

# ПОСЛІДОВНИЙ ПЕРИФЕРІЙНИЙ ІНТЕРФЕЙС SPI

## 1 ЗАГАЛЬНІ ВІДОМОСТІ

Крім задачі передачі неперервного потоку інформації, достатньо часто необхідно передавати окремі цифрові пакети чи команди керування. Ці пакети можуть передаватись досить рідко. Саме для передачі такого виду інформації призначений синхронний послідовний периферійний інтерфейс (SPI) [1,2].

Шина послідовного периферійного інтерфейсу (SPI) з самого початку була розроблена компанією Motorola наприкінці 1980-х для мікроконтролерів серії 68000. Завдяки простоті та популярності шини, багато інших виробників використовують цей стандарт вже багато років.

В даній лекції розглядається загальна структура та функціонування модуля SPI, детально описуються режими передачі даних, а також представлений перелік пристроїв з даним інтерфейсом.

Послідовний периферійний інтерфейс SPI (Serial Peripheral Interface), який реалізований у всіх мікроконтролерах сімейства Mega та Xmega, має подвійне призначення. По-перше, з його допомогою може здійснюватися обмін даними на відстань до 3-ох метрів зі швидкістю до 2 Мбіт/с між мікроконтролером і різними периферійними пристроями, такими, як цифрові потенціометри, ЦАП/АЦП, FLASH-ПЗП і т.ін. За допомогою цього інтерфейсу також може відбуватися обмін даними між декількома AVR-мікроконтролерами. Використання інтерфейсу SPI в якості високошвидкісного каналу зв'язку і розглядається в даному розділі.

Крім того, через інтерфейс SPI може бути здійснено програмування мікроконтролера (т.зв. режим послідовного програмування).

При обміні даними інтерфейсом SPI AVR-мікроконтролер може працювати як ведучий (режим «Master»), або як ведений (режим «Slave»). При цьому користувач може задавати швидкість передачі (сім

програмованих значень) і формат передачі (від молодшого розряду до старшого або навпаки).

Додатковою можливістю інтерфейсу SPI є «пробудження» мікроконтролера з режиму Idle при надходженні даних.

## **2 Використання модуля SPI**

### **2.1 Загальна характеристика архітектури модуля SPI**

Структурна схема модуля SPI наведена на рисунку 1. Модуль SPI використовує чотири виводи мікроконтролера. Як і для більшості інших периферійних пристроїв, ці виводи є лініями портів введення/виведення загального призначення. Як приклад, в таблиці 1 наведені виводи, які використовуються модулем SPI деяких МК-в сімейства Mega.

При включеному модулі SPI режим роботи зазначених виводів (напряв передачі даних) перевизначається згідно з таблицею 2. Напряв передачі даних визначається станом відповідного розряду регістра DDRB.

Як видно з таблиці 2, у певних випадках користувач повинен самостійно задати режим роботи виводу, що використовується модулем SPI, відповідно до його призначення (див. далі). При цьому можливість керування внутрішніми підтягуючими резисторами виводів, що працюють як входи, зберігається незалежно від способу керування їхнім режимом роботи.

Вивід SCK мікроконтролера в режимі Master працює як вихід, тому відповідна лінія порту введення/виведення загального призначення повинна бути запрограмована на виведення.

Для керування модулем SPI призначений регістр керування SPCR, який в моделях ATmega8515x/8535x/162x/8x/16x/32x/64x/128x розташований за адресою \$0D (\$2D), а в інших моделях – за адресою

\$2C (\$4C). Формат цього регістра наведено на рисунку 2, а короткий опис функцій розрядів регістра наведено в таблиці 3. Докладне використання різних розрядів регістра буде описано далі [2].

