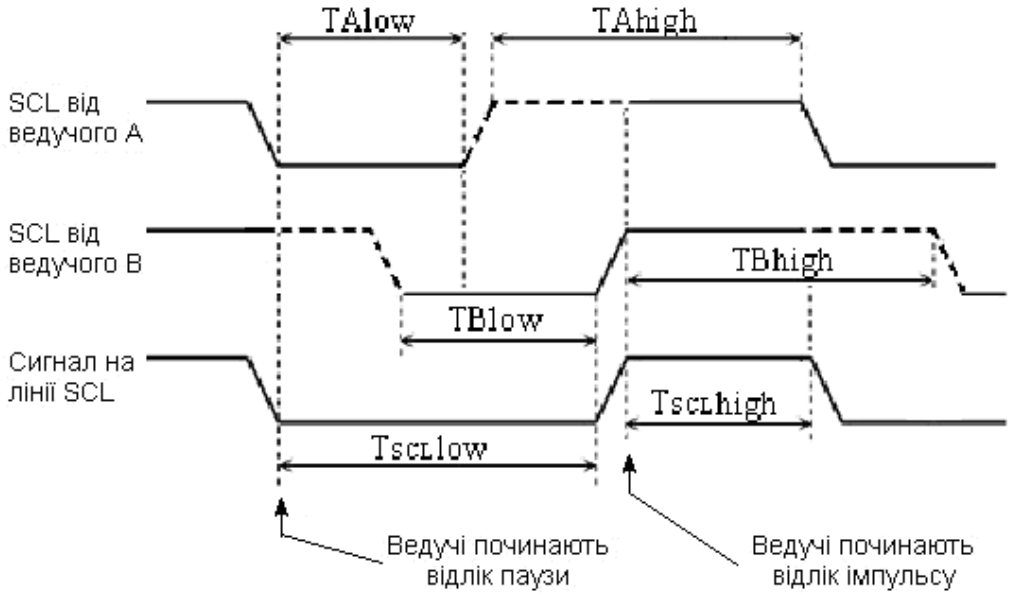


Рисунок 3.8 – Синхронізація сигналу SCL декількох ведучих

Іншою задачею, яку доводиться вирішувати при підключенні до шини декількох ведучих пристроїв, є задача розподілу пріоритетів, у випадку, якщо два і більше ведучих одночасно намагаються почати передачу. При виникненні такої ситуації передачу може здійснити тільки один ведучий, інші ж повинні переключитися в режим веденого. Причому передані дані під час розподілу пріоритетів не повинні спотворюватися.

Для розв’язання описаної задачі всі ведучі пристрої після видачі даних на лінію SDA контролюють її стан. Якщо стан лінії відрізняється від того, у який її переводив ведучий, він втрачає пріоритет (рисунок 3.9).

Ще раз нагадаємо, що таке можливо тільки у випадку видачі ним сигналу ВИСОКОГО рівня під час видачі іншими ведучими (які отримали пріоритет) сигналу НИЗЬКОГО рівня. Ведучий, який втратив пріоритет, повинен негайно переключитися в режим веденого і перевірити, чи не був він адресований яким-небудь із ведучих, що отримали пріоритет. При цьому він повинен продовжувати утримувати на лінії SDA сигнал ВИСОКОГО рівня. Однак він може продовжувати генерувати тактовий сигнал до закінчення передачі поточного пакета.

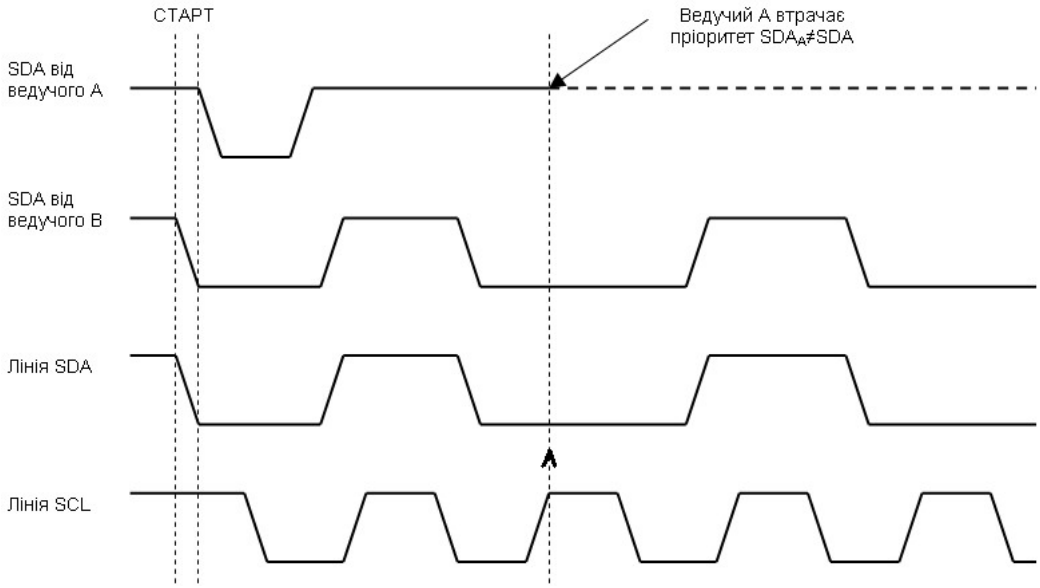


Рисунок 3.9 – Розподіл пріоритетів між двома ведучими

Процес розподілу пріоритетів триває доти, доки на шині не залишиться тільки один ведучий. У випадку, якщо декілька ведучих намагаються адресувати одного й того ж веденого, процес розподілу пріоритетів триває і під час передачі пакета даних. Звідси випливає, зокрема, що всі цикли обміну повинні містити однакову кількість пакетів даних, у іншому випадку результат процесу розподілу пріоритетів буде невизначеним.

При передачі даних шиною TWI разом з ними передається певна службова інформація. Сукупність даних і відповідної службової інформації називається пакетом. Розрізняють адресні пакети і пакети даних.

3.1.2.2.2 Формат адресного пакета

Усі адресні пакети, які передаються шиною TWI, мають довжину 9 біт. Пакет включає 7-розрядну адресу (першим передається старший

розряд), керуючий біт $R/\overline{W}$ і біт підтвердження $\overline{ACK}$ (рисунок 3.10).

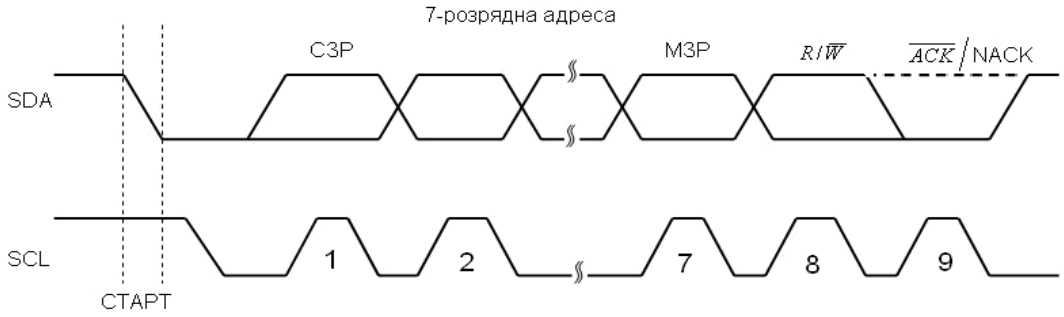


Рисунок 3.10 – Формат адресного пакета

Значення керуючого біта $R/\overline{W}$ визначає напрям передачі даних шиною. Скинутий в «0» біт означає передачу даних, а встановлений в «1» – запит даних (читання). Пакети зі скинутим і встановленим керуючим бітом позначаються відповідно $SLA+\overline{W}$ і $SLA+R$.

При розпізнаванні веденим своєї адреси він повинен під час 9-го тактового імпульсу сформувати підтвердження шляхом видачі на лінію SDA сигналу НИЗЬКОГО рівня ($\overline{ACK}$). Якщо ведений з якихось причин не може обслужити запит ведучого, він повинен під час 9-го тактового імпульсу утримувати на лінії SDA сигнал ВИСОКОГО рівня (NACK).

Веденим пристроям можуть бути призначені будь-які адреси, за винятком нульової адреси і адреси із діапазону «1111000...1111111». Доступних адрес, які можуть бути призначені веденим пристроям, всього 119. Нульова адреса зарезервована для реалізації так званих загальних викликів, а адреси формату «1111xxx», де x=0/1 повинні бути зарезервовані для використання в подальшому.

Загальний виклик використовується, коли ведучий хоче передати одне й те ж саме повідомлення всім веденим, підключеним до шини. При прийомі адресного пакета з нульовою адресою і скинутим керуючим бітом (запит на передачу) всі ведені пристрої, які підключені до шини, повинні сформувати підтвердження. Винятки становлять лише ті пристрої, у яких

розпізнавання загального виклику з якихось причин заборонено. Відповідно, наступні пакети даних, що посилаються ведучим, будуть отримані всіма веденими пристроями, які розпізнали адресу загального виклику.

Очевидно, що загальний виклик із встановленим керуючим розрядом (запит на читання) не має сенсу, оскільки різні пристрої не можуть одночасно здійснювати передачу даних шиною.

3.1.2.2.3 Формат пакета даних

Всі пакети даних, які передаються шиною TWI, теж мають довжину 9 біт. Пакет складається з байта даних (першим передається старший розряд) і біта підтвердження $\overline{ACK}$ (рисунок 3.11).

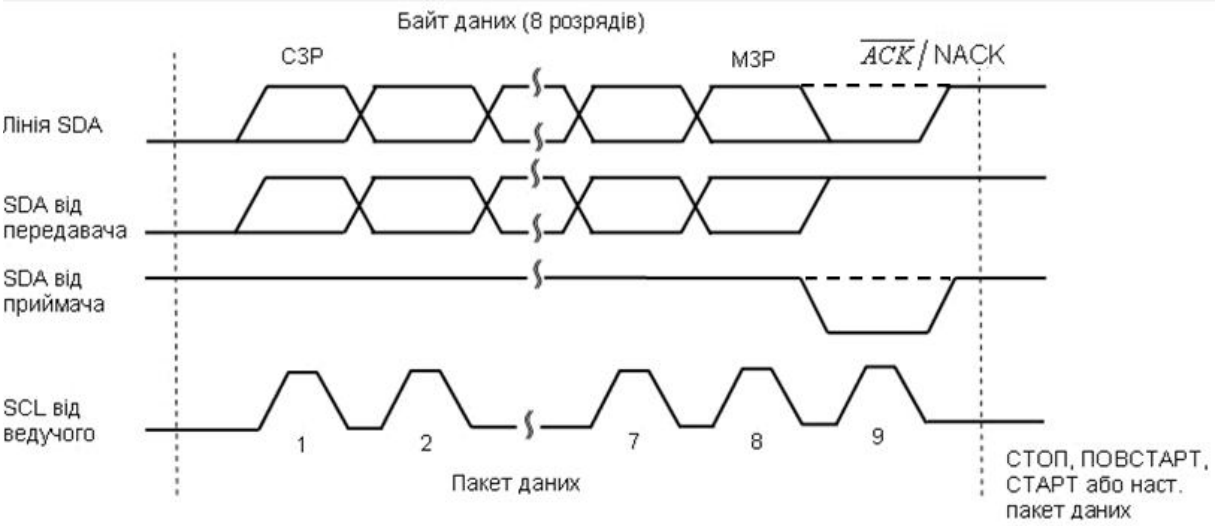


Рисунок 3.11 – Формат пакета даних

Генерація тактового сигналу і формування станів СТАРТ і СТОП здійснюється ведучим. Приймач, у свою чергу, повинен після прийому кожного байта формувати підтвердження ($\overline{ACK}$) шляхом видачі на лінію SDA сигналу НИЗЬКОГО рівня під час 9-го тактового імпульсу. Якщо

приймач одержав останній байт або з яких-небудь інших причин не може продовжувати прийом даних, він повинен під час 9-го тактового імпульсу утримувати на лінії сигнал ВИСОКОГО рівня (NACK).

Не одержавши підтвердження від приймача, ведучий, як правило, припиняє передачу даних, сформувавши стан СТОП.

На практиці кожен цикл обміну шиною TWI складається з наступних етапів (рисунок 3.12):

- формування стану СТАРТ;
- передача адресного пакета $SLA + R/\overline{W}$;
- передача одного або декількох пакетів даних;
- формування стану СТОП.

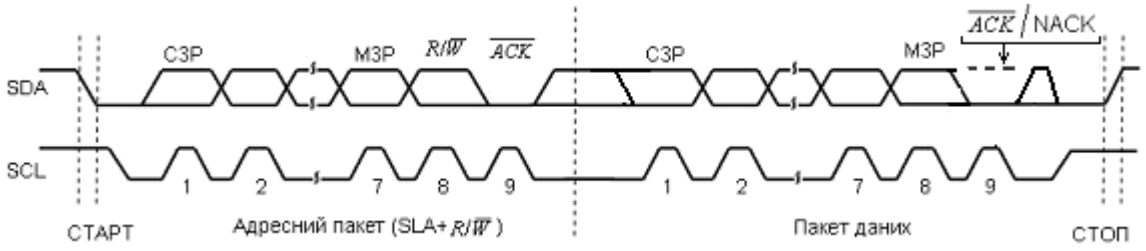


Рисунок 3.12 – Типовий цикл обміну

Швидкість обміну шиною TWI задається ведучим, тому що саме він генерує тактові імпульси. Однак, якщо ведений не може приймати дані з такою швидкістю або йому просто потрібен час на обробку даних між прийомом пакетів, він може збільшити паузу між тактовими імпульсами, утримуючи на лінії SCL сигнал НИЗЬКОГО рівня. На тривалість тактових імпульсів це не впливає.

3.1.3 Опис типового модуля TWI

3.1.3.1 Структура модуля

Структурна схема модуля TWI наведена на рисунку 3.13. Модуль містить у собі кілька блоків, кожний з яких виконує певні функції.

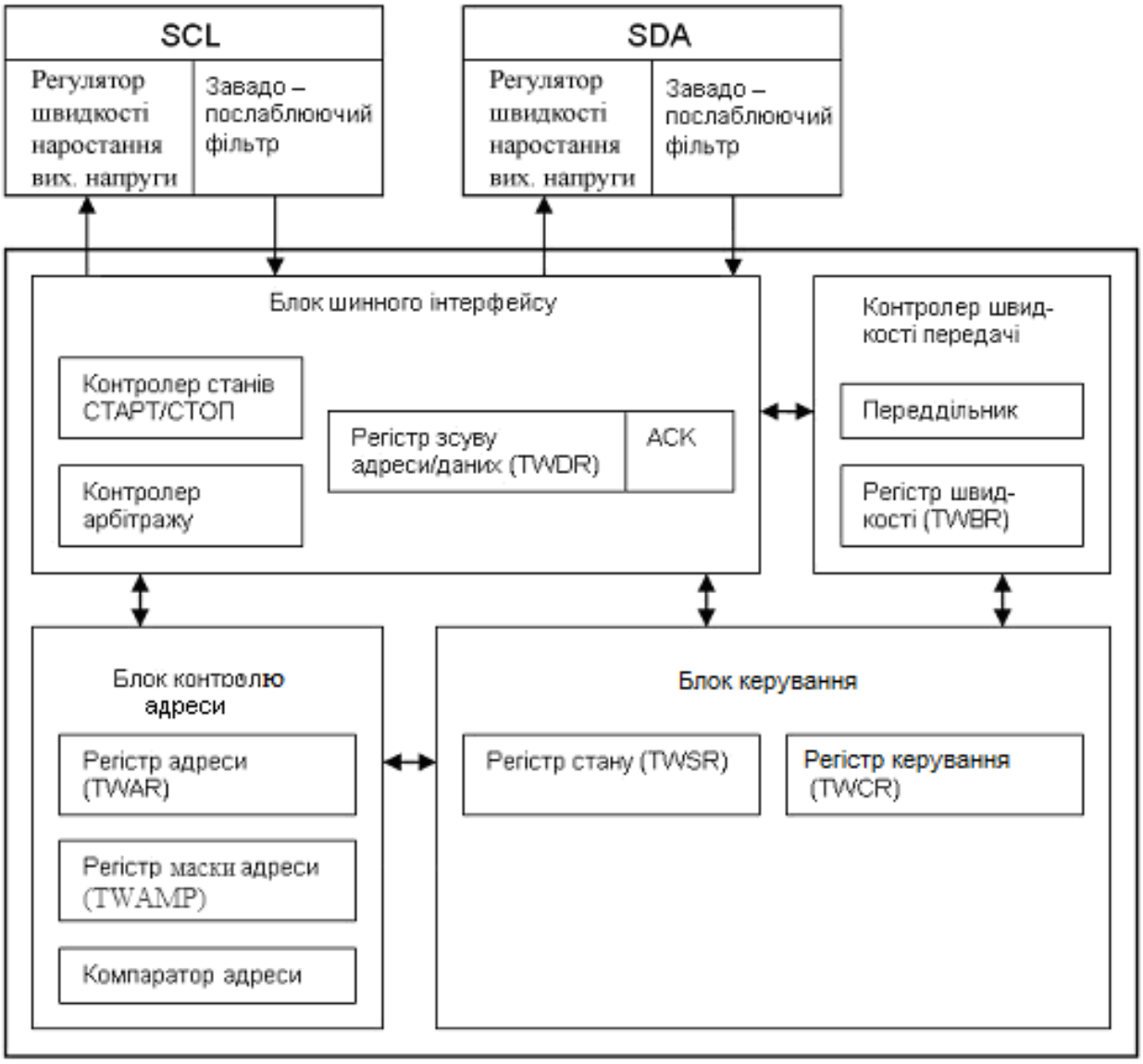


Рисунок 3.13 – Структурна схема модуля TWI

Взаємодія програми з модулем TWI здійснюється за допомогою п'яти (у нових моделях – шести) регістрів введення/виведення. Назви і

призначення цих регістрів, а також їх положення в адресному просторі регістрів статичної пам'яті даних та регістрів введення/виведення деяких моделей сімейства Mega наведено в таблиці 3.2.