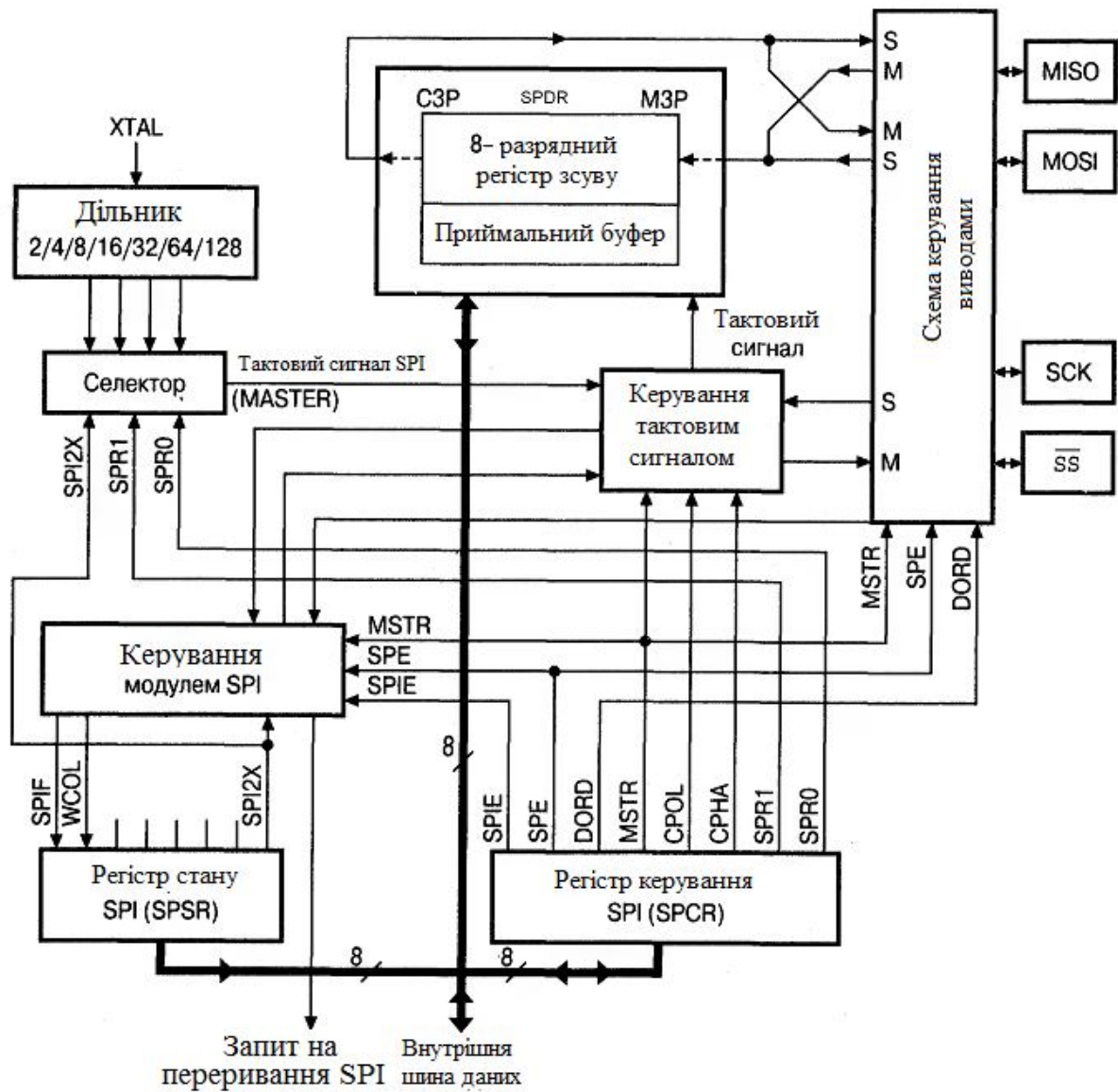


Рисунок 1 – Структурна схема модуля SPI

Таблиця 1 – Виводи, що використовуються модулем SPI

Вивід	ATmega8x /48x/88x/168x	ATmega8515x/8535x	ATmega16x/32x	ATmega161x/323x	ATmega162x/163x	ATmega164x/324x/644x	Atmega165x/64x/128x	ATmega325x/3250x/645x/6450x	ATmega640x/1280x/1281x/2560x/2561x	Призначення
SCK	PB5	PB7	PB7	PB7	PB7	PB7	PB1	PB1	PB1	Вихід (master) /вхід (slave) тактовий сигнал
MISO	PB4	PB6	PB6	PB6	PB6	PB6	PB3	PB3	PB3	Вхід (master)/вихід (slave) даних
MOSI	PB3	PB5	PB5	PB5	PB5	PB5	PB2	PB2	PB2	Вихід (master)/вхід (slave) даних
$\overline{SS}$	PB2	PB4	PB4	PB4	PB4	PB4	PB0	PB0	PB0	Вибір веденого пристрою

Таблиця 2 – Перепризначення режиму роботи виводів модуля SPI

Вивід	Режим «Master»	Режим «Slave»
MOSI	Визначається користувачем як вихід	Вхід
MISO	Вхід	Визначається користувачем як вихід
SCK	Визначається користувачем як вихід	Вхід
$\overline{SS}$	Визначається користувачем як вхід або вихід	Вхід

	7	6	5	4	3	2	1	0
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Читання(R)/ Запис (W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Початкове значення	0	0	0	0	0	0	0	0

Рисунок 2 – Формат регістра SPCR

Таблиця 3 – Призначення розрядів регістра SPCR

Розряд	Назва	Опис
7	SPIE	Дозвіл переривання від SPI, якщо SPIE = 1
6	SPE	Ввімкнення/вимкнення модуля SPI: SPI = 1 – ввімкнений; SPI = 0 – вимкнений
5	DORD	Порядок передачі даних: DORD = 1 – першим передається МЗР, DORD = 0 – першим СЗР
4	MSTR	Вибір режиму роботи («Master»/«Slave»): MSTR = 1 – Master, MSTR = 0 – Slave
3	CPOL	Визначає полярність тактового сигналу, див. таблицю 4.5
2	CPHA	Фаза тактового сигналу – визначає момент зчитування сигналу, див. таблицю 4.5
1, 0	SPR1:SPR0	Швидкість передачі, див. таблицю 4.6

Контроль стану модуля, а також додаткове керування швидкістю обміну здійснюється за допомогою регістра стану SPSR, розташованого за адресою \$0E(\$2E) в моделях ATmega8515x/8535x/162x/8x/16x/32x/64x/128x, та за адресою \$2D (\$4D) – в інших моделях. Розряди з 7-го по 1-й цього регістра доступні тільки для читання, а 0-й розряд – як для читання, так і для запису. Формат цього регістра наведено на рисунку 3, а призначення його розрядів описано в таблиці 4.

	7	6	5	4	3	2	1	0
	SPIF	WCOL	—	—	—	—	—	SPI2X
Читання(R)/ Запис (W)	R	R	R	R	R	R	R	R/W
Початкове значення	0	0	0	0	0	0	0	0

Рисунок 3 – Формат регістра SPSR

Таблиця 4.4 – Призначення розрядів регістра SPSR

Розряд	Назва	Опис
7	SPIF	<u>Прапорець «Кінець передачі»</u> Даний прапорець встановлюється в «1» по закінченні передачі/прийому чергового байта. Якщо прапорець SPIE регістра SPCR встановлений в «1» і переривання дозволені, одночасно з встановленням прапорця генерується переривання від модуля SPI. Також прапорець SPIF встановлюється в «1» при переведенні мікроконтролера з режиму «Master» у режим «Slave» за допомогою виводу $\overline{SS}$ , див.2.3. Прапорець скидається апаратно, або при старті підпрограми обробки переривання, або після читання регістра стану SPI з наступним зверненням до регістра даних SPI (SPDR)
6	WCOL	<u>Прапорець конфлікту запису</u> Даний прапорець встановлюється в «1» при спробі запису в регістр даних (SPDR) під час передачі чергового байта. Прапорець скидається апаратно після читання регістра стану SPI з наступним зверненням до регістра даних SPI
5...1	-	Зарезервовано, читаються як «0»
0	SPI2X	<u>Подвоєння швидкості обміну</u> При встановленні цього розряду в «1» і роботі мікроконтролера в режимі «Master» частота сигналу SCK подвоюється