Таблиця 3.2 – Регістри введення/виведення модуля TWI

Регістр	ATmega8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega164x/324x/644x	ATmega640x ATmega1280x/1281x ATmega2560x/2561x	Призначення
TWBR	$00 ($20)	$00 ($20)	($70)	($B8)	($B8)	($B8)	Регістр швидкості передачі
TWSR	$01 ($21)	$01 ($21)	($71)	($B9)	($B9)	($B9)	Регістр стану
TWAR	$02 ($22)	$02 ($22)	($72)	($BA)	($BA)	($BA)	Регістр адреси
TWDR	$03 ($23)	$03 ($23)	($73)	($BB)	($BB)	($BB)	Регістр даних
TWCR	$36 ($56)	$36 ($56)	($74)	($BC)	($BC)	($BC)	Регістр керування
TWAMR	-	-	-	($BD)	($BD)	($BD)	Регістр маски адреси

3.1.3.2 Виводи SCL і SDA

Ці виводи використовуються для підключення модуля до однойменних ліній шини TWI. У відповідності зі специфікацією інтерфейсу TWI, вихідні каскади, підключені до виводів, містять обмежувачі швидкості наростання сигналу, а вхідні каскади - фільтр, який придушує паразитні імпульси тривалістю менше 50 нс.

Обидва виводи є лініями портів введення/виведення мікроконтролерів (таблиця 3.3).

Таблиця 3.3 – Виводи деяких мікроконтролерів, що використовуються модулем TWI

Назва	ATmega8x	ATmega16x/32x	ATmega163x	ATmega48x/88x/168x	ATmega323x/324x ATmega464x/644x	ATmega64x/128x/640x ATmega1280x/1281x ATmega2560x/2561x	Призначення
SDA	PC4	PC1	PC1	PC4	PC1	PD1	Лінія даних
SCL	PC5	PC0	PC0	PC5	PC0	PD0	Лінія тактового сигналу

При використанні зазначених виводів мікроконтролера модулем TWI, до них, як і при звичайному їхньому використанні, можна підключити внутрішні підтягуючі резистори. Це дозволить у ряді випадків обійтися без зовнішніх підтягуючих резисторів, що необхідно згідно специфікації інтерфейсу TWI.

3.1.3.3 Контролер швидкості передачі (Bit Rate Generator)

Цей блок задає період проходження імпульсів лінією SCL при роботі мікроконтролера у режимі ведучого. Наприклад, у моделях ATmega163x і ATmega323x керування швидкістю передачі (частотою сигналу SCL) здійснюється за допомогою регістра TWBR. Частота сформованого сигналу SCL (при роботі пристрою в режимі ведучого) для цих моделей визначається виразом (3.1) і знаходиться в межах [0.03; 1] МГц [1,2].

$$f_{sck} = f_{clk} /(16 + 2TWBR) \quad (3.1)$$

У сучасних моделях сімейства Mega разом з регістром TWBR використовуються два молодших розряди (TWPS1:TWPS0) регістра TWSR. Для цих моделей частота сформованого сигналу SCL визначається виразом [1,2]:

$$f_{scl} = f_{clk} /(16 + 2TWBR \cdot 4^{TWPS}). \quad (3.2)$$

У наведених формулах (3.1) і (3.2) f_{scl} – швидкість передачі (частота сигналу SCL), f_{clk} – тактова частота процесора, TWBR – значення, записане в регістрі TWBR, TWPS – десяткове значення розрядів TWPS1:TWPS0 регістра TWSR.

Вміст регістра TWBR не повинен бути менше 10-ти, тому що ведучий може сформувати некоректний сигнал на шині SDA або SCL [1,2]. Значення TWPS обчислюється наступним чином:

TWPS1	TWPS0	Значення 4^{TWPS}
0	0	1
0	1	4
1	0	16
1	1	64

Регістр TWBR і розряди TWPS1:TWPSO регістра TWSR доступні як для читання, так і для запису в будь-який момент часу. При роботі в режимі веденого вміст зазначених розрядів не має значення, однак тактова частота мікроконтролера в цьому випадку повинна бути, як мінімум, в 16 разів вище частоти сигналу SCL [1,2].

Ще раз нагадуємо, що будь-який пристрій, підключений до шини TWI, може збільшувати тривалість паузи між тактовими імпульсами, знижуючи таким чином швидкість передачі даних шиною.

3.1.3.4 Блок шинного інтерфейсу (Bus Interface Unit)

До складу цього блоку окрім регістра зсуву входять два вузли:

- контролер станів СТАРТ/СТОП, який формує і виявляє стани СТАРТ, ПОВСТАРТ і СТОП. Моніторинг стану шини ведеться навіть при знаходженні мікроконтролера у «сплячому» режимі. Завдяки цьому при необхідності забезпечується вихід мікроконтролера зі «сплячого» режиму при адресації його яким-небудь ведучим;
- контролер арбітражу, який визначає наявність конфліктів на шині при роботі мікроконтролера в режимі ведучого. У випадку втрати пристроєм пріоритету контролер інформує про це блок керування, який робить необхідні дії та формує відповідні коди стану.

Крім того, до складу блоку входить регістр зсуву адреси/даних TWDR, який містить адресу і дані переданого або прийнятого пакета. Одночасно з передачею вмісту регістра на шину дані з неї записуються у цей регістр. Таким чином, майже завжди, за винятком моменту виходу мікроконтролера зі «сплячого» режиму, у регістрі TWDR утримується останній байт, що був на шині. При включенні живлення всі розряди цього регістра встановлюються в «1». При цьому ініціалізація регістра користувачем може бути здійснена тільки після генерації першого переривання [1,2].

Крім регістра TWDR у складі блоку є спеціальний регістр, один з розрядів якого містить значення переданого або прийнятого біта підтвердження. При прийомі стан цього біта, що передається передавачу, задається одним з розрядів регістра керування TWCR, а при передачі стан отриманого біта підтвердження може бути визначений кодом, що перебуває в регістрі стану TWSR.

3.1.3.5 Блок контролю адреси (Address Match Unit)

Блок контролю адреси перевіряє прийняту адресу на відповідність значенню, яке перебуває у старших семи розрядах регістра адреси TWAR. Також він перевіряє наявність загальних викликів, якщо дозволено їхнє розпізнавання. При виявленні коректної адреси він інформує про це блок керування, який формує або не формує підтвердження одержання адресного пакета. Контроль адресних пакетів здійснюється блоком навіть при знаходженні мікроконтролера у «сплячому» режимі, що дозволяє перевести мікроконтролер у робочий режим у випадку адресації його ведучим пристроєм. Формат регістра TWAR показаний на рисунку 3.14, а опис його розрядів наведено в таблиці 3.4.

	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Читання(R)/Запис($\overline{W}$)	$R/\overline{W}$	$R/\overline{W}$	$R/\overline{W}$	$R/\overline{W}$	$R/\overline{W}$	$R/\overline{W}$	$R/\overline{W}$	$R/\overline{W}$
Початкове значення	1	1	1	1	1	1	1	0

Рисунок 3.14 – Формат регістра TWAR

Таблиця 3.4 – Розряди регістра адреси TWAR

Розряд	Назва	Опис
7...1	TWA6...TWA0	Адреса пристрою У цих розрядах утримується адреса, на яку пристрій буде відзиватися при роботі в режимі веденого. При роботі пристрою в режимі ведучого вміст цих розрядів не має значення.
0	TWGCE	Дозвіл розпізнавання загальних викликів Якщо цей розряд встановлений в «1», пристрій буде озиватися на загальні виклики (пакети з адресою $00) так само, як і на виклики з адресою, що перебуває в розрядах TWA6...0 регістра. Якщо розряд скинутий в «0», розпізнавання загальних викликів заборонено.

У моделях ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega640x та ATmega1280x/1281x/2560x/2561x, крім регістра адреси

TWAR, є додатковий регістр маски адреси. При встановленні будь-якого біта регістру TWAMR в 1 він буде маскувати (анулювати) значення відповідного біта регістра адреси TWAR. Інакше кажучи, якщо біт регістра маски адреси встановлено в 1, то порівняння між відповідними бітами вмісту регістра TWAR і прийнятої адреси не відбувається. Формат регістра TWAMR наведено на рисунку 3.15, а спрощена структурна схема блоку контролю адреси розглянутих моделей наведено на рисунку 3.16.

	7	6	5	4	3	2	1	0
	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	–
Читання(R)/Запис(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Початкове значення	0	0	0	0	0	0	0	0

Рисунок 3.15 – Формат регістра TWAMR

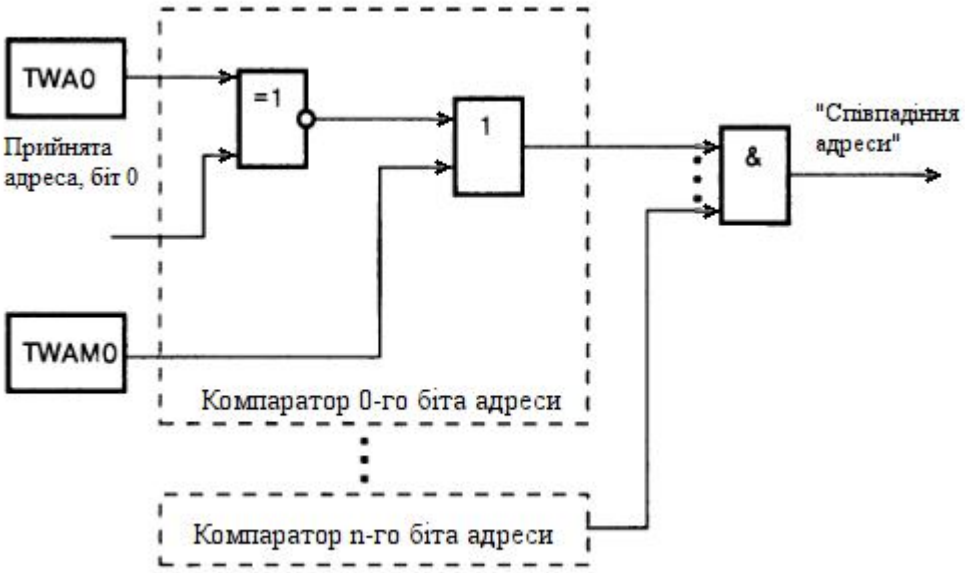


Рисунок 3.16 – Структурна схема блоку контролю адреси

3.1.3.6 Блок керування (Control Unit)

Цей блок здійснює керування модулем TWI відповідно до значень регістра керування TWCR та інформацією, яка надходить до нього від інших блоків модуля. При настанні певних подій, зазначених нижче, блок керування формує в регістрі стану TWSR код статусу, що відповідає події і встановлює прапорець запиту на переривання TWINT в регістрі TWCR. До моменту скидання цього прапорця на лінії SCL утримується НИЗЬКИЙ рівень, припиняючи тим самим передачу даних шиною.

Формування запиту на переривання здійснюється при виникненні наступних подій:

- закінчення формування стану СТАРТ/ПОВСТАРТ;
- закінчення передачі адресного пакета $(SLA + R/\overline{W})$;
- закінчення передачі байта даних;
- втрата пристроєм пріоритету;
- адресація пристрою або наявність загального виклику;
- закінчення прийому байта даних;
- виникнення помилок на шині, обумовлених неприпустимими умовами формування станів СТАРТ/СТОП.

Формат регістра TWCR показаний на рисунку 3.17, а опис його розрядів наведено в таблиці 3.5.

	7	6	5	4	3	2	1	0
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Читання(R)/Запис($\overline{W}$)	$R/\overline{W}$	$R/\overline{W}$	$R/\overline{W}$	$R/\overline{W}$	R	$R/\overline{W}$	R	$R/\overline{W}$
Початкове значення	1	1	1	1	1	1	0	0

Рисунок 3.17 – Формат регістра TWCR

Таблиця 3.5 – Розряди регістра керування TWCR

Розряд	Назва	Опис
7	TWINT	Прапорець переривання від модуля TWI Цей прапорець встановлюється апаратно після виконання чергової операції, коли модуль очікує відгук з боку програми. Якщо встановлені прапорці: I регістра SREG і TWIE регістра TWCR, генерується переривання і здійснюється виклик відповідного оброблювача. Доки прапорець TWINT встановлений, на лінії SCL утримується сигнал НИЗЬКОГО рівня. Скидання прапорця може бути здійснене тільки записом у нього лог. 1
6	TWEA	Дозвіл біта підтвердження Розряд TWEA керує формуванням біта підтвердження. Якщо цей розряд встановлений в «1», то пристрій формує сигнал підтвердження, коли це необхідно. При скиданні розряду в «0» пристрій віртуально відключається від шини TWI, тобто біт підтвердження не формується
5	TWSTA	Прапорець стану СТАРТ При запису в розряд TWSTA лог. 1 модуль перевіряє стан шини TWI та, якщо шина вільна, формує стан СТАРТ. Якщо шина зайнята, модуль TWI очікує появи стану СТОП, і тільки після цього формує стан СТАРТ. Прапорець скидається апаратно по закінченні формування стану СТАРТ
4	TWSTO	Прапорець стану СТОП У режимі ведучого встановлення прапорця TWSTO в «1» приводить до формування на шині стану СТОП. Прапорець скидається апаратно по закінченні формування стану СТОП. Встановлення прапорця TWSTO у режимі веденого може використовуватися для виходу з помилкових ситуацій. Після запису в цей розряд лог. 1 модуль TWI вертається в режим неадресованого веденого, а виводи SCL і SDA для передачі переводяться у третій стан. Стан СТОП не формується
3	TWWC	Прапорець конфлікту запису Прапорець встановлюється в «1» при спробі запису в регістр TWDR, коли прапорець переривання TWINT скинутий. Прапорець скидається при записі в регістр TWDR, коли прапорець переривання TWINT встановлений
2	TWEN	Дозвіл роботи модуля TWI Розряд TWEN керує функціонуванням модуля TWI. При записі в цей розряд лог. 1 модуль TWI включається і бере на себе керування контактами введення/виведення мікроконтролера, які відповідають виводам SCL і SDA. При скиданні розряду TWEN в 0 модуль TWI вимикається
1	—	Зарезервовано, читається як «0»
0	TWIE	Дозвіл переривання від модуля TWI Якщо в цьому розряді записана лог. 1 і прапорець I регістра SREG також встановлений в «1», то переривання від модуля TWI дозволено

Формат регістра TWSR показаний на рисунку 3.18, а опис його розрядів наведено в таблиці 3.6.

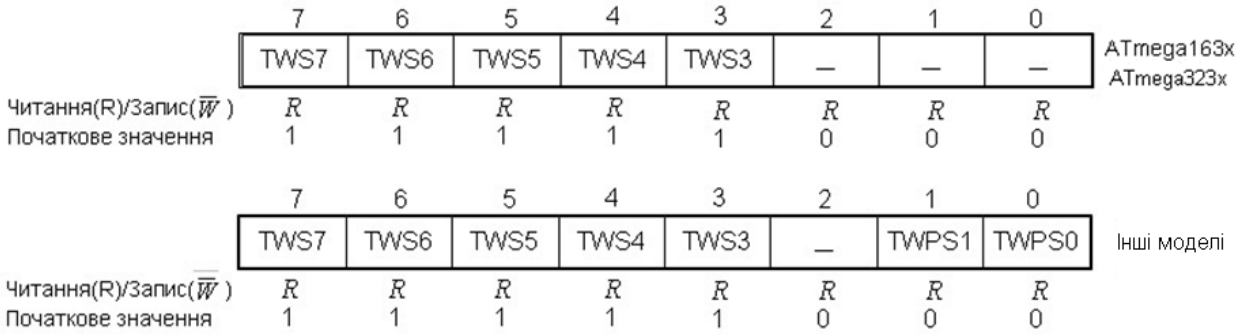


Рисунок 3.18 – Формат регістра TWSR

Таблиця 3.6 – Розряди регістра стану TWSR

Розряд	Назва	Опис
7...3	TWS7...TWS3	Стан модуля TWI Значення, яке утримується в цих розрядах, відображає стан вузлів модуля і шини TWI. Можливі коди статусу будуть описані при подальшому розгляді модуля TWI. Розряди TWS7...3 доступні тільки для читання
2	—	Зарезервовано, читається як «0»
1, 0	TWPS1, TWPS0	Коефіцієнт ділення попереднього дільника контролера швидкості передачі Стан цих розрядів визначає коефіцієнт ділення попереднього дільника контролера швидкості передачі, керуючий частотою сигналу SCL, що генерується (див. опис контролера швидкості передачі на початку цього підрозділу)

3.1.4 Взаємодія прикладної програми з модулем TWI

Взаємодія прикладної програми з модулем TWI 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 TWI з якихось причин використовувати не можна, його можна заборонити. У цьому випадку програма повинна буде постійно слідкувати за станом прапорця TWINT для реагування на події, які відбуваються на шині.

Встановлення прапорця TWINT регістра TWCR означає, що модуль TWI закінчив виконання чергової операції і очікує реакції програми.

У старших п'яти розрядах регістра TWSR при цьому формується відповідне значення, яке характеризує поточний стан шини TWI. Відповідно, програма повинна проаналізувати це значення і задати подальше поводження модуля TWI, маніпулюючи вмістом регістрів TWCR і TWDR.

На рисунку 3.19 показана взаємодія прикладної програми з модулем TWI на найпростішому прикладі передачі одного байта даних від ведучого до веденого.

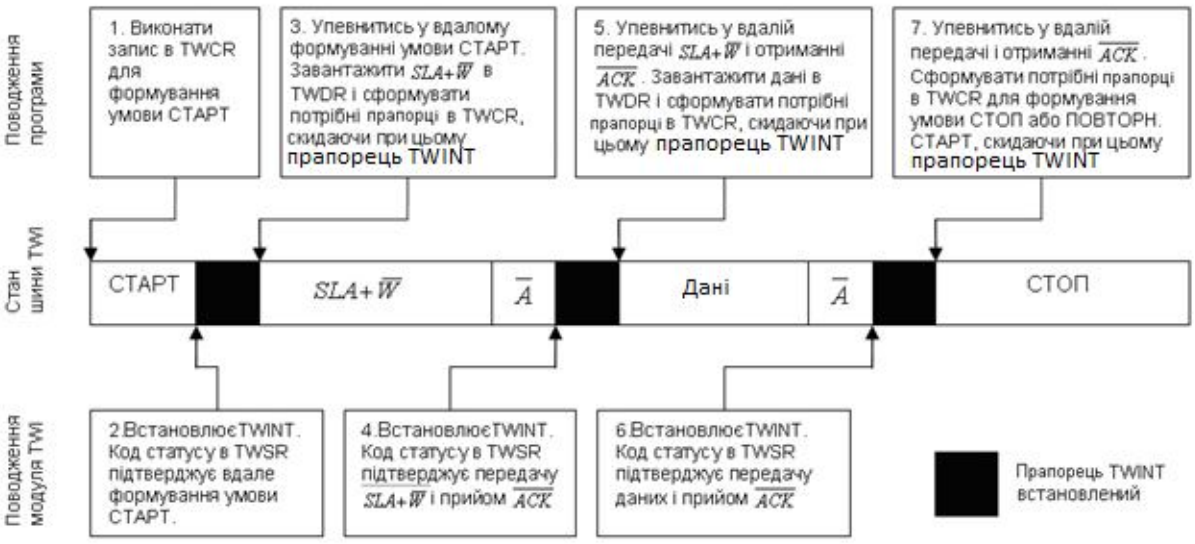


Рисунок 3.19 – Приклад взаємодії програми з модулем TWI

Передача даних відбувається у наступній послідовності.

1) Першою операцією при передачі даних шиною TWI є формування стану СТАРТ. Для цього варто записати певне значення в регістр TWCR, відповідно до якого модуль TWI сформує на шині стан СТАРТ. При цьому для скидання прапорця TWINT розряд, який відповідає прапорцю TWINT, повинен бути встановлений в «1». Формування стану СТАРТ почнеться відразу ж після скидання прапорця TWINT.

2) Після формування стану СТАРТ встановлюється прапорець TWINT. Число, яке перебуває у регістрі стану TWSR, відображає результат виконання цієї операції.

3) Необхідно впевнитися в успішному формуванні стану СТАРТ, перевіривши вміст регістра TWSR. Якщо код статусу відповідає очікуваному, варто завантажити в регістр TWDR вміст пакета $SLA + \overline{W}$ і сформувати в регістрі TWCR команду на передачу пакета (не забуваючи скинути при цьому прапорець TWINT). Передача адресного пакета почнеться відразу ж після скидання прапорця.

4) По закінченні передачі адресного пакета встановлюється прапорець TWINT. Код статусу, що перебуває в регістрі TWSR, говорить про успішну передачу пакета, а також про одержання підтвердження від веденого пристрою.

5) Необхідно впевнитися в успішній передачі пакета та одержанні підтвердження, перевіривши вміст регістра TWSR. Якщо код статусу відповідає очікуваному, слід завантажити дані в регістр TWDR, а потім сформувати в регістрі TWCR команду на передачу пакета (не забуваючи скинути при цьому прапорець TWINT). Передача пакета даних почнеться відразу ж після скидання прапорця.

6) По закінченні передачі пакета даних встановлюється прапорець TWINT. Код статусу, що перебуває в регістрі TWSR, говорить про успішну

передачу пакета, а також про одержання підтвердження від веденого пристрою.

7) Необхідно впевнитися в успішній передачі пакета та одержанні підтвердження, перевіривши вміст регістра TWSR. Якщо код статусу відповідає очікуваному, слід записати в регістр TWCR значення (не забуваючи скинути при цьому прапорець TWINT), відповідно до якого модуль TWI сформує на шині стан СТОП або ПОВСТАРТ. Формування стану СТОП або ПОВСТАРТ почнеться відразу ж після скидання прапорця TWINT.

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

1) Після завершення модулем виконання чергової операції, він встановлює прапорець TWINT регістра TWCR, записує у регістр TWSR код статусу і очікує реакції програми. Доки прапорець встановлений в «1», на лінії SCL утримується НИЗЬКИЙ рівень.

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

3) Після оновлення вмісту регістрів модуля, користувач повинен сформувати в регістрі TWCR команду для виконання наступного етапу обміну. При завантаженні в регістр нового значення необхідно скинути прапорець TWINT записом у нього лог. 1. Після скидання прапорця модуль почне виконання операції, обумовленої вмістом регістра TWCR.

Можливі значення кодів статусу, про які згадується в прикладі, будуть наведені далі при описі конкретних режимів роботи модуля TWI. Крім того, є два коди стану ($F8 і $00), не пов'язані з певним режимом роботи (таблиця 3.7).

Таблиця 3.7 – Коди статусу загального призначення

Код статусу	Стан шини і модуля TWI	Дії програми					Наступна дія, яку виконує модуль TWI
		в/із TWDR	Запис у регістр TWCR				
			TWSTA	TWSTO	TWINT	TWEA	
$F8	Немає інформації; TWINT = «0»	Немає дій	Немає дій				Чекати встановлення прапорця TWINT
$00	Помилка на шині в результаті некоректного формування стану СТАРТ або СТОП	Немає дій	0	1	1	x	Всі дії виконуються апаратно. Шина звільняється, а прапорець TWSTO скидається в «0»

У таблиці 3.7 та наступних таблицях, при описанні дій програми у відповідь на отримання відповідного коду статусу, розглядаються тільки чотири прапорці регістра TWCR із семи: TWSTA (STA), TWSTO (STO), TWINT і TWEA. Інші три прапорці мають такі значення: TWWC = x(0/1), що залежить від відсутності/наявності помилки запису при передачі; TWEN = 1 (модуль TWI- включений); TWIE = x(0/1) – забороняє/дозволяє переривання при встановленні прапорця TWINT.

Стан з кодом $F8 є проміжним. Наявність цього коду означає відсутність інформації у зв'язку з тим, що прапорець TWINT не встановлений.

Код статусу $00 сигналізує про помилку на шині. Така помилка виникає при формуванні ведучим станів СТАРТ або СТОП у неналежному місці, наприклад, під час передачі байта адреси, байта даних або біта підтвердження. Для виходу з таких ситуацій необхідно записати лог. 1 у розряди TWSTO і TWINT. У результаті модуль перейде в режим неадресованого веденого, звільнить лінії SDA і SCL і скине прапорець TWSTO. Стан СТОП у цій ситуації сформовано не буде.

3.1.5 Мікроконтролери AVR, які мають інтерфейс TWI

Таблиця 3.8 – Перелік мікроконтролерів AVR, які мають інтерфейс TWI

Сімейство	Модель мікросхеми
Xmega	всі моделі
Mega	ATmega48 ATmega8 ATmega88 ATmega8535 ATmega16 ATmega168 ATmega32 ATmega644 ATmega640 ATmega128 ATmega1281 ATmega1280 ATmega2561 ATmega2560
Tiny	ATtiny48 ATtiny88
picoPower	ATmega48P ATmega88P ATmega168P ATmega328P* ATmega324P ATmega644P ATmega1284P**
SmartBattery AVR	ATmega406
Automotive	ATmega164P Automotive ATmega324P Automotive ATmega644P Automotive ATmega48 Automotive ATmega88 Automotive ATmega168 Automotive AT90CAN32 Automotive AT90CAN64 Automotive AT90CAN128 Automotive

3.1.6 Пристрої, які підтримують інтерфейс TWI

Нижче наведений приклад деяких пристроїв, які підтримають інтерфейс TWI (таблиця 3.9).

Таблиця 3.9 – Пристрої, які підтримують інтерфейс TWI

Пристрій	Опис пристрою
CP2120	Виробник: Silabs *Інтегральний контролер моста SPI в TWI* Напруга живлення: 2.7…3.6 В Швидкість прийому даних веденим SPI-інтерфейсом: до 1.0 Mbit/s Швидкість передачі даних ведучим TWI-інтерфейсом: до 400 kHz
PCA8581 PCA8581C	Виробник: NXP *Постійний запам'ятовуючий пристрій, який програмується та використовує електричне стирання інформації* Максимальна частота SCL-сигналу: 100 КГц Максимальна напруга живлення: 4,5…5,5 В (PCA8581) Максимальна напруга живлення: 2,5…6,0 В (PCA8581C)
TSA6057	Виробник: Philips Semiconductors *Синтезатор частоти* Єдиний задаючий генератор (4 МГц) для діапазонів AM та FM Налаштовувана мережа частот 1, 10, 25 КГц Програмний контроль переключення діапазонів
TDA8424 (TDA8421, TDA8425, TDA8426)	Виробник: PhilipsSemiconductors *Стереофонічний аудіопроцесор* Регулювання тембру по високим і низьким частотам
PCF8574	Виробник: NXP *8-ми бітовий перетворювач даних від шини TWI* Діапазон робочої напруги: 2,5…6,0 В Споживання струму в режимі очікування: 10 мкА 8-ми бітовий двонаправлений I/O порт Корпус: DIP16 абоSO16
PCF8583	Виробник: NXP Semiconductors *Годинник/Календар з 240*8 біт RAM* Керування по інтерфейсу TWI Діапазон робочої напруги: 2,5…6,0 В Діапазон робочої напруги для годинника: 1,0…6,0 В Максимальний робочий струм: 50 мкА 240*8 біт низьковольтне RAM Базова частота: 32768 та 50 Гц

Продовження таблиці 3.9

Пристрій	Опис пристрою
DS1621	Виробник: Dallas Semiconductor *Цифровий термометр і термостат* Час перетворення сигналу: 1 с. Діапазон температур: -55…+125 (крок 0,5 С) або -67…+257 (крок 0,9 F)
PCF8591	Виробник: NXP Semiconductors *8-ми бітовий АЦП/ЦАП* Діапазон робочої напруги: 2,5…6,0 В (одне джерело живлення) Використання АЦП з послідовним наближенням та 8-розрядним діапазоном чисел Мультиплексований ЦАП з одним аналоговим виходом
TDA1551Q	Виробник: Philips Semiconductors *Мостовий підсилювач потужності (ПНЧ) із вбудованими засобами діагностики стану* Керування і контроль за допомогою TWI Наявність режимів блокування звуку та «сну», що передаються по TWI

3.2 Програмування інтерфейсу TWI

3.2.1 Режими роботи модуля TWI

3.2.1.1 Загальна характеристика

Модуль TWI, реалізований у мікроконтролерах сімейства Mega, може працювати в наступних режимах [1,2]:

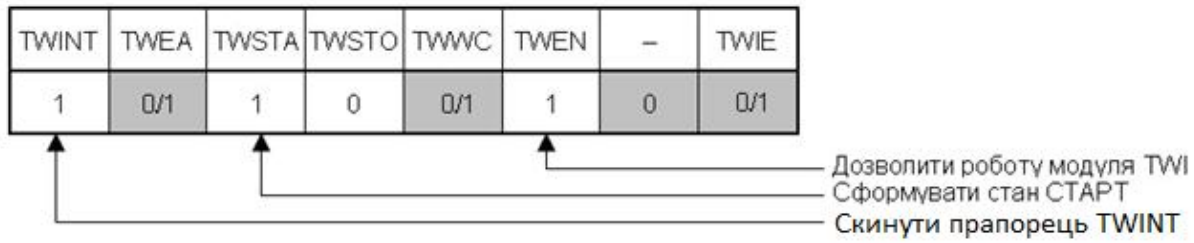
- ведучий передавач (MasterTransmitter);
- ведучий приймач (MasterReceiver);
- ведений передавач (SlaveTransmitter);
- ведений приймач (SlaveReceiver).

Вибір конкретного режиму визначається логікою роботи програми і, відповідно, виконуваними діями.

3.2.1.2 Режим «Ведучий передавач»

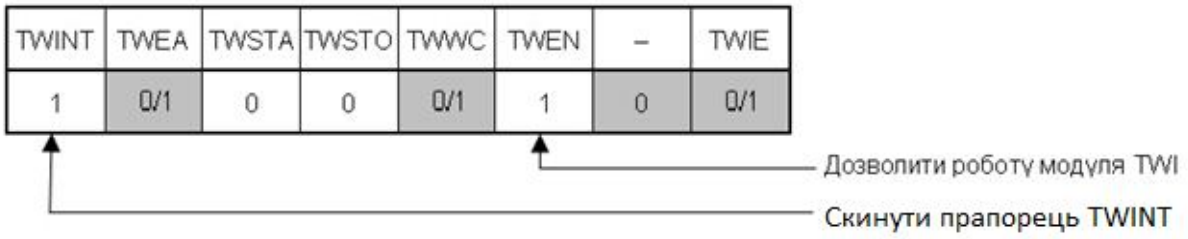
У режимі «Ведучий передавач» (MasterTransmitter) здійснюється передача даних від ведучого пристрою до веденого. Для перемикання пристрою в режим «Ведучий» необхідно записати у регістр TWCR керуюче слово, яке переводить модуль TWI у стан СТАРТ. Формат адресного пакета, який потім передається, визначає, в якому з режимів буде працювати ведучий. При передачі пакета $SLA + \overline{W}$ модуль переходить у режим «Ведучий передавач», а при передачі пакета $SLA + R$ - у режим «Ведучий приймач».

Формування стану СТАРТ почнеться після запису в регістр TWCR наступного значення:



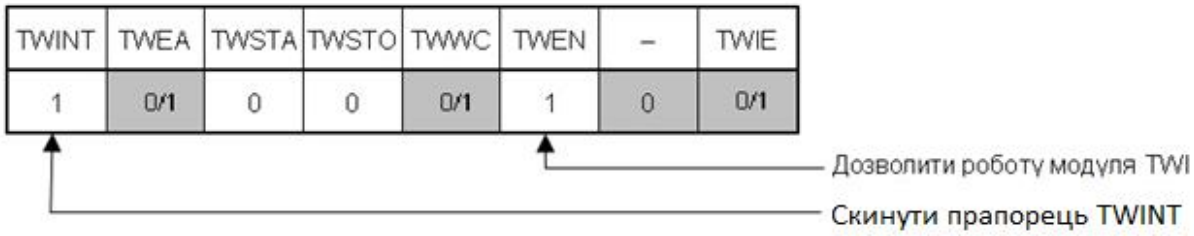
У результаті запису зазначеного значення модуль TWI почне контролювати стан шини і сформує стан СТАРТ відразу ж, як тільки вона стане вільною.

По закінченні формування стану СТАРТ встановлюється прапорець TWINT; код статусу повинен при цьому мати значення $08 (таблиця 3.10). Для перемикання модуля в режим «Ведучий передавач» необхідно передати шиною пакет $SLA + \overline{W}$. Для цього вміст пакета завантажується в регістр TWDR, а в регістр TWCR заноситься наступне значення:

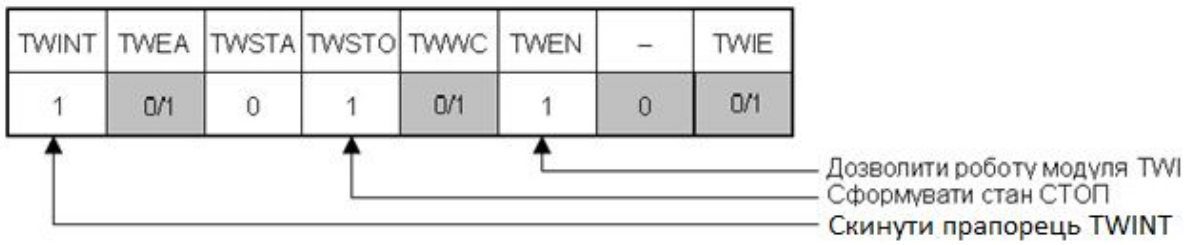


Після передачі адресного пакета і прийому біта підтвердження прапорець TWINT знову встановлюється в «1». Код статусу на цьому етапі може мати одне з наступних значень: $18, $20 або $38. Які дії необхідно виконати при виявленні того або іншого коду, докладно розглянуто в таблиці 3.10.