Дані, що передаються, записуються, а дані, які приймаються, зчитуються з регістру даних SPDR, розташованого за адресою \$0F (\$2F) в моделях ATmega8515x/8535x/162x/8x/16x/32x/64x/128x, а в інших моделях – за адресою \$2E (\$4E). Запис у цей регістр ініціює початок передачі, а при

його читанні зчитується вміст приймаючого буфера регістра зсуву. Інакше кажучи, регістр даних служить буфером між регістровим файлом мікроконтролера і регістром зсуву модуля SPI. При записі роль регістра даних виконує регістр зсуву (немає додаткового буфера).

З'єднання двох мікроконтролерів (ведучий-ведений) інтерфейсом SPI показано на рисунку 4. Вивід SCK ведучого мікроконтролера є виходом тактового сигналу, а веденого мікроконтролера – входом.

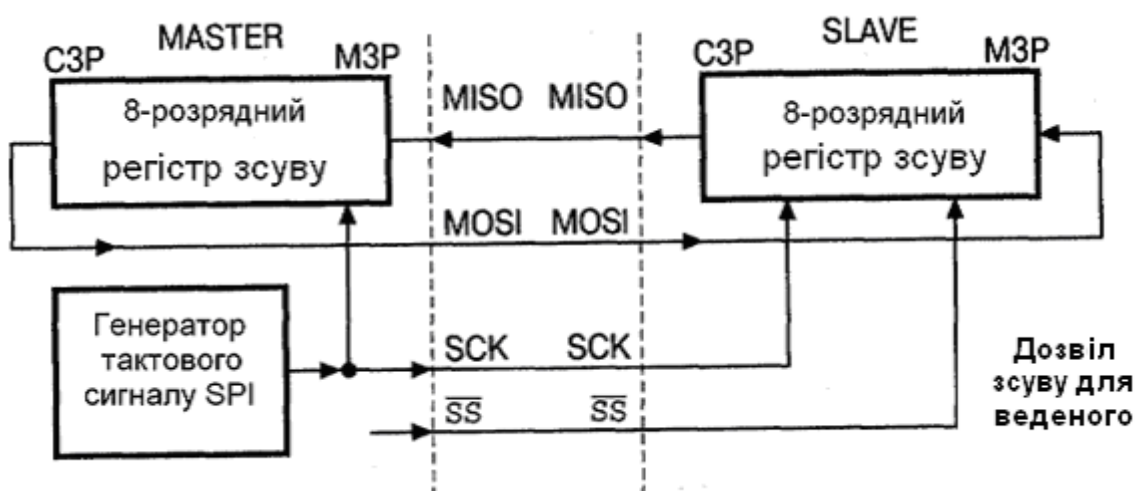


Рисунок 4 – З'єднання мікроконтролерів інтерфейсом SPI

Перед виконанням обміну необхідно, насамперед, дозволити роботу модуля SPI. Для цього потрібно встановити в «1» розряд SPE регістра SPCR. Режим роботи визначається станом розряду MSTR цього регістра: якщо розряд встановлено в «1», мікроконтролер працює в режимі «Master», якщо скинуто в «0» – у режимі «Slave».

Передача даних здійснюється в наступний спосіб. При записі в регістр даних SPI ведучого мікроконтролера запускається генератор тактового сигналу модуля SPI, і дані починають порозрядно видаватися на вивід MOSI і, відповідно, надходити на вивід MOSI веденого мікроконтролера. Порядок передачі розрядів даних визначається станом

розряду DORD регістра SPCR. Якщо розряд встановлено в «1», першим передається молодший розряд байта, якщо ж скинутий в «0» – старший розряд. Після видачі останнього розряду поточного байта генератор тактового сигналу зупиняється з одночасним встановленням в «1» прапорця «Кінець передачі» (SPIF). Якщо переривання від модуля SPI дозволені (прапорець SPIE регістра SPCR встановлений в «1»), генерується запит на переривання. Після цього ведучий мікроконтролер, записом чергового байта в регістр SPDR, може почати передачу наступного байта (режими 2, 3, рисунок 4.8, б), або, подавши на вхід  $\overline{SS}$  веденого мікроконтролера напругу ВИСОКОГО рівня, перевести останній у стан очікування, після чого (режими 0, 1), подавши на вхід  $\overline{SS}$  веденого напругу НИЗЬКОГО рівня, почати передачу наступного байта (рисунок 4.8, а). Одночасно з передачею даних від ведучого до веденого відбувається передача у зворотному напрямку за умови, що на вході  $\overline{SS}$  веденого присутня напруга НИЗЬКОГО рівня. Таким чином, у кожному циклі зсуву відбувається обмін даними між двома пристроями. Наприкінці кожного циклу прапорець SPIF встановлюється в «1» як у ведучому мікроконтролері, так і у веденому. Прийняті байти зберігаються в приймальних буферах для подальшого використання.

У модулі реалізована одинарна буферизація при передачі і подвійна при прийомі. Це означає, що готовий для передачі байт даних не може бути записаний у регістр даних SPI до закінчення попереднього циклу обміну. При спробі змінити вміст регістра даних під час передачі встановлюється в «1» прапорець WCOL регістра SPSR. Цей прапорець скидається після читання регістра SPSR з наступним зверненням до регістра даних SPI.