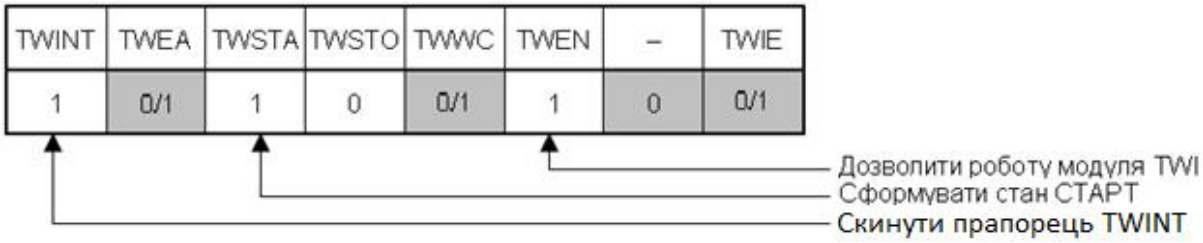
Після передачі адресного пакета повинні бути передані пакети даних. Значення байта даних завантажується в регістр TWDR. Передача пакета даних починається після запису в регістр TWCR наступного значення:



Описана процедура використовується для передачі всіх пакетів даних. Після передачі останнього байта даних, ведучий повинен сформувати на шині стан СТОП або ПОВСТАРТ. Формування стану СТОП почнеться після запису в регістр TWCR наступного значення:



Для формування стану ПОВСТАРТ у регістр TWCR необхідно занести наступне значення:



Після формування на шині стану ПОВСТАРТ (код статусу $10) ведучий може адресувати той же або інший ведений, не формуючи стан СТОП. Інакше кажучи, використання стану ПОВСТАРТ дозволяє здійснювати зміну ведених пристроїв, а також переключатися між режимами «Ведучий передавач» і «Ведучий приймач» без втрати контролю над шиною.

Таблиця 3.10 – Коди статусу для режиму «Ведучий передавач»

Код статусу	Стан шини та модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$08	Було сформовано стан СТАРТ	Завантажити $SLA+\overline{W}$	0	0	1	X	1	Буде передано $SLA+\overline{W}$ і отримано $\overline{ACK}$ або $NACK$
$10	Було сформовано стан ПОВСТАРТ	Завантажити $SLA+\overline{W}$	0	0	1	X	1	Буде передано $SLA+\overline{W}$ і отримано $\overline{ACK}$ або $NACK$
		Завантажити $SLA+R$	0	0	1	X	1	Буде передано $SLA+R$. Модуль переключиться в режим «Ведучий приймач»

Продовження таблиці 3.10

Код статусу	Стан шини та модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$18	Було передано пакет $SLA+\overline{W}$ і прийнято підтвердження ($\overline{ACK}$)	Завантажити дані	0	0	1	X	1	Буде передано байт даних і отримано $\overline{ACK}$ або $NACK$
		Немає дій	1	0	1	X	1	Буде сформовано стан ПОВСТАРТ
		Немає дій	0	1	1	X	1	Буде сформовано стан СТОП (прапорець TWSTO буде скинуто)
		Немає дій	1	1	1	X	1	Буде сформовано стан СТОП, а потім стан СТАРТ (прапорець TWSTO буде скинуто)
$20	Було передано пакет $SLA+\overline{W}$ і прийнято непідтвердження ($NACK$)	Завантажити дані	0	0	1	X	1	Буде передано байт даних і отримано $\overline{ACK}$ або $NACK$
		Немає дій	1	0	1	X	1	Буде сформовано стан ПОВСТАРТ
		Немає дій	0	1	1	X	1	Буде сформовано стан СТОП (прапорець TWSTO буде скинуто)
		Немає дій	1	1	1	X	1	Буде сформовано стан СТОП, а потім стан СТАРТ (прапорець TWSTO буде скинуто)
Код статусу	Стан шини та модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	

Продовження таблиці 3.10

Код статусу	Стан шини та модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$28	Було передано пакет даних і прийнято підтвердження ($\overline{ACK}$)	Завантажити дані	0	0	1	X	1	Буде передано байт даних і отримано $\overline{ACK}$ або *NACK*
		Немає дій	1	0	1	X	1	Буде сформовано стан ПОВСТАРТ
		Немає дій	0	1	1	X	1	Буде сформовано стан СТОП (прапорець TWSTO буде скинуто)
		Немає дій	1	1	1	X	1	Буде сформовано стан СТОП, а потім стан СТАРТ (прапорець TWSTO буде скинуто)
$30	Було передано пакет даних і прийнято непідтвердження (*NACK*)	Завантажити дані	0	0	1	X	1	Буде передано байт даних і отримано $\overline{ACK}$ або *NACK*
		Немає дій	1	0	1	X	1	Буде сформовано стан ПОВСТАРТ
		Немає дій	0	1	1	X	1	Буде сформовано стан СТОП (прапорець TWSTO буде скинуто)
		Немає дій	1	1	1	X	1	Буде сформовано стан СТОП, а потім стан СТАРТ (прапорець TWSTO буде скинуто)
$38	Втрата пріоритету при передачі пакета адреси або даних	Немає дій	0	0	1	X	1	Пристрій звільнить шину і перейде в режим неадресованого веденого
		Немає дій	1	0	1	X	1	Після звільнення шини буде сформовано стан СТАРТ

У підрозділі 3.2.2 наведена схема алгоритму роботи модуля в даному режимі, а у 3.2.3– програмна реалізація цього алгоритму.

На рисунку 3.20 наведені різні стани модуля TWI у режимі «Ведучий передавач».

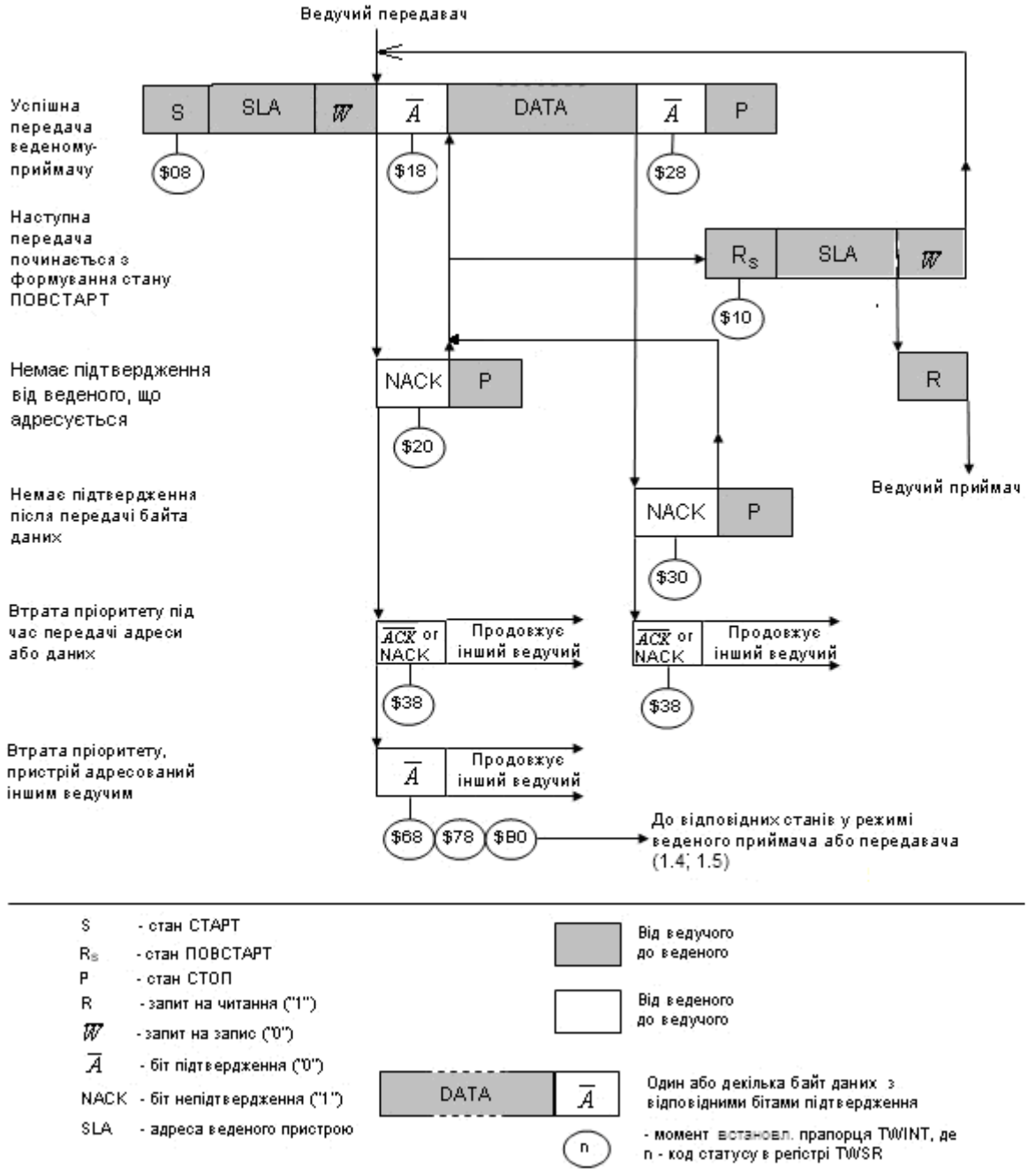
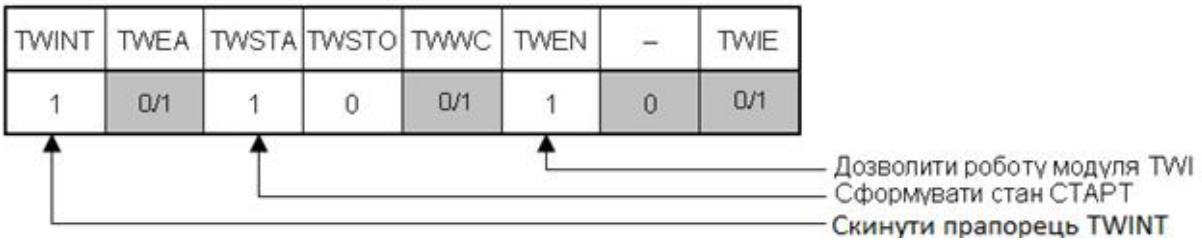


Рисунок 3.20 – Стани модуля TWI у режимі “Ведучий передавач”

3.2.1.3 Режим «Ведучий приймач»

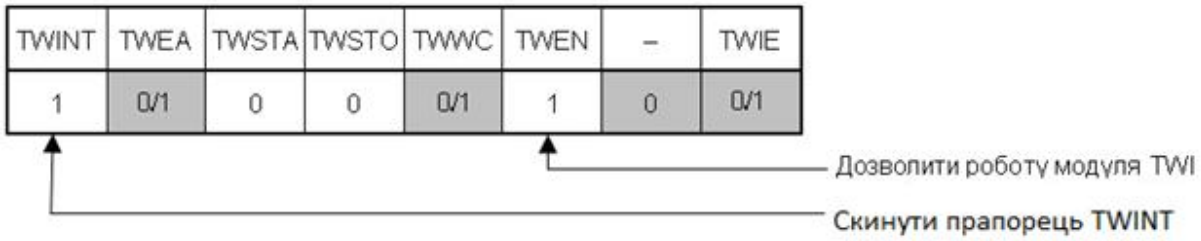
У режимі «Ведучий приймач» ведучий здійснює прийом даних від веденого пристрою. Як зазвичай, для переключення пристрою в режим «Ведучий» необхідно записати в регістр команд команду для формування сигналу СТАРТ. Формат адресного пакета, який передається потім, визначає, у якому з режимів буде працювати ведучий. При передачі пакета $SLA+\overline{W}$ модуль переходить у режим «Ведучий передавач», а при передачі пакета $SLA+R$ переходить у режим «Ведучий приймач».

Формування стану СТАРТ починається після запису в регістр TWCR наступного значення:



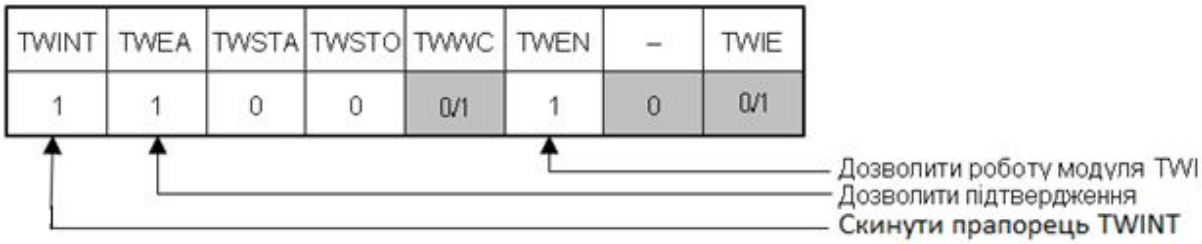
У результаті запису зазначеного значення модуль TWI почне контролювати стан шини і сформує стан СТАРТ одразу ж, як тільки вона стане вільною.

По закінченні формування стану СТАРТ встановлюється прапорець TWINT; код статусу повинен при цьому мати значення, що дорівнює $08 (таблиця 3.11). Для переключення модуля в режим «Ведучий приймач» необхідно передати шиною пакет $SLA+R$. Для цього вміст пакета завантажується в регістр TWDR, а в регістр TWCR заноситься наступне значення:



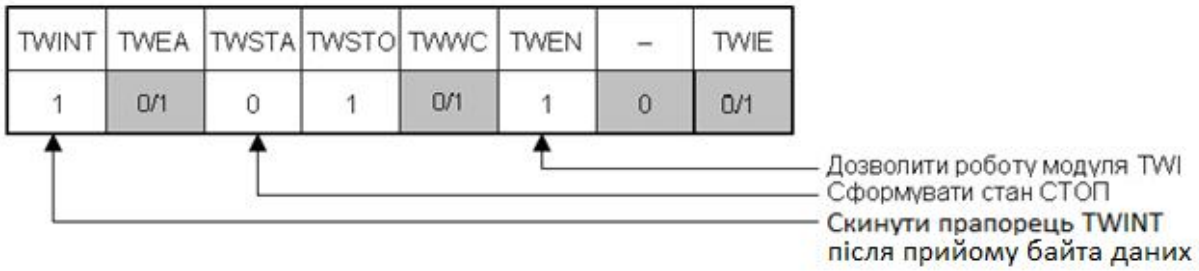
Після передачі адресного пакета і прийому біта підтвердження прапорець TWINT знову встановлюється в «1». Код статусу на цьому етапі може мати одне з наступних значень: $38, $40 або $48. Які дії необхідно почати при виявленні того або іншого коду, докладно розглянуто в таблиці 3.11.

Для скидання біта TWINT і дозволу формування біта підтвердження $\overline{ACK}$ у відповідь на прийнятий байт від веденого пристрою необхідно занести в регістр TWCR наступне значення:

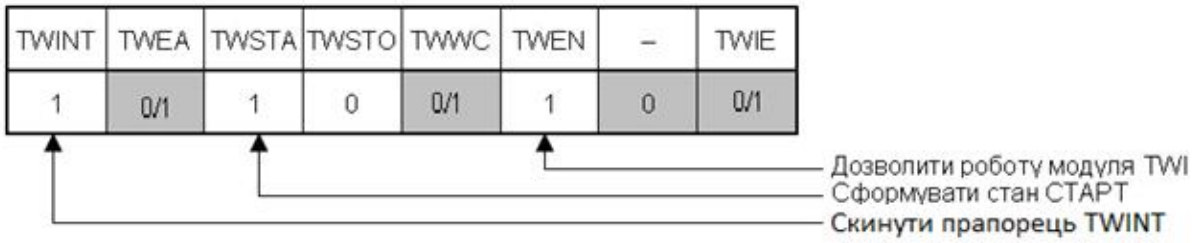


Після прийому байта код статусу має значення: $50 або $58. Дії модуля TWI у відповідь на це наведено в таблиці 3.11.

Описана процедура використовується для прийому всіх пакетів даних. Після прийому останнього байта даних ведучий повинен проінформувати про це веденого передавача, пославши сигнал непідтвердження (NACK). Потім ведучий повинен сформувати на шині стан СТОП або ПОВСТАРТ. Формування стану СТОП почнеться після запису в регістр TWCR наступного значення:



Для формування стану ПОВСТАРТ у регістр TWCR необхідно занести наступне значення:



Як уже було сказано, після формування на шині стану ПОВСТАРТ (код статусу $10) ведучий може адресувати той же або інший ведений, не формуючи стану СТОП. Інакше кажучи, використання стану ПОВСТАРТ дозволяє здійснювати зміну ведених пристроїв, а також переключатися між режимами «Ведучий передавач» і «Ведучий приймач» без втрати контролю над шиною.

Таблиця 3.11 – Коди статусу для режиму «Ведучий приймач»

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в/із TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$08	Було сформовано стан СТАРТ	Завантажити $SLA+R$	0	0	1	X	1	Буде передано $SLA+R$. Буде отримано $\overline{ACK}$ або $NACK$

Продовження таблиці 3.11

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в/із TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
\$10	Було сформовано стан ПОВСТАРТ	Завантажити $SLA+R$	0	0	1	X	1	Буде передано $SLA+R$. Буде отримано $\overline{ACK}$ або *NACK*
		Завантажити $SLA+\overline{W}$	0	0	1	X	1	Буде передано $SLA+\overline{W}$. Режим «Ведучий передавач»
\$38	Втрата пріоритету при передачі пакета адреси або даних	Немає дій	0	0	1	X	1	Пристрій звільнить шину й перейде в режим неадресованого веденого
\$40	Було передано пакет $SLA+R$ і прийнято підтвердження $\overline{ACK}$	Немає дій	0	0	1	0	1	Буде прийнято байт даних і передано непідтвердження (*NACK)*
		Немає дій	0	0	1	1	1	Буде прийнято байт даних і передано підтвердження ($\overline{ACK}$)
\$48	Було передано пакет $SLA+R$ і прийнято не підтвердження *NACK*	Немає дій	1	0	1	X	1	Буде сформовано стан ПОВСТАРТ
		Немає дій	0	1	1	X	1	Буде сформовано стан СТОП (прапорець TWSTO буде скинуто)
		Немає дій	1	1	1	X	1	Буде сформовано стан СТОП, а потім стан СТАРТ (прапорець TWSTO буде скинуто)
\$50	Було прийнято байт даних і передано підтвердження $\overline{ACK}$	Зчитати дані	0	0	1	0	1	Буде прийнято байт даних і передано *NACK*
			0	0	1	1	1	Буде прийнято байт даних і передано підтвердження $\overline{ACK}$
\$58	Було прийнято байт даних і передано не підтвердження (*NACK*)	Зчитати дані	1	0	1	X	1	Буде сформовано стан ПОВСТАРТ
		Зчитати дані	0	1	1	X	1	Буде сформовано стан СТОП (прапорець TWSTO буде скинуто)
		Зчитати дані	1	1	1	X	1	Буде сформовано стан СТОП, а потім стан СТАРТ (прапорець TWSTO буде скинуто)

У підрозділі 3.2.2 наведена схема алгоритму роботи модуля в даному режимі, а у3.2.3– програмна реалізація цього алгоритму.