Відповідно, при прийомі даних прийнятий байт повинен бути прочитаний з регістра даних SPI до того, як у регістр зсуву надійде останній розряд наступного байта. У іншому випадку перший байт буде загублений.



Як показано на рисунку 5, до інтерфейсу ведучого SPI в якості ведених можуть бути одночасно підключені декілька периферійних пристроїв, але активним буде тільки той, у якого на вхід  $\overline{SS}$  через відповідну лінію порту ведучого пристрою буде подано рівень лог. 0. Виводи MISO незадіяних ведених блоків знаходяться у високоомному стані і не впливають на процес передачі даних.

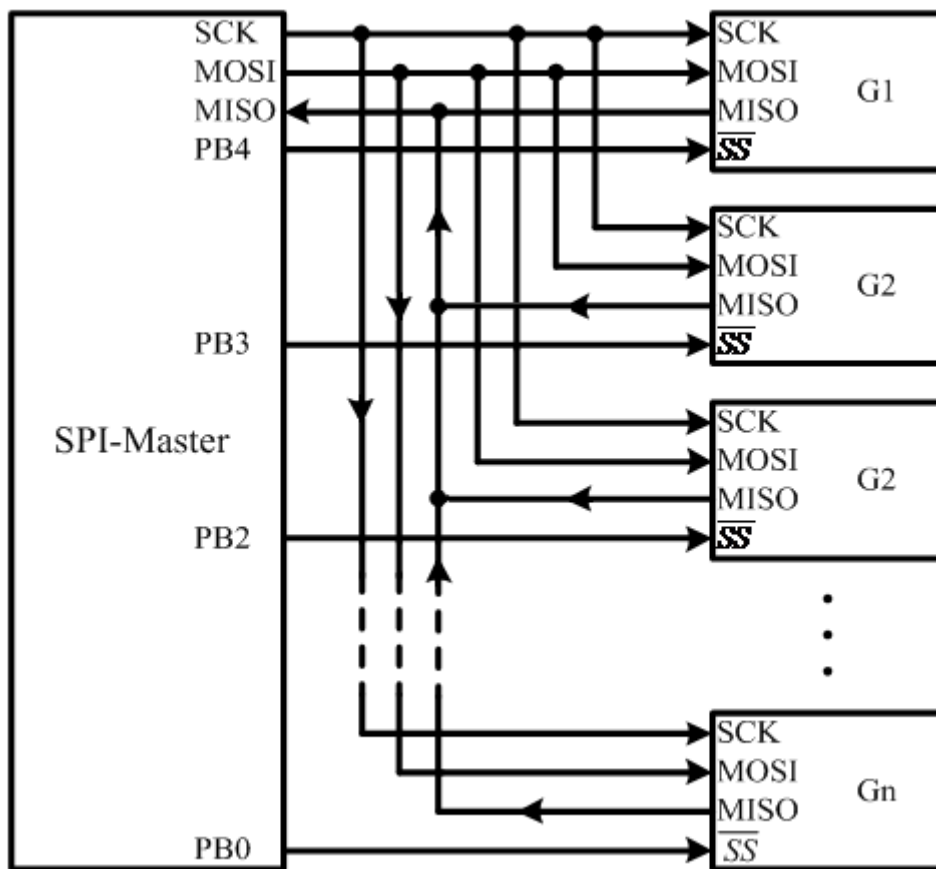


Рисунок 5 – Підключення декількох пристроїв до інтерфейсу ведучого SPI

Блокам G1, G2 ... Gn на рисунку 5 відповідають повноцінні ведені інтерфейси SPI, у яких дані передаються в обох напрямках. Блок G2, з точки зору ведучого блоку, може бути, наприклад, тільки блоком видачі (в якості такого блоку можна привести аналого-цифровий перетворювач з інтерфейсом SPI). Блок G3 за функціональністю може бути аналогічний

блоку G2, але тільки як блок прийому (наприклад, цифро-аналоговий перетворювач).

## 2.2 Режими передачі даних

Специфікація інтерфейсу SPI передбачає 4 режими передачі даних. Ці режими розрізняються відповідністю між фазою тактового сигналу SCK (визначає момент зчитування сигналу), його полярністю та даними, які передаються. Усього існує 4 такі комбінації, обумовлені станом розрядів CPOL і CPHA і CPOL регістра SPCR (таблиця 5).

Таблиця 5 – Задання режиму передачі даних

Розряд	Опис
CPOL	<u>Полярність тактового сигналу</u> «0» – генеруються імпульси додатної полярності, при відсутності імпульсів на виводі присутній НИЗЬКИЙ рівень; «1» – генеруються імпульси від’ємної полярності, при відсутності імпульсів на виводі присутній ВИСОКИЙ рівень (рисунки 6, 7)
CPHA	<u>Фаза тактового сигналу</u> «0» – обробка даних виконується за переднім фронтом імпульсів сигналу SCK (для CPOL = «0» – за наростаючим фронтом, а для CPOL = «1» – за спадаючим фронтом) (рисунок 6); «1» – обробка даних виконується за заднім фронтом імпульсів сигналу SCK (для CPOL = «0» – за спадаючим фронтом, а для CPOL = «1» – за наростаючим фронтом) (рисунок 7)

Формати обміну даними через SPI, які відповідають цим режимам наведені на рисунку 6 і рисунку 7 (передача ведеться від старшого розряду до молодшого).