На рисунку 3.21 наведені різні стани модуля TWI у режимі «Ведучий приймач».

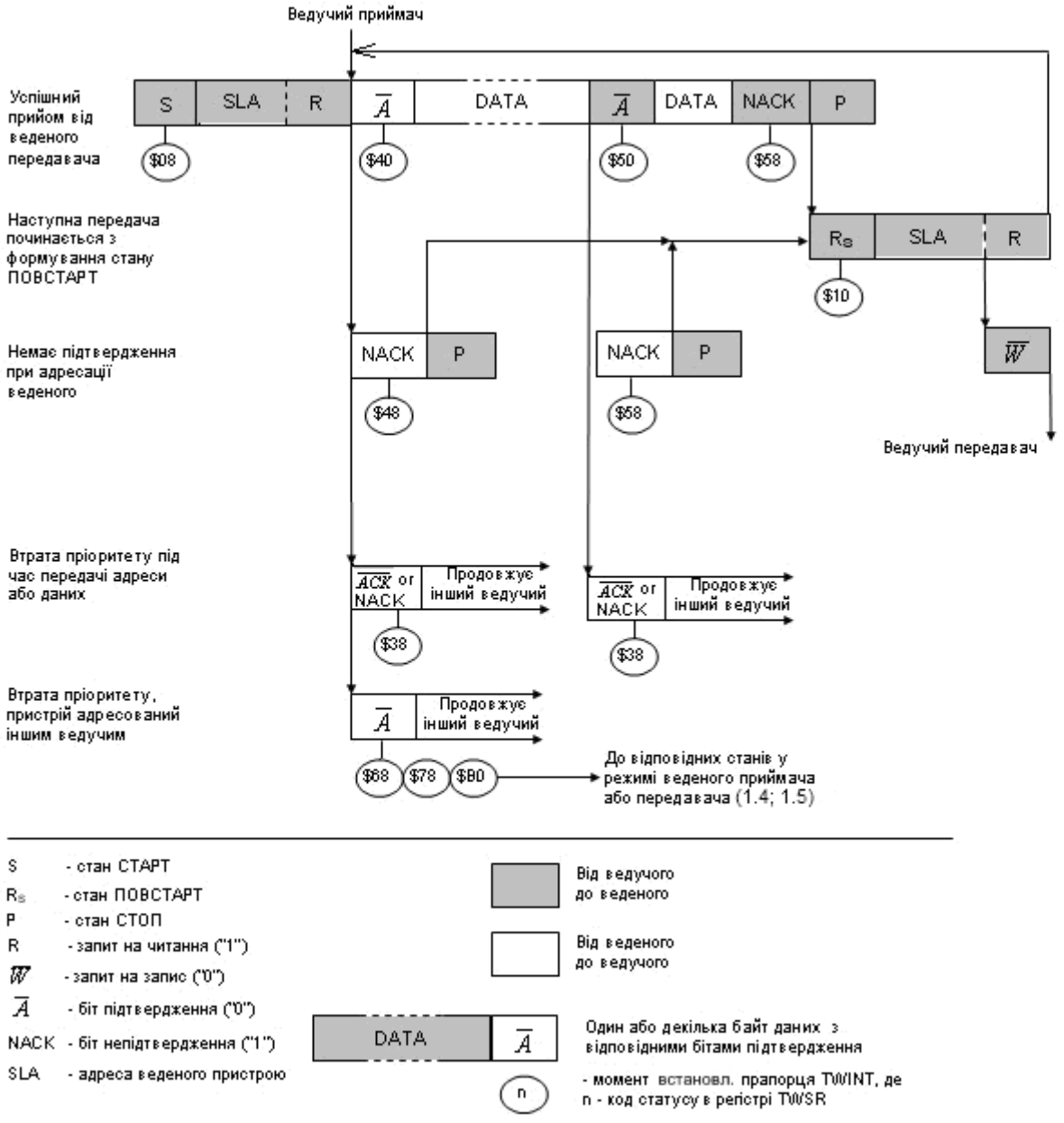


Рисунок 3.21 – Стани модуля TWI у режимі «Ведучий приймач»

3.2.1.4 Режим «Ведений приймач»

У режимі «Ведений приймач» ведений пристрій здійснює прийом даних від ведучого. Перед тим як переключити модуль у режим «Ведений приймач», слід занести у старші розряди регістра TWAR адресу пристрою і відповідно до логіки роботи програми встановити або скинути молодший розряд цього регістра (TWGCE). Потім необхідно записати в регістр TWCR наступне значення:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
1	1	0	0	0	1	0	0/1

Дозволити роботу модуля TWI
Дозволити біт підтвердження
Скинути прапорець TWINT

Після ініціалізації регістрів TWAR і TWCR модуль буде чекати на адресний пакет з адресою, зазначеною в регістрі TWAR, або загальний виклик (якщо його розпізнавання дозволено). Наступне за адресою значення біта напряму визначить режим, у який переключиться модуль. Якщо цей біт буде скинутий в «0» (запис), модуль TWI переключиться в режим «Ведений приймач». У іншому випадку модуль переключиться в режим «Ведений передавач». Пристрій також може самостійно переключитися в режим «Ведений приймач» з режиму ведучого при втраті пріоритету (див. опис кодів статусу $68 і $78 у таблиці 3.12).

Після прийому пакета $SLA + \overline{W}$ встановиться прапорець TWINT і стан обміну шиною можна буде, як зазвичай, визначити за кодом статусу. Можливі дії з боку програми, що відповідають тому або іншому коду статусу, зазначені в таблиці 3.12.

Щоб перервати потік даних, варто скинути в «0» розряд TWEA регістра TWCR (це можна зробити і під час обміну, тобто при скинутому

прапорці TWINT). У результаті після прийому наступного байта на лінію SDA буде виданий сигнал непідтвердження, що сигналізує ведучому про те, що ведений не може більше здійснювати прийом даних. Обробка адресних пакетів при скинутому розряді TWEA припиняється, але може бути відновлена в будь-який момент часу повторним встановленням цього розряду.

Якщо адресація пристрою відбудеться при перебуванні мікроконтролера у «сплячому» режимі і розряд TWEN буде при цьому встановлений, модуль TWI переведе мікроконтролер у робочий режим.

Таблиця 3.12 – Коди статусу для режиму «Ведений приймач»

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в/із TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$60	Було прийнято $SLA+\overline{W}$ з власною адресою і послано підтвердження ($\overline{ACK}$)	Немає дій	0	0	1	0	1	Буде прийнято байт даних і передано непідтвердження (*NACK*)
		Немає дій	0	0	1	1	1	Буде прийнято байт даних і передано підтвердження ($\overline{ACK}$)
$68	Втрата пріоритету в режимі ведучого під час передачі $SLA+R/\overline{W}$; було прийнято $SLA+\overline{W}$ з власною адресою і послано ($\overline{ACK}$)	Немає дій	0	0	1	0	1	Буде прийнято байт даних і передано непідтвердження (*NACK*)
		Немає дій	0	0	1	1	1	Буде прийнято байт даних і передано підтвердження ($\overline{ACK}$)
$70	Було прийнято загальний виклик і послано підтвердження ($\overline{ACK}$)	Немає дій	0	0	1	0	1	Буде прийнято байт даних і передано непідтвердження (*NACK*)
		Немає дій	0	0	1	1	1	Буде прийнято байт даних і передано підтвердження ($\overline{ACK}$)

Продовження таблиці 3.12

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в/із TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$78	Втрата пріоритету в режимі ведучого під час передачі $SLA+R/\overline{W}$; було прийнято загальний виклик і послано підтвердження ($\overline{ACK}$)	Немає дій	0	0	1	0	1	Буде прийнято байт даних і передано непідтвердження (*NACK*)
		Немає дій	0	0	1	1	1	Буде прийнято байт даних і передано підтвердження ($\overline{ACK}$)
$80	Пристрій уже адресовано: було прийнято байт даних і послано підтвердження ($\overline{ACK}$)	Зчитати дані	0	0	1	0	1	Буде прийнято байт даних і передано *NACK*
		Зчитати дані	0	0	1	1	1	Буде прийнято байт даних і передано ($\overline{ACK}$)
$88	Пристрій уже адресовано: було прийнято байт даних і послано непідтвердження (*NACK*)	Зчитати дані	0	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання викликів відключено
		Зчитати дані	0	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»
		Зчитати дані	1	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів заборонено; після звільнення шини буде сформовано стан СТАРТ
		Зчитати дані	1	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»; після звільнення шини буде сформ. стан СТАРТ

Продовження таблиці 3.12

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в/із TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$90	Пристрій уже адресовано (загальний виклик): було прийнято байт даних і послано підтвердження ($\overline{ACK}$)	Зчитати дані	0	0	1	0	1	Буде прийнято байт даних і передано *NACK*
		Зчитати дані	0	0	1	1	1	Буде прийнято байт даних і передано $\overline{ACK}$
$98	Пристрій уже адресовано (загальний виклик): було прийнято байт даних і послано непідтвердження (*NACK*)	Зчитати дані	0	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів відключено
		Зчитати дані	0	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»
		Зчитати дані	1	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів заборонено; після звільнення шини буде сформовано стан СТАРТ
		Зчитати дані	1	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»; після звільнення шини буде сформовано стан СТАРТ

Продовження таблиці 3.12

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в/із TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$A0	Було виявлено стан СТОП або ПОВСТАРТ у той час, коли пристрій було адресовано в якості веденого	Немає дій	0	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів заборонено
		Немає дій	0	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»
		Немає дій	1	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів заборонено; після звільнення шини буде сформовано стан СТАРТ
		Немає дій	1	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»; після звільнення шини буде сформовано стан СТАРТ

У підрозділі 3.2.2 наведена схема алгоритму роботи модуля в даному режимі, а у 3.2.3– програмна реалізація цього алгоритму.

На рисунку 3.22 наведено різні стани модуля TWI у режимі “Ведений приймач”.

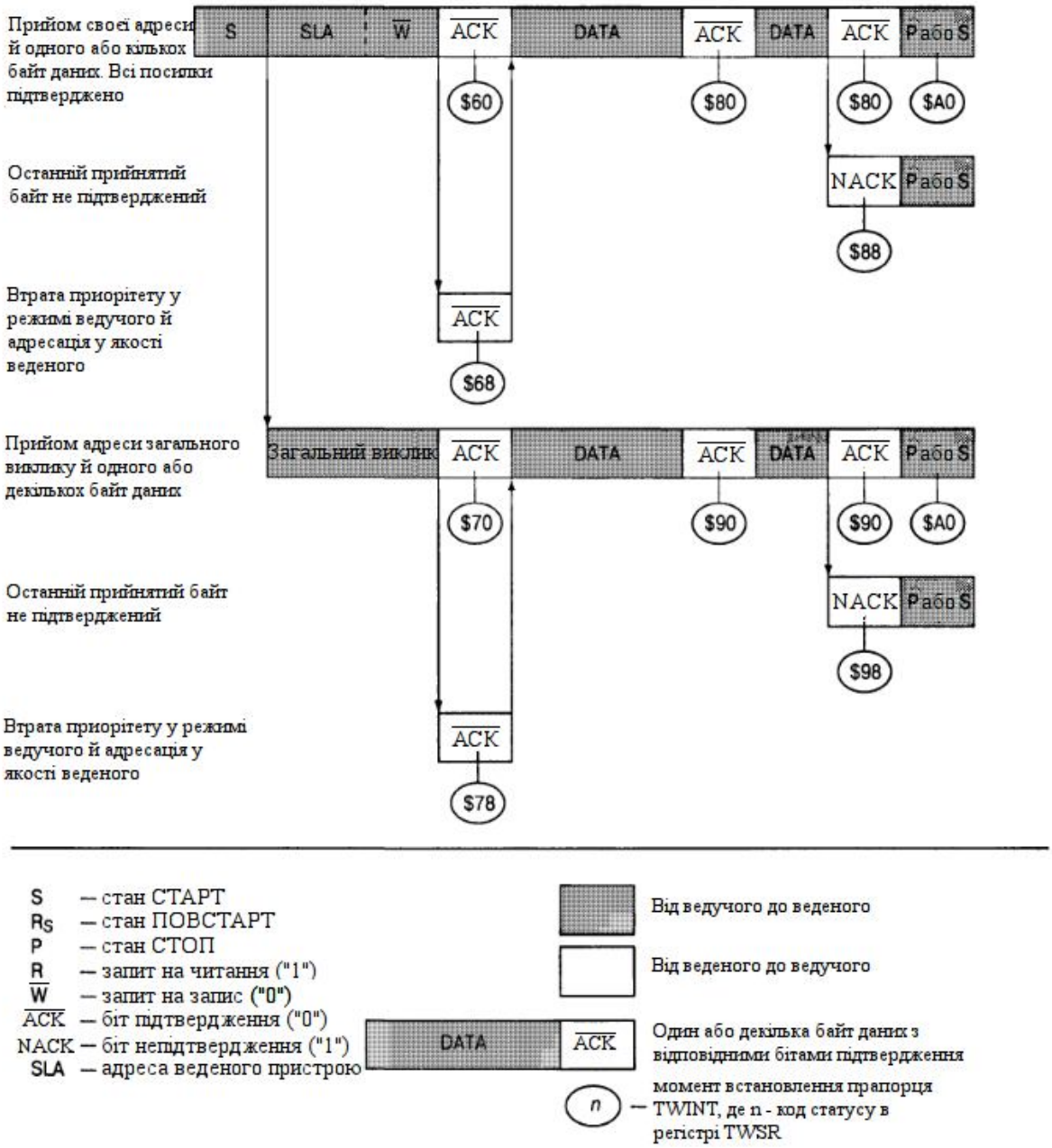
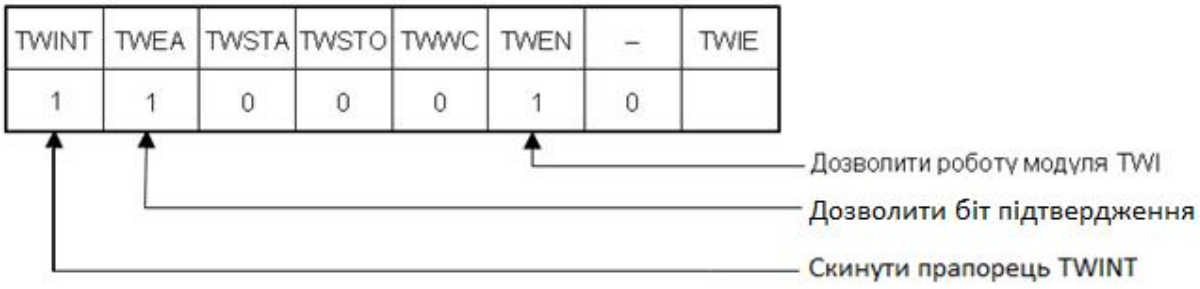


Рисунок 3.22 – Стани модуля TWI у режимі “Ведений приймач”

3.2.1.5 Режим «Ведений передавач»

У режимі «Ведений передавач» ведений пристрій здійснює передачу даних ведучому, який у цьому випадку є приймачем. Перед тим як переключити модуль у цей режим, слід занести у старші розряди регістра TWAR адресу пристрою і відповідно до логіки роботи програми

встановити або скинути молодший розряд регістра (TWGCE). Потім необхідно записати в регістр TWCR наступне значення:



Після ініціалізації регістрів TWAR і TWCR модуль буде чекати на адресний пакет з адресою, зазначеною в регістрі TWAR. Наступне за адресою значення біта напряму визначить режим, у який переключиться модуль. Якщо цей біт буде встановлено в «1» (читання), модуль TWI переключиться в режим «Ведений передавач». У іншому випадку модуль переключиться в режим «Ведений приймач». Варто пам'ятати, що пристрій також може автоматично переключитися в режим «Ведений передавач» з режиму ведучого при втраті пріоритету (див. опис коду статусу $B0 у таблиці 3.13).

Після прийому пакета $SLA + R$ встановиться прапорець TWINT, і стан обміну можна буде, як зазвичай, визначити за кодом статусу. Можливі дії з боку програми, які відповідають тому або іншому коду статусу, зазначені в таблиці 3.13.

При передачі останнього байта даних необхідно скинути в «0» розряд TWEA. Після цього модуль перейде в стан з кодом статусу $C0 або $C8 залежно від того, який сигнал ($\overline{ACK}$ або *NACK*) передасть ведучий у відповідь. У стан з кодом статусу $C8 модуль TWI перейде у випадку, якщо ведучий забажав додаткові дані, передавши підтвердження ($\overline{ACK}$), незважаючи на те, що ведений передавач передав останній байт і очікував сигналу *NACK*. Але реакція передавача на стан $C8 подібна реакції на $C0.

Після переходу в кожний із двох останніх станів модуль TWI буде ігнорувати звернення до нього ведучого. Відповідно, якщо ведучий буде продовжувати обмін шиною, він буде постійно приймати значення «1». Обробка адресних пакетів при скинутому розряді TWEA також припиняється, але може бути відновлена в будь-який момент часу повторним встановленням цього розряду.

Таблиця 3.13 – Коди статусу для режиму «Ведений передавач»

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$A8	Було прийнято $SLA + R$ з власною адресою і послано підтвердження ($\overline{ACK}$)	Завантажити дані	0	0	1	0	1	Було передано останній байт даних; повинно бути отримано непід-твердження (NACK)
		Завантажити дані	0	0	1	1	1	Буде передано черговий байт даних; повинно бути отримано підтвердження ($\overline{ACK}$)
$B0	Втрата пріоритету в режимі ведучого під час передачі $SLA+R/\overline{W}$; був прийнято $SLA + R$ із власною адресою і послано підтвердження ($\overline{ACK}$)	Завантажити дані	0	0	1	0	1	Буде передано останній байт даних; повинно бути отримано непідтвердження (*NACK*)
		Завантажити дані	0	0	1	1	1	Буде передано черговий байт даних; повинно бути отримано підтвердження ($\overline{ACK}$)

Продовження таблиці 3.13

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$B8	Було передано байт даних і отримано підтвердження ($\overline{ACK}$)	Завантажити дані	0	0	1	0	1	Буде передано останній байт даних; повинно бути отримано непідтвердження (*NACK*)
		Завантажити дані	0	0	1	1	1	Буде передано черговий байт даних; повинно бути отримано підтвердження ($\overline{ACK}$)
$C0	Було передано останній байт даних і отримано непідтвердження (*NACK*)	Немає дій	0	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів заборонено
		Немає дій	0	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»
		Немає дій	1	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів заборонено; після звільнення шини буде сформовано стан СТАРТ
		Немає дій	1	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»; після звільнення шини буде сформовано стан СТАРТ

Продовження таблиці 3.13

Код статусу	Стан шини модуля TWI	Дії програми						Наступна дія, що виконується модулем TWI
		в TWDR	в регістр TWCR					
			TWSTA	TWSTO	TWINT	TWEA	TWEN	
$C8	Було передано останній байт даних і отримано підтвердження ($\overline{ACK}$)	Немає дій	0	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів заборонено
		Немає дій	0	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»
		Немає дій	1	0	1	0	1	Переключиться в режим неадресованого веденого; розпізнавання будь-яких викликів заборонено; після звільнення шини буде сформовано стан СТАРТ
		Немає дій	1	0	1	1	1	Переключиться в режим неадресованого веденого; дозволено розпізнавання SLA із власною адресою; дозволено розпізнавання загальних викликів, якщо TWGCE = «1»; після звільнення шини буде сформовано стан СТАРТ

У підрозділі 3.2.2 наведена схема алгоритму роботи модуля в даному режимі, а у 3.2.3– програмна реалізація цього алгоритму.

На рисунку 3.23 наведені різні стани модуля TWI у режимі «Ведений передавач».

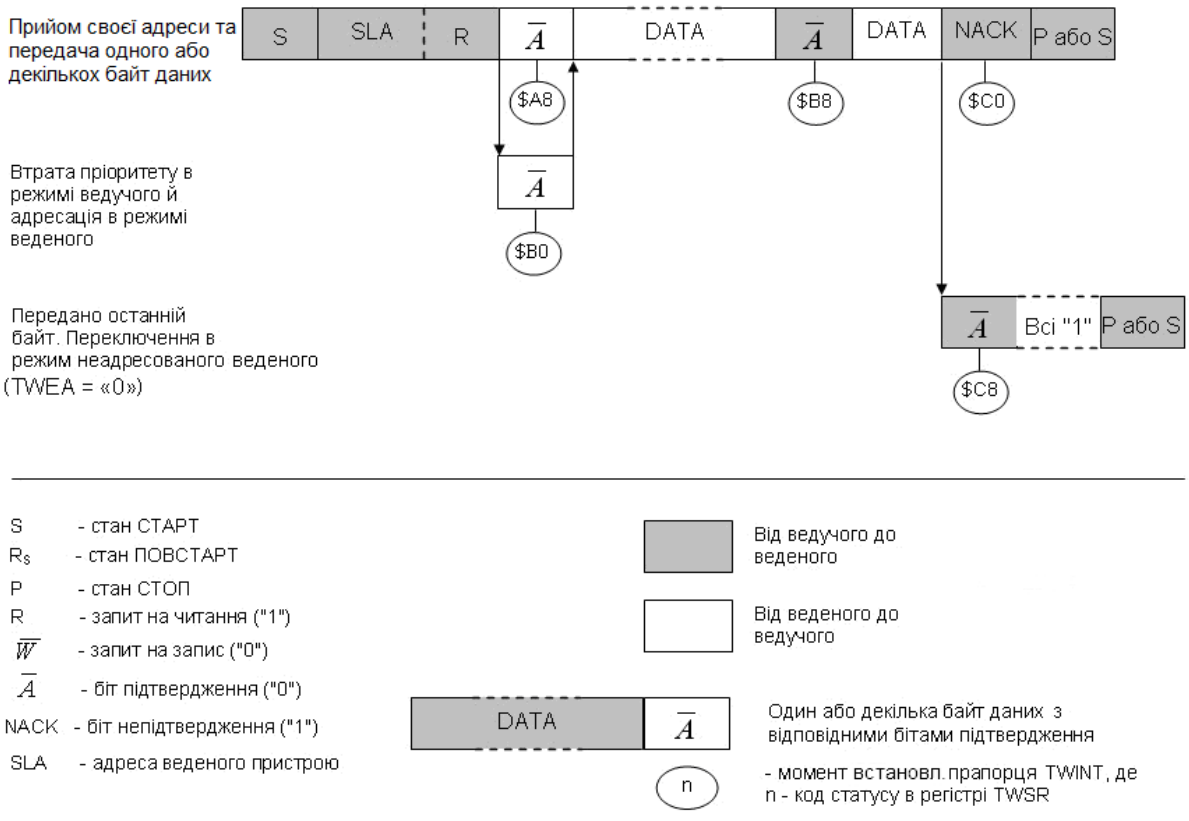


Рисунок 3.23 – Стани модуля TWI у режимі «Ведений передавач»

3.2.1.6 Комбінування різних режимів

На практиці для виконання чергової операції шиною TWI кожному пристрою доводиться використовувати кілька режимів, переключаючись між ними при необхідності. Як приклад розглянемо операцію читання даних із зовнішньої пам’яті даних типу EEPROM.

Всі операції такого роду можна розбити на чотири етапи:

– ініціювання обміну;

– передача адреси, за якою потрібно прочитати дані;

– виконання читання;

– завершення передачі.

Зі сказаного зрозуміло, що при виконанні операції відбувається передача інформації як від ведучого до веденого, так і навпаки. Після

ініціювання обміну шиною ведучий повинен перебувати в режимі «Ведучий передавач», щоб повідомити веденому адресу комірки пам’яті, за якою він має намір здійснити читання. Для виконання наступного етапу операції (читання даних) ведучий повинен переключитися в режим «Ведучий приймач». При цьому він повинен зберігати контроль над шиною під час виконання всіх етапів операції. Для цього використовується стан ПОВСТАРТ. У розглянутому випадку ведучий формує цей стан між передачею адреси і прийомом даних, як показано на рисунку 3.24.

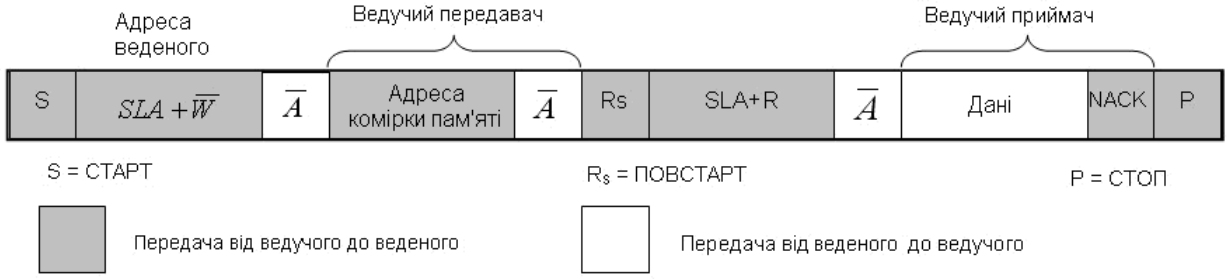


Рисунок 3.24 – Приклад звернення до зовнішньої пам’яті даних типу EEPROM

3.2.1.7 Арбітраж

У випадку присутності на шині декількох ведучих можлива ситуація, при якій декілька ведучих одночасно почнуть процес обміну. Для таких випадків специфікацією TWI передбачений процес розподілу пріоритетів (арбітраж), у результаті виконання якого на шині залишається тільки один ведучий пристрій. Принципи цього процесу були розглянуті в 3.1.2.2, однак його виконання може розвиватися за різними сценаріями (рисунок 3.25):

1) Два або більше ведучих здійснюють однотипний обмін з одним і тим самим веденим. У цьому випадку ніхто з них не зможе розпізнати конфлікт на шині.

2) Два або більше ведучих звертаються до одного й того ж самого веденого з різними даними або різними типами обміну (читання/запис). У цьому випадку розподіл пріоритетів відбудеться під час передачі бітів даних або біта напряму. Ведучий, що втратив пріоритет, може або переключитися в режим неадресованого веденого, або дочекатися звільнення шини і сформувати стан СТАРТ для її захоплення.

3) Два або більше ведучих звертаються до різних ведених. У цьому випадку розподіл пріоритетів почнеться під час передачі бітів адресного пакета. Ведучий, що втратив пріоритет, переключиться в режим веденого для перевірки, чи не був він адресований іншим ведучим. Якщо колишній ведучий був адресований, він переключиться в режим «Ведений передавач» або «Ведений приймач» залежно від значення біта напряму. Якщо ж він не був адресований, він переключиться в режим неадресованого веденого, або дочекається звільнення шини і сформує новий стан СТАРТ.

3.2.1.8 Параметри інтерфейсу

Вимоги, які пред’являються до пристроїв, що підключаються до шини TWI наведені у [1,2]

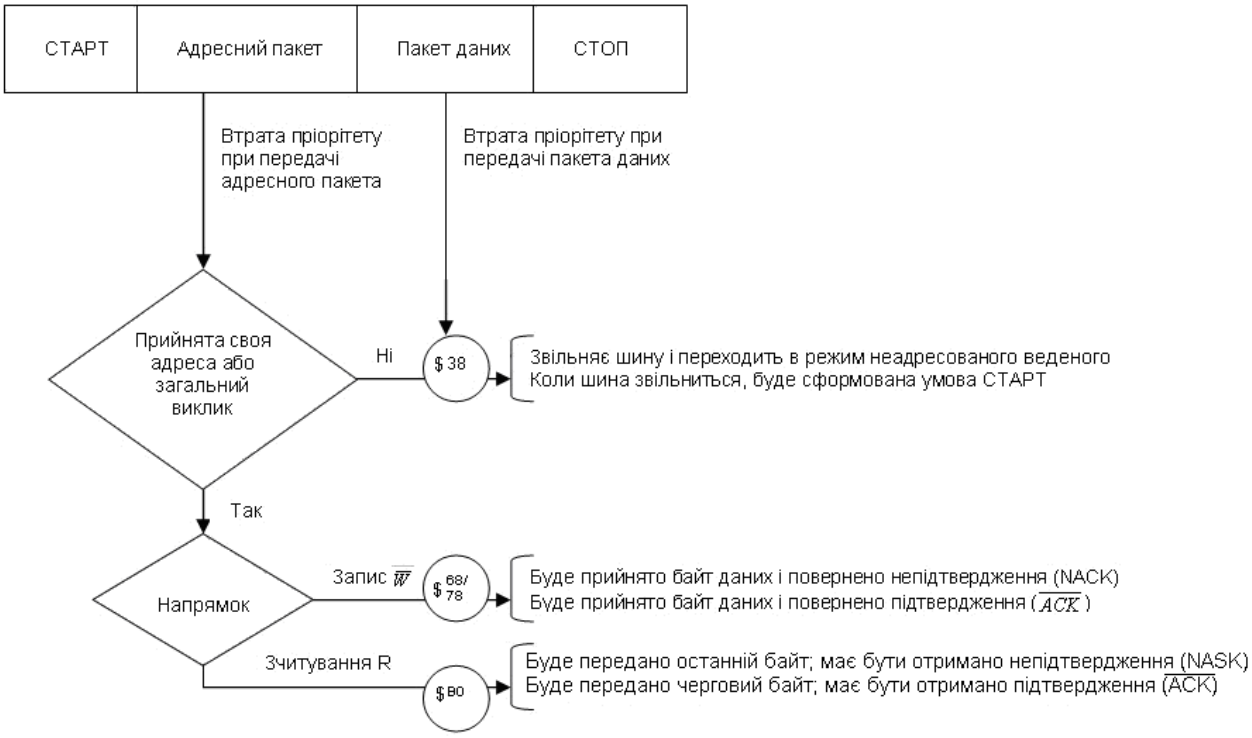


Рисунок 3.25 – Стани модуля TWI під час арбітражу

3.2.2 СХЕМИ АЛГОРИТМІВ РОБОТИ МОДУЛЯ TWI

Схеми алгоритмів роботи модуля TWI у розглянутих вище режимах наведені на рисунках 3.26...3.29.

У наведених нижче алгоритмах використані наступні скорочення:

- $MT\_SLA\_\overline{ACK}$: означає, що ведучим було передано пакет $SLA+\overline{W}$ і отримано підтвердження ($\overline{ACK}$);
- $MT\_DATA\_\overline{ACK}$: означає, що ведучим було передано пакет даних та отримано підтвердження ($\overline{ACK}$);
- $MR\_SLA\_\overline{ACK}$: означає, що ведучим було передано пакет $SLA+R$ і отримано підтвердження ($\overline{ACK}$);
- $MR\_DATA\_\overline{ACK}$: означає, що ведучим було прийнято байт даних і передано підтвердження ($\overline{ACK}$);

- MR_DATA_NACK: означає, що ведучим було прийнято останній байт даних і передано непідтвердження ($NACK$);
- $SR\_SLA\_\overline{ACK}$: означає, що веденим було прийнято пакет $SLA+\overline{W}$ з власною адресою і передано підтвердження ($\overline{ACK}$);
- $SR\_DATA\_\overline{ACK}$: означає, що веденим було прийнято байт даних і передано підтвердження ($\overline{ACK}$);
- SR_DATA_NACK: означає, що веденим було прийнято останній байт та передано непідтвердження ($NACK$);
- $ST\_SLA\_\overline{ACK}$: означає, що веденим було прийнято пакет $SLA+R$ з власною адресою і передано підтвердження ($\overline{ACK}$);
- $ST\_DATA\_\overline{ACK}$: означає, що веденим було передано байт даних і отримано підтвердження ($\overline{ACK}$);
- ST_DATA_NACK: означає, що веденим було передано останній байт і отримано непідтвердження ($NACK$).