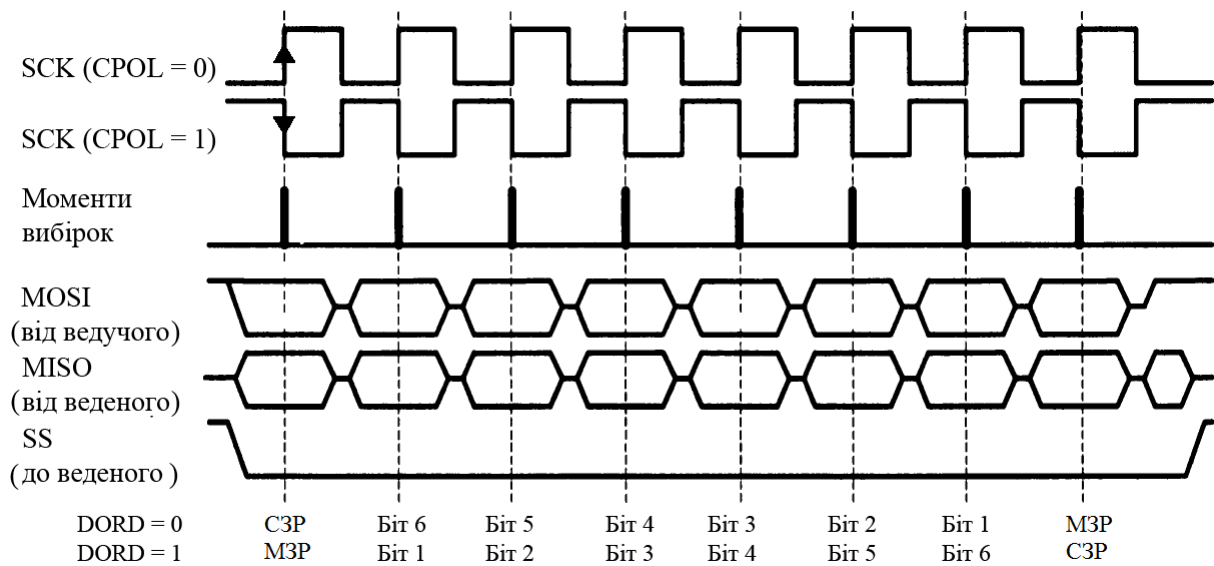


Рисунок 6 – Передача даних при СРНА = «0» (режими 0, 1)

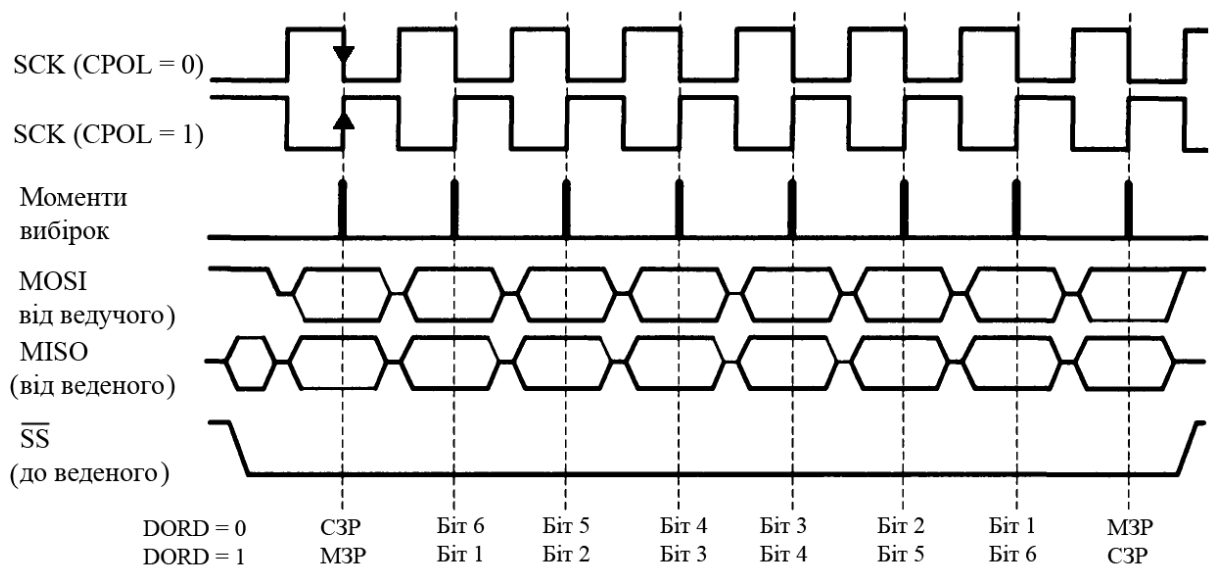


Рисунок 7 – Передача даних при СРНА = «1» (режими 2, 3)

У режимах 0, 1 (СРНА = 0) початок обміну визначається встановленням сигналу на виводі  $\overline{SS}$  веденого в активний стан – лог. 0. При цьому ведений виставляє на лінію MISO старший розряд даних, якщо першим передається СЗР, або молодший – якщо першим передається МЗР

(рисунок 6). Ведучий видає на лінію MOSI СЗР (МЗР) на  $0,5 \cdot T_{SCK}$  раніше, ніж на лінії SCK з'являється перший імпульс ( $T_{SCK}$  – період синхроімпульсів).

Перед входами восьмирозрядних регістрів зсуву ведучого та веденого мікроконтролерів (рисунок 4) знаходяться два синхронних тригери (буфери) (на рисунку не показані), в які першим перепадом сигналу на лінії SCK записується значення сигналу, яке присутнє на їх інформаційному вході. Цей момент на рисунку 6 відмічений на лінії «Моменти вибірок».

Другим перепадом імпульсів на лінії SCK відбувається зсув інформації вліво відповідно до рисунку 4. При цьому стан буферів переписується в молодші розряди регістрів зсуву, а черговий вихідний біт з регістрів зсуву виставляється на лінії MOSI/MISO.

Вказаний зсув у часі між моментом видачі чергового розряду в лінію зв'язку між мікроконтролерами і моментом фіксації цього біта в буфері дозволяє компенсувати часові затримки при передачі сигналів між мікроконтролерами.

Далі аналогічно здійснюється обмін між ведучим та веденим всіма наступними бітами. З кожним непарним перепадом сигналу SCK відбувається фіксація чергового біта в буфері, а з кожним парним – зсув інформації вліво.

В режимах 0, 1 сигнал на лінії вибору веденого повинен бути повернутий у неактивний стан (лог. 1) після передачі кожного байта в будь-якому напрямі. Тоді передача кожного нового байта буде супроводжуватися попереднім встановленням  $\overline{SS}$  в «0» (рисунок 8, а), що буде визначати початок обміну наступним байтом.

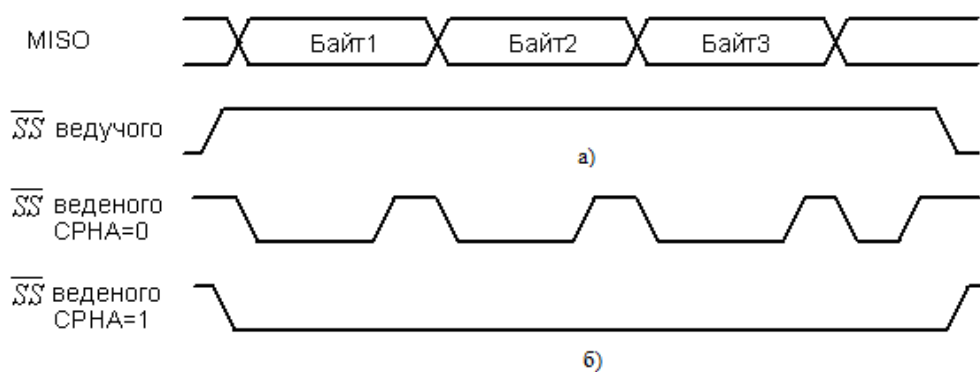


Рисунок 8 – Вплив сигналу  $\overline{SS}$  на початок обміну даними

У режимах 2, 3 (CPHA = 1) початок обміну визначає перша зміна сигналу на лінії SCK після встановлення сигналу вибору веденого  $\overline{SS}$  в активний стан лог. 0 (рисунок 7).

Моменти фіксації біта в буферах перед входами регістрів зсуву передавача та приймача на цьому рисунку відмічені на лінії «Моменти вибірок».

При передачі даних від ведучого до веденого та у зворотному напрямку всі непарні перепади SCK викликають видачу чергового біта послідовності з регістру зсуву передавача на лінію. Кожний парний перепад використовується для запису цього біта в буфер перед регістром зсуву. Тому сигнал вибору веденого може залишатися в активному стані лог. 0 протягом передачі декількох байт інформації (рисунок 8,б). Це трохи спрощує логіку програмного драйвера SPI.

Біти даних виводяться (висуваються) з виходів регістрів зсуву фронтами сигналу SCK, а фіксуються в буферах на входах регістрів фронтами синхросигналу SCK, які зсунуті на половину періоду (відносно моментів зміни бітів на виході регістрів зсуву) (рисунок 7). Це гарантує достатній час на встановлення даних на входах відповідних регістрів після висування.

Частота тактового сигналу SCK і, відповідно, швидкість передачі даних інтерфейсом визначаються станом розрядів SPR1:SPR0 регістра SPCR і розряду SPI2X регістра SPSR (таблиця 6). Зрозуміло, мова йде про мікроконтролер, який працює в режимі «Master», тому що саме він є джерелом тактового сигналу. Для пристрою, який перебуває в режимі «Slave», стан цих розрядів байдужий.

Таблиця 6 – Задання частоти тактового сигналу SCK

SPI2X	SPR1	SPR0	Частота сигналу SCK
0	0	0	$f_{CLK}/4$
0	0	1	$f_{CLK}/16$
0	1	0	$f_{CLK}/64$
0	1	1	$f_{CLK}/128$
1	0	0	$f_{CLK}/2$
1	0	1	$f_{CLK}/8$
1	1	0	$f_{CLK}/32$
1	1	1	$f_{CLK}/64$

*Примітка:*  $f_{CLK}$  – тактова частота мікроконтролера.

Варто мати на увазі, що функціонування мікроконтролера в режимі «Slave» гарантується тільки на частотах, менших або рівних  $f_{CLK}/4$  [ 1,2 ].

Максимальна швидкість передачі обмежується часовими діаграмами встановлення та утримання сигналів. SPI використовується в широкому діапазоні швидкостей зв'язку, починаючи з кількох Кбіт/с і до кількох Мбіт/с.

### 2.3 Використання виводу $\overline{SS}$

Цей вивід призначений для вибору активного веденого пристрою і у режимі «Slave» завжди є входом (рисунок 5). При подачі на нього напруги НИЗЬКОГО рівня модуль SPI активується і вивід MISO переключається в режим виведення даних (якщо це задано користувачем, таблиця 2). Інші виводи модуля SPI в цьому режимі є входами. А при подачі на вивід  $\overline{SS}$  напруги ВИСОКОГО рівня всі виводи модуля SPI переключаються в режим введення даних. При цьому модуль переходить у неактивний стан і прийом даних не відбувається. Як правило, у цьому стані програма змінює вміст регістра даних.

Варто пам'ятати, що щораз, коли на вивід  $\overline{SS}$  веденого подається напруга ВИСОКОГО рівня, відбувається скидання модуля SPI. Відповідно, якщо зміна стану цього виводу відбудеться під час обміну даними, і прийом, і передача негайно припиняться, а переданий і прийнятий байти будуть загублені.

Якщо ж мікроконтролер перебуває в режимі «Master» (розряд MSTR регістра SPCR встановлений в «1»), напрям передачі даних через вивід  $\overline{SS}$  визначається користувачем. Якщо вивід сконфігурований як вихід, він використовується для керування виводом  $\overline{SS}$  мікроконтролера, що працює в режимі «Slave».

Якщо ж в режимі «Master» вивід сконфігурований як вхід, то для забезпечення нормальної роботи модуля SPI на нього повинна бути подана напруга ВИСОКОГО рівня. Подача на цей вхід напруги НИЗЬКОГО рівня від якої-небудь зовнішньої схеми буде сприйнята модулем SPI як вибір даного мікроконтролера в якості веденого, і відповідно, початок передачі йому даних. Щоб уникнути конфлікту на шині модуль SPI у таких випадках виконує наступні дії.

1) Прапорець MSTR регістра SPCR скидається, і мікроконтролер переключається в режим «Slave». Як наслідок, виводи MOSI і SCK починають функціонувати як входи (таблиця 2).

2) Встановлюється прапорець SPIF регістра SPSR, генеруючи запит на переривання від SPI. Якщо переривання від SPI дозволені і прапорець I регістра SREG встановлений в «1», відбувається запуск підпрограми обробки переривання.

Таким чином, якщо ведучий мікроконтролер використовує передачу даних, керовану перериванням, і існує ймовірність подачі на вхід  $\overline{SS}$  напруги НИЗЬКОГО рівня, у підпрограмі обробки переривання від SPI обов'язково повинна здійснюватися перевірка стану прапорця MSTR. При виявленні скидання цього прапорця він повинен бути програмно встановлений назад в «1» для зворотного переведення мікроконтролера в режим «Master».

### **3 Режим програмування пам'яті послідовним каналом**

Режим програмування пам'яті послідовним каналом підтримується всіма мікроконтролерами сімейства Mega, XМega, а також деякими мікроконтролерами сімейства Tiny. У цьому режимі програмування пам'яті програм і даних здійснюється через послідовний інтерфейс SPI. Як правило, розглянутий режим використовується для програмування (перепрограмування) мікроконтролера безпосередньо в пристрої.

Схема включення мікросхеми у режимі програмування послідовним каналом наведена на рисунку 9.





**Зауваження (для ATmega) - якщо у якості тактового використовується внутрішній RC-генератор, вихід XTAL1 залишають не підключеним**

Рисунок 9 – Включення мікроконтролерів у режимі програмування послідовним каналом

Як видно з рисунка 9, для підключення програматора до пристрою використовуються три лінії інтерфейсу: SCK (тактовий сигнал), MOSI (вхід даних) і MISO (вихід даних). Відповідність між лініями інтерфейсу і контактами портів виведення/виведення деяких мікроконтролерів наведено в таблиці 7.

Таблиця 7 – Виводи, що використовуються при програмуванні послідовним каналом

Назва лінії інтерфейсу	ATmega8515x/8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x ATmega325x/3250x ATmega645x/6450x	ATmega640x/1280x/2560x	ATmega1281x/ATmega2561x	Призначення виводів
SCK	PB7	PB5	PB7	PB1	PB5	PB7	PB7	PB1	PB1	PB1	Вхід тактового сигналу
MISO (PDO)	PB6	PB4	PB6	PE1	PB4	PB6	PB6	PB3	PE1	PB3	Вихід даних
MOSI (PDI)	PB5	PB3	PB5	PE0	PB3	PB5	PB5	PB2	PE0	PB2	Вхід даних

Варто мати на увазі, що в окремих мікроконтролерах, наприклад, ATmega64x і ATmega128x виводи, які використовуються для програмування, не збігаються з виводами, призначеними для штатної роботи інтерфейсу SPI (таблиця 7).

Часові діаграми сигналів при програмуванні мікроконтролерів у розглянутому режимі представлені на рисунку 4.10, а значення параметрів сигналів наведені в [1,2].

Як і в робочому режимі, при програмуванні послідовним каналом мікроконтролеру потрібно джерело тактового сигналу. У якості такого може використовуватися кожне із можливих для мікроконтролера джерел. При цьому повинна виконуватися наступна умова: тривалість імпульсів як НИЗЬКОГО рівня, так і ВИСОКОГО рівня сигналу SCK повинна бути більше 2-х (при  $f_{CLK} < 12$  МГц) або 3-х (при  $f_{CLK} \geq 12$  МГц) періодів тактового сигналу мікроконтролера.

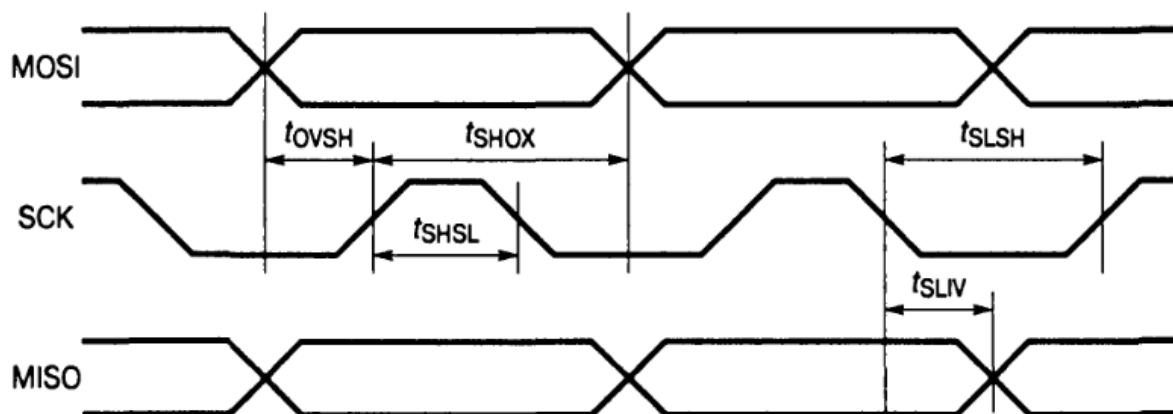


Рисунок 10 – Часові діаграми сигналів при програмуванні послідовним каналом

Програмування здійснюється шляхом посилки 4-байтних команд на вивід MOSI мікроконтролера. Результат виконання команд читання знімається з виводу MISO мікроконтролера. Передача команд і видача

результатів їхнього виконання здійснюється від старшого розряду до молодшого. При цьому запис вхідних даних виконується за наростаючим фронтом сигналу SCK, а зсув вихідних даних – за спадаючим (рисунок 11).

Більш детально це питання розглянуте у [1,2].

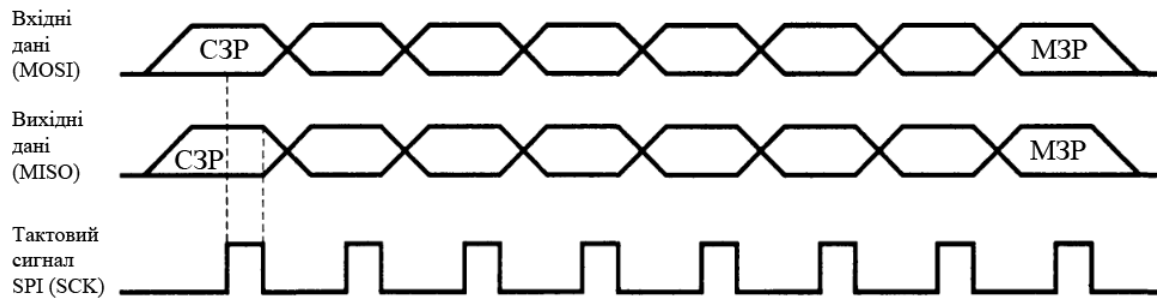


Рисунок 11 – Передача даних при програмуванні послідовним каналом

#### 4 Периферійні пристрої з SPI-інтерфейсом

Багато хто наштовхувався на проблему вибору елементної бази при проектуванні електричних схем. Для спрощення вирішення цієї задачі нижче наведена таблиця 8, в якій наведені деякі мікросхеми, що містять інтерфейс SPI.

Таблиця 8 – Мікросхеми, що містять інтерфейс SPI

Найменування	Короткий опис
AD5160	Виробник: Analog Devices 8-ми бітний цифровий потенціометр Номінальний опір: 5 К, 10 К, 50 К, 100 К Напруга живлення: 2,7...5.5 В

Найменування	Короткий опис
DS1267	<p>Виробник: Dallas Semiconductors  <i>Двоканальний цифровий монолітний потенціометр</i>            Стандартні опори: 10 Ом, 50 кОм і 100 кОм.            Діапазони температур:                – комерційний: 0...70°C;                – індустриальний: -40...+85°C</p> <p>Напруга живлення: 5 В</p>
MAX5481	<p>Виробник: Maxim  <i>10-бітний енергонезалежний лінійний цифровий потенціометр</i>            Стандартні опори: 10 Ом, 50 кОм            Однополярне живлення: 2.7...5.25 В            Двополярне живлення: ±2.5 В</p>
ADS7816	<p>Виробник: Burr-Brown  <i>12-ти бітний одноканальний АЦП з диференціальним входом</i>            Максимальна частота перетворення: 200 КГц.            Напруга живлення: 4,5...5.5 В            Автовимикання (при живленні від батареї)</p>
TLC1518	<p>Виробник: Texas Instruments  <i>8-ми канальний 10-ти бітний АЦП послідовного наближення</i>            Максимальна частота перетворення: 500 КГц.            Напруга живлення: 4.5...5.5 В            Режим автопрограмування</p>
DAC7512	<p>Виробник: Texas Instruments  <i>12-ти бітний одноканальний ЦАП з виходом 0..5 В (rail-to-rail вихід)</i>            Вихідний опір: 1 Ом            Максимальний вихідний струм: 50 мА            Час становлення вихідного сигналу: 10 мкс            Напруга живлення: 2.7...5.5 В</p>

Найменування	Короткий опис
AD5308	<p>Виробник: Analog Devices  <i>Чотири 8-разрядних ЦАП в одному 16-вивідному TSSOP корпусі з підсилювачами виходів</i>  Енергоспоживання: 0,7 мА при 3 В живлення  Температурний діапазон: -40...+105°C</p>
DS1307	<p>Виробник: Dallas Semiconductors  <i>Годинник реального часу з можливістю роботи від батареї</i>  Автоматичне визначення дня тижня аж до 2100 року  56 регістри даних та керування. 4 режими зарядки батареї  Напруга живлення: 1,8...5.5 В</p>
AT45DB161B	<p>Виробник: Atmel  <i>16-мегабітна Flash-пам'ять</i>  2 буфери: один для зчитування, інший - для запису; по 528 байт кожен.  Низьковольтна логіка та живлення: 2.7...3.6 В  4096 сторінок по 528 байт кожна  З мікроконтролером потрібно з'єднувати через схему узгодження рівнів</p>
W25Q32	<p>Виробник: Rainbow Technologies  <i>Послідовна флеш-пам'ять ємністю 32 Мб</i>  Напруга живлення: 2.7...3.6 В  Температурний діапазон: -40...+85°C  20-річне зберігання даних</p>
DS1626	<p>Виробник: Dallas Semiconductors  <i>Високоточні цифрові термометри/термостати</i>  Діапазон вимірюваної температури: -55...+125°C  Діапазон напруги живлення: 2.7...5.5 В  Можливість обрання користувачем 9-, 10-, 11- чи 12-бітної розрядності</p>
MAX6627	<p>Виробник: Maxim  <i>Вимірювач віддаленої температури в корпусі SOT</i>  Температурний діапазон: -55...+125°C  Діапазон напруги живлення: 3...5.5 В</p>

Найменування	Короткий опис
MAX3420E	Виробник: Maxim <i>Периферійний USB-контролер</i> Відповідає стандарту USB 2.0 Подвійна буферизація кінцевих точок
NRF905-MOD	Виробник: TechToys <i>Безпроводний