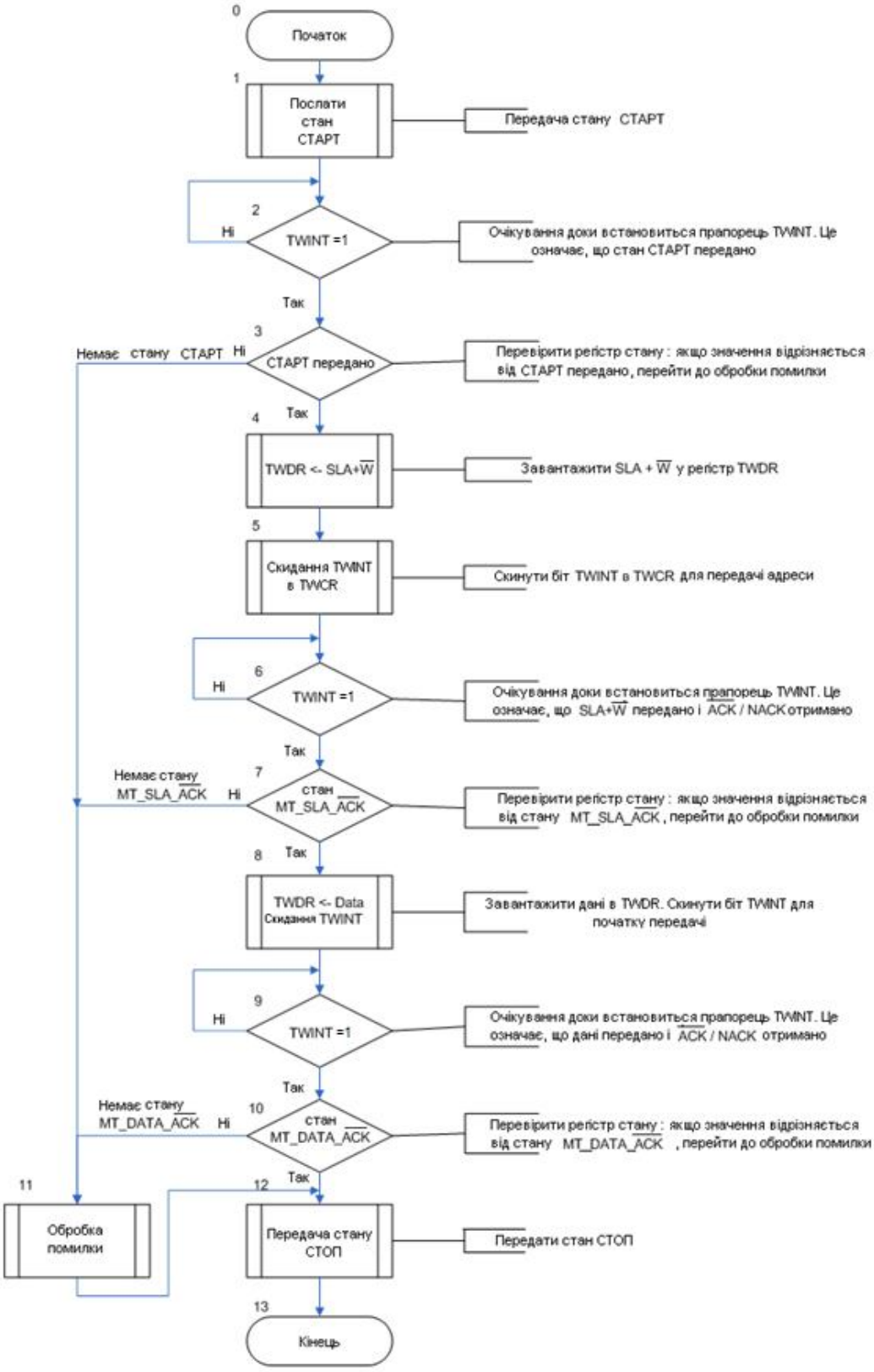


Рисунок 3.26 – Схема алгоритму роботи модуля в режимі «Ведучий передавач»

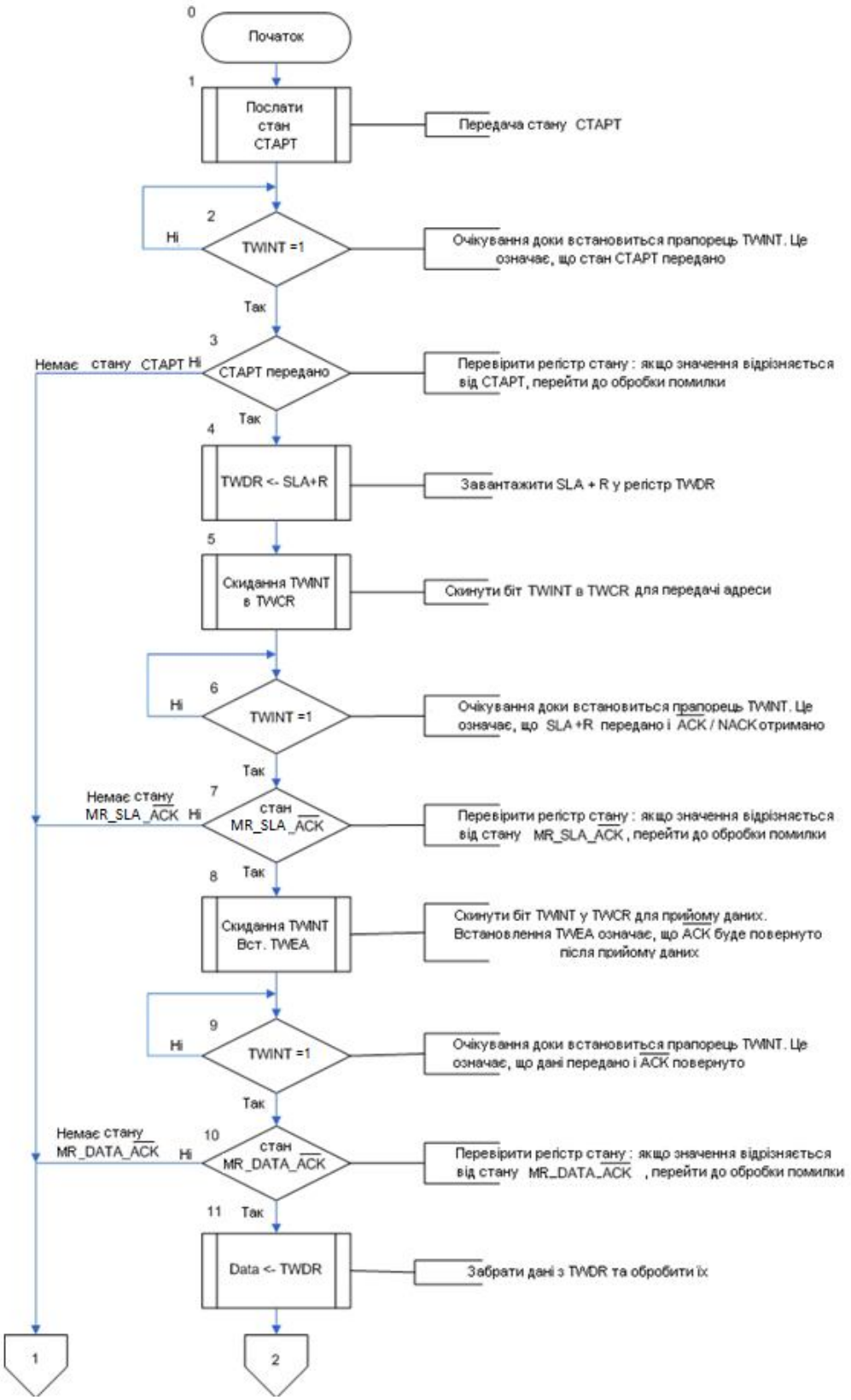


Рисунок 3.27 – Схема алгоритму роботи модуля у режимі «Ведучий приймач».

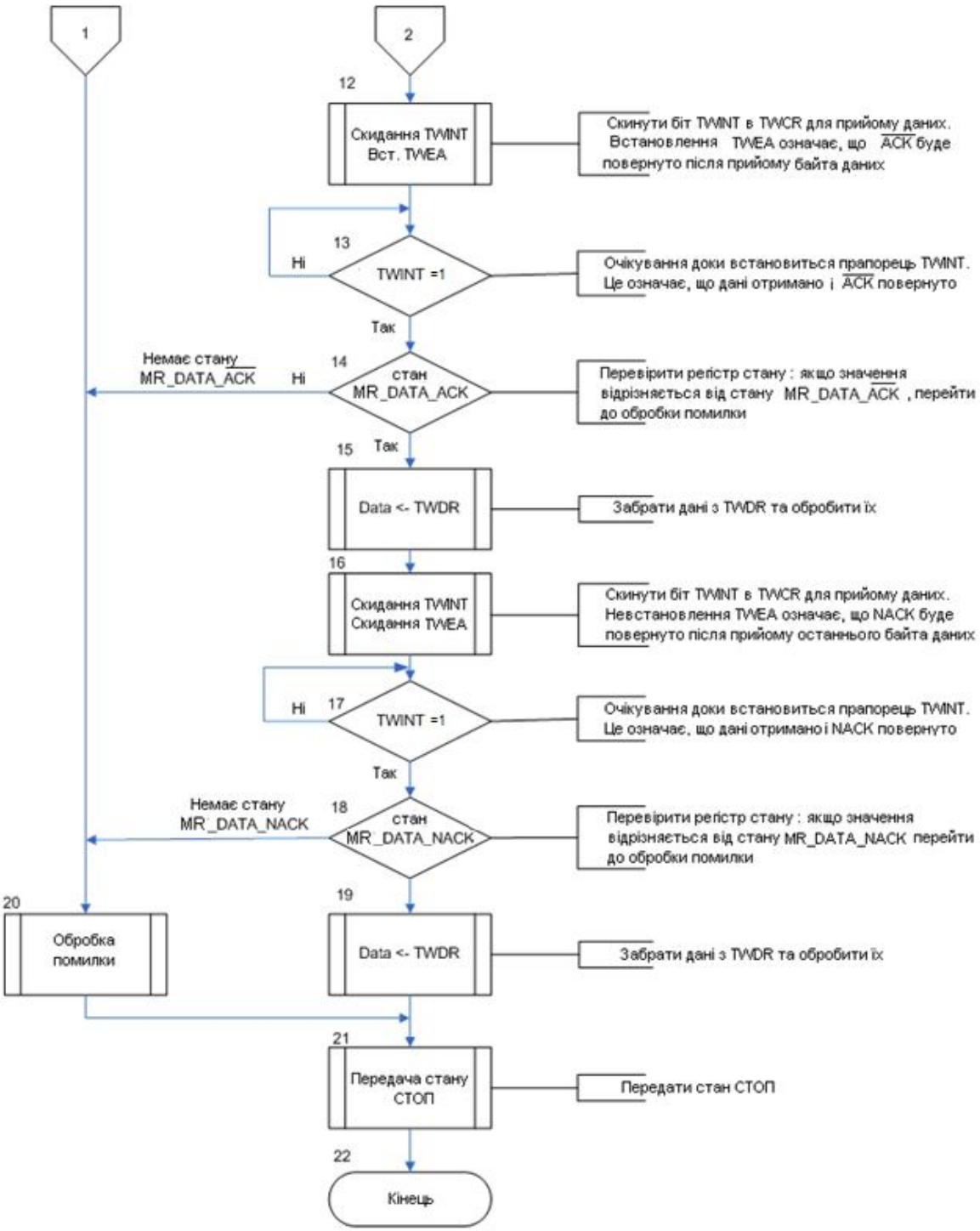


Рисунок 3.27 – Схема алгоритму роботи модуля у режимі «Ведучий приймач» (продовження)

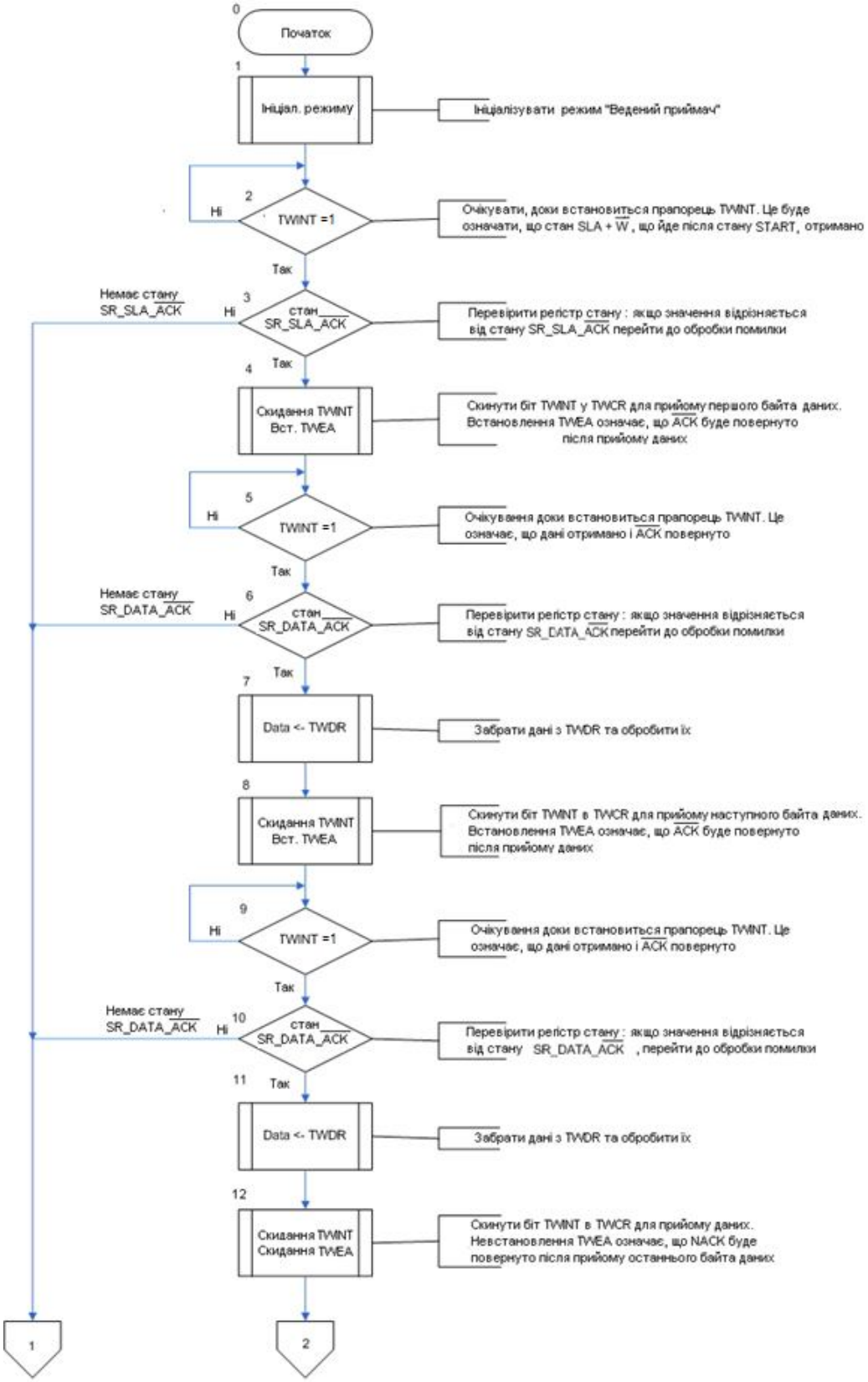


Рисунок 3.28 – Схема алгоритму роботи модуля у режимі «Ведений приймач».

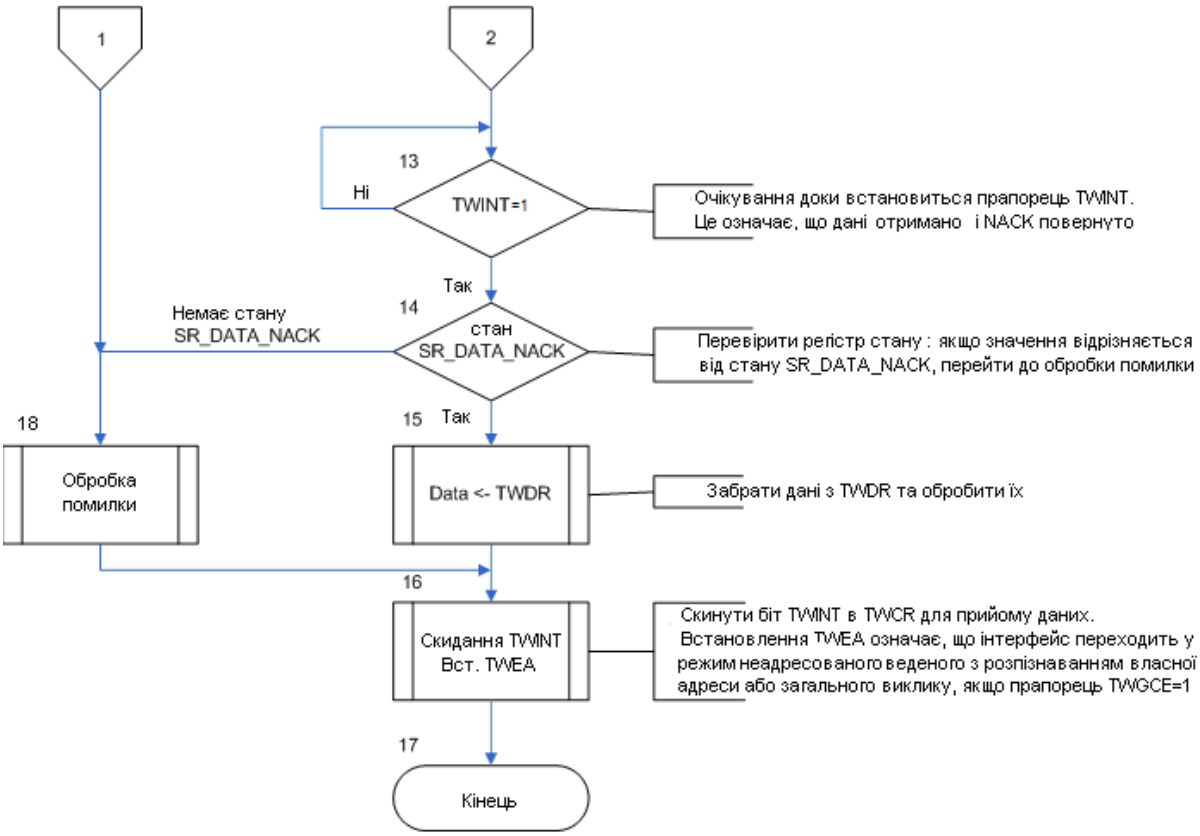


Рисунок 3.28 – Схема алгоритму роботи модуля у режимі «Ведений приймач» (продовження)

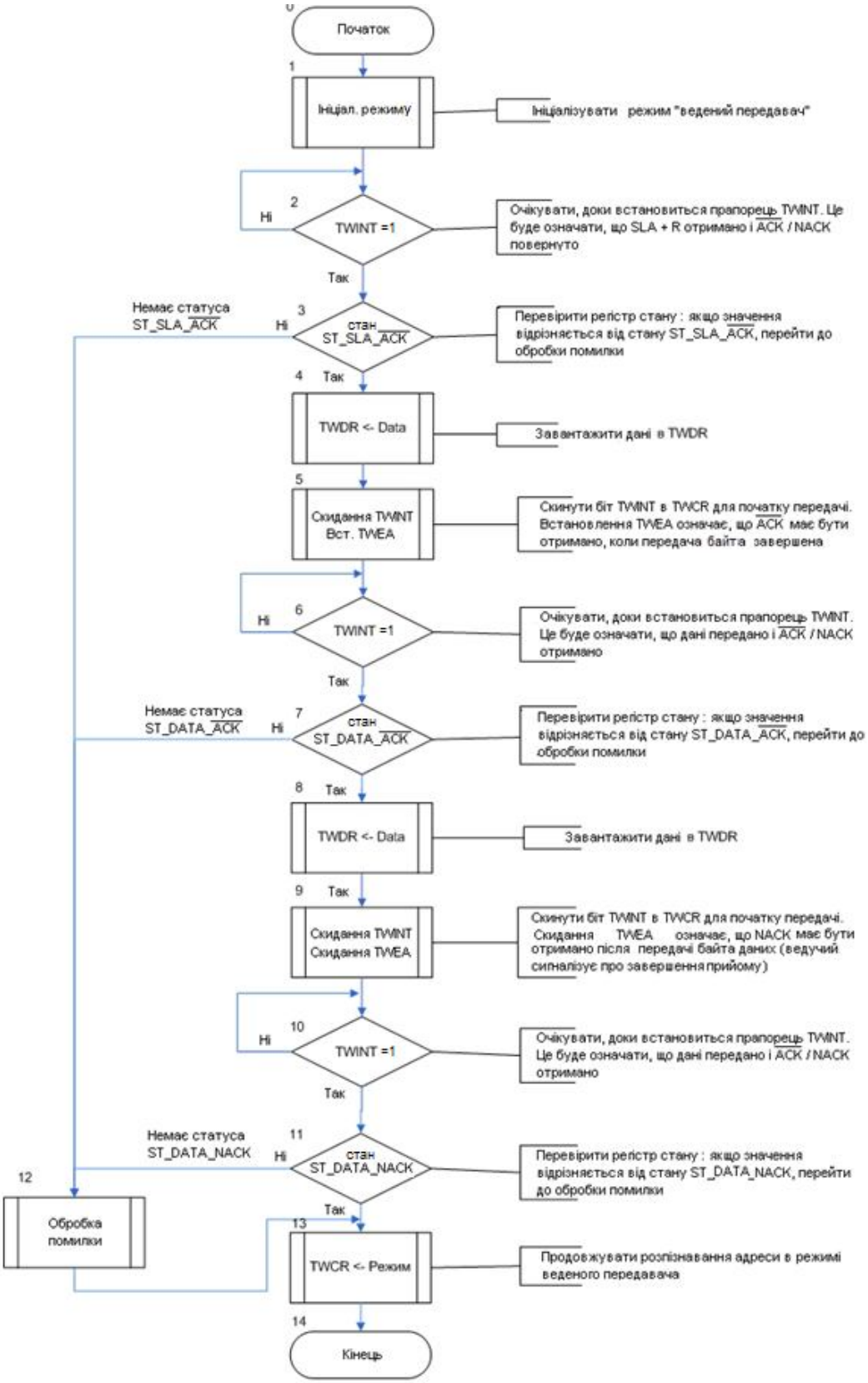


Рисунок 3.29 – Схема алгоритму роботи модуля в режимі «Ведений передавач».

3.2.3 Програмна реалізація алгоритмів роботи модуля twi

Нижче представлені робочі програми, які реалізують наведені вище алгоритми мовою асемблер.

3.2.3.1 Програма роботи у режимі «Ведучий передавач»

```
; Адреса веденого 0x64. У лістингу не приводиться обробка
; помилок у підпрограмі ERROR.
; Додатково у лістинг повинні бути включені заголовочні файли
; пристрою з оголошенням змінних, що відповідають
; регістрам TWDR, TWSR, TWCR, TWAR, TWBR та їх окремим бітам,
; та ініціалізація регістрів TWAR, TWCR, TWBR

        ldi r16,
            (1<<TWSTA)|
            (1<<TWEN)|(1<<TWINT)
        out TWCR, r16               ; послати стан START (блок 1)
wait1:  in r16,TWCR                 ; очікувати доки встановиться
        sbrs r16,TWINT              ; прапорець TWINT. Це буде
        rjmp wait1                  ; означати, що стан START
                                    ; передано (блок 2)
        in r16, TWSR                ; перевірити регістр стану.
        cpi r16, $08                ; Якщо значення
        brne ERROR                  ; відрізняється від стану
                                    ; START, перейти до обробки
                                    ; помилки (блок 3)
        ldi r16, 0xc8               ; завантажити SLA+W у регістр
        out TWDR, r16               ; TWDR (блок 4)
        ldi r16,                    ; скинути біт TWINT в TWCR
            (1<<TWINT)|(1<<TWEN)    ; для передачі адреси(блок5)
        out TWCR, r16
wait2:  in r16, TWCR                ; очікувати доки встановиться
        sbrs r16, TWINT             ; прапорець TWINT.
        rjmp wait2                  ; Це буде означати, що
                                    ; SLA+W передано і ACK/NACK
                                    ; отримано (блок 6)
        in r16, TWSR                ; перевірити регістр стану.
        cpi r16, $18                ; Якщо значення відрізняється
        brne ERROR                  ; від MT_SLA_ACK , перейти
                                    ; до обробки помилки;(блок 7)
        ldi r16, 0x33               ; завантажити дані (0x33) в
        out TWDR, r16               ; TWDR (блок 8)
        ldi r16,                    ; скинути біт TWINT для
            (1<<TWINT)|(1<<TWEN)    ; початку передачі (блок 8)
```

```
        out TWCR, r16
wait3:  in r16, TWCR                ; очікувати доки встановиться
        sbrs r16, TWINT             ; прапорець TWINT. Це
        rjmp wait3                  ; означає, що дані передано
                                    ; ACK / NACK отримано(блок 9)
        in r16, TWSR                ; перевірити регістр стану.
        cpi r16, $28                ; Якщо значення відрізняється
        brne ERROR                  ; від MT_DATA_ACK , перейти
                                    ; до обробки помилки(блок 10)
; передача наступних даних....
        ldi r16, (1<<TWINT) |           ; передача стану STOP (блок 12)
            (1<<TWSTO) |
            (1<<TWEN)
        out TWCR, r16
;підпрограма обробки помилки
        ERROR:
            …
            …
            …
```

3.2.3.2 Програма роботи у режимі «Ведучий приймач»

```
; Додатково у лістинг повинні бути включені заголовочні файли
; пристрою з оголошенням змінних, що відповідають регістрам
; TWDR, TWSR, TWCR, TWAR, TWBR та їх окремим бітам,
; та ініціалізація регістрів TWAR, TWCR, TWBR

        ldi r16, (1<<TWSTA)|
           (1<<TWEN)|(1<<TWINT)
         out TWCR, r16              ; послати стан START (блок 1)
wait1:   in r16, TWCR               ; очікувати доки встановиться
         sbrs r16, TWINT            ; прапорець TWINT. Це буде
         rjmp wait1                 ; означати, що стан
                                    ; START передано (блок 2)
         in r16, TWSR               ; перевірити регістр стану.
         cpi r16, $08               ; Якщо значення
         brne ERROR                 ; відрізняється від стану
                                    ; START, перейти до обробки
                                    ; помилки (блок 3)
         ldi r16, 0xc9              ; завантажити SLA+R у регістр
         out TWDR, r16              ; TWDR (блок 4)
         ldi r16,(1<<TWINT) |       ; скинути біт TWINT в TWCR
             (1<<TWEN)              ; для передачі SLA+R(блок 5)
         out TWCR, r16
wait2:   in r16, TWCR               ; очікувати доки встановиться
         sbrs r16, TWINT            ; прапорець TWINT. Це означає,
         rjmp wait2                 ; що SLA+R передано та
                                    ; ACK / NACK отримано (блок 6)
```

```
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16, $40            ; Якщо значення відрізняється
        brne ERROR              ; від MR_SLA_ACK , перейти до
                                ; обробки помилки (блок 7)
        ldi r16, (1<<TWINT) |
            (1<<TWEA) |
            (1<<TWEN)
        out TWCR, r16           ; скинути біт TWINT в TWCR
                                ; для прийому даних.
                                ; Встановлення TWEA означає,
                                ; що ACK буде повернуто після
                                ; отримання даних (блок 8)
wait7:  in r16,TWCR             ; очікувати встановлення
        sbrs r16, TWINT         ; прапорця TWINT. Це означає,
        rjmp wait7              ; що дані отримано і
                                ; ACK повернуто (блок 9)
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16, $50            ; Якщо значення відрізняється
        brne ERROR              ; від MR_DATA_ACK , перейти
                                ; до обробки помилки (блок 10)
        in r16, TWDR            ; забрати дані з TWDR (блок 11)

        ;⋮                      ; обробити дані

        ldi r16, (1<<TWINT) |
            (1<<TWEA) |
            (1<<TWEN)
        out TWCR, r16           ; скинути біт TWINT в TWCR
                                ; для прийому даних. (блок 12)
; отримання інших байтів
…
; отримання передостаннього байта
wait8:  in r16,TWCR             ; очікувати встановлення
        sbrs r16, TWINT         ; прапорця TWINT. Це означає,
        rjmp wait8              ; що дані отримано та ACK
                                ; повернуто (блок 13)
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16,$50             ; Якщо значення відрізняється
        brne ERROR              ; від MR_DATA_ACK , перейти до
                                ; обробки помилки (блок 14)
        in r16, TWDR            ; забрати дані з TWDR (блок 15)

        ;…                      ; обробити дані

        ldi r16,
            ~(1<<TWINT) |
            (1<<TWEN)
        out TWCR, r16           ; скинути біт TWINT в TWCR для
                                ; прийому даних. Не
```

```
                                ; встановлення TWEA означає,
                                ; що NACK буде повернуто після
                                ; отримання останнього байта
                                ; даних (блок 16)
; отримання останнього байта даних.
; Сигнал веденому — NACK
wait9:  in r16,TWCR             ; очікувати встановлення
        sbrs r16, TWINT         ; прапорця TWINT. Це означає,
        rjmp wait9              ; що дані отримано та
                                ; NACK повернуто (блок 17)
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16, $58            : Якщо значення відрізняється
        brne ERROR              ; від MR_DATA_NACK, перейти до
                                ; обробки помилки (блок 18)
        in r16, TWDR            ; забрати дані з TWDR (блок 19)

        ;…                      ; обробити дані

        ldi r16, (1<<TWINT) |   ; передача стану STOP (блок 21)
            (1<<TWSTO) |
            (1<<TWEN)
        out TWCR, r16
;підпрограма обробки помилки
        ERROR:
            …
            …
            …
```

3.2.3.3 Програма роботи у режимі «Ведений приймач»

```
; Додатково у лістинг повинні бути включені заголовочні файли
; пристрою з оголошенням змінних, що відповідають
; регістрам TWDR, TWSR, TWCR, TWAR, TWBR та їх окремим бітам,
; та ініціалізація регістрів TWAR, TWCR, TWBR

         ldi r16, (1<<TWINT) |
             (1<<TWEA) |
             (1<<TWEN)
         out TWCR, r16          ; ініціалізувати режим
                                ; «Ведений приймач» (блок 1)
wait10:  in r16,TWCR            ; очікувати доки встановиться
         sbrs r16,TWINT         ; прапорець TWINT. Це буде
         rjmp wait10            ; означати, що стан SLA+W̄ ,
                                ; що йде за станом START,
                                ; отримано. (блок 2)
         in r16, TWSR           ; перевірити регістр стану.
         cpi r16, $60           ; Якщо значення відрізняється
         brne ERROR             ; від стану SR_SLA_ACK̄ ,
```

```
                                ; перейти до обробки
                                ; помилки (блок 3)
        ldi r16, (1<<TWINT) |
            (1<<TWEA) |
            (1<<TWEN)
        out TWCR, r16           ; скинути біт TWINT в TWCR для
                                ; прийому першого байта даних.
                                ; Встановлення TWEA означає,
                                ; що ACK буде повернуто після
                                ; прийняття першого байта
                                ; даних (блок 4)
wait11: in r16,TWCR             ; очікувати встановлення
        sbrs r16, TWINT         ; прапорця TWINT.
        rjmp wait11             ; Це означає, що дані отримано
                                ; та ACK повернуто (блок 5)
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16, $80            ; Якщо значення відрізняється
        brne ERROR              ; від SR_DATA_ACK, перейти до
        in r16, TWDR            ; обробки помилки;(блок 6)
                                ; забрати дані з TWDR

        ;…                      ; обробити дані (блок 7)

        ldi r16, (1<<TWINT) |
            (1<<TWEA) |
            (1<<TWEN)
        out TWCR, r16           ; скинути біт TWINT в TWCR для
                                ; прийому байтів даних.
                                ; Встановлення TWEA означає,
                                ; що ACK буде повернуто після
                                ; прийняття наступних байтів
                                ; даних (блок 8)
wait12: in r16,TWCR             ; очікувати встановлення прапорця
        sbrs r16, TWINT         ; TWINT. Це означає, що дані
        rjmp wait12             ; отримано та ACK
                                ; повернуто (блок 9)
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16, $80            ; Якщо значення відрізняється
        brne ERROR              ; від SR_DATA_ACK, перейти до
                                ; обробки помилки (блок 10)
        in r16, TWDR            ; забрати дані з TWDR

        ;…                      ; обробити дані (блок 11)

        ldi r16,                ; скинути біт TWINT в TWCR для
            (1<<TWINT)|         ; прийому даних. Невстановлення
            (1<<TWEN)           ; TWEA означає, що NACK буде
        out TWCR, r16           ; повернуто після отримання
                                ; останнього байта даних
```

```
                              ; (блок 12)
wait13:  in r16,TWCR          ; очікувати встановлення прапорця
         sbrs r16, TWINT      ; TWINT. Це означає, що дані
         rjmp wait13          ; отримано та NACK повернуто
                              ; (блок 13)
         in r16, TWSR         ; перевірити регістр стану. Якщо
         cpi r16, $88         ; значення відрізняється від
         brne ERROR           ; SR_DATA_NACK, перейти до
                              ; обробки помилки;(блок 14)
         in r16, TWDR         ; забрати дані з TWDR (блок 15)

         ;…                   ; обробити дані

         ldi r16,
             (1<<TWINT) |
             (1<<TWEA) |
             (1<<TWEN)
         out TWCR, r16        ; скинути біт TWINT в TWCR для
                              ; прийому даних. Встановлення
                              ; TWEA означає,  що інтерфейс
                              ; переходить у режим
                              ; неадресованого веденого з
                              ; розпізнаванням власної адреси
                              ; (блок 16)
;підпрограма обробки помилки
         ERROR:
             …
             …
```

3.2.3.4 Програма роботи у режимі «Ведений передавач»

```
; Додатково у лістинг повинні бути включені заголовочні файли
; пристрою з оголошенням змінних, що відповідають
; регістрам TWDR, TWSR, TWCR, TWAR, TWBR та їх окремим бітам,
; та ініціалізація регістрів TWAR, TWCR, TWBR

         ldi r16,
             (1<<TWINT)|
             (1<<TWEA) |
             (1<<TWEN)
         out TWCR, r16          ; увімкнути режим веденого
                                передавача (блок 1)
; отримати стан START і SLA+R
wait14:  in r16, TWCR           ; очікувати встановлення
         sbrs r16, TWINT        ; прапорця TWINT. Це означає,
         rjmp wait14            ; що SLA+R отримано та $\overline{ACK}/NACK$
                                ; повернуто (блок 2)
```

```
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16, $A8            ; Якщо значення відрізняється
        brne ERROR              ; від ST_SLA_ACK , перейти до
                                ; обробки помилки (блок 3)
        ldi r16, 0x33           ; завантажити дані (0x33) в
        out TWDR, r16           ; TWDR (блок 4)
        ldi r16, (1<<TWINT)|
            (1<<TWEA)|
            (1<<TWEN)
        out TWCR, r16           ; скинути біт TWINT для початку
                                ; передачі. Встановлення TWEA
                                ; означає, що ACK буде
                                ; отримано, коли передачу байта
                                ; завершено. (блок 5)
; передача наступних даних
wait15: in r16, TWCR            ; очікувати встановлення
        sbrs r16, TWINT         ; прапорця TWINT. Це означає,
        rjmp wait15             ; що дані передано та ACK/NACK
                                ; отримано (блок 6)
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16, $B8            ; Якщо значення відрізняється
        brne ERROR              ; від ST_DATA_ACK , перейти до
                                ; обробки помилки (блок 7)
        ldi r16, 0x44           ; завантажити дані (0x44) в
        out TWDR, r16           ; TWDR (блок 8)
        ldi r16, (1<<TWINT)|    ; скинути біт TWINT для початку
            (1<<TWEN)           ; передачі. Скидання TWEA
        out TWCR, r16           ; означає, що NACK буде
                                ; отримано після передачі
                                ; байта даних (ведучий
                                ; сигналізує про завершення
                                ; прийому). (блок 9)
wait16: in r16, TWCR            ; очікувати встановлення
        sbrs r16, TWINT         ; прапорця TWINT. Це означає,
        rjmp wait16             ; що дані переданота ACK/NACK
                                ; отримано (блок 10)
        in r16, TWSR            ; перевірити регістр стану.
        cpi r16, $C0            ; Якщо значення відрізняється
        brne ERROR              ; від ST_DATA_NACK, перейти до
                                ; обробки помилки (блок 11)
        ldi r16, (1<<TWINT)|    ; продовжувати розпізнавання
            (1<<TWEA)|          ; адреси у режимі веденого
            (1<<TWEN)           ; передавача (блок 13)
        out TWCR, r16
;підпрограма обробки помилки
        ERROR:
            …
            …
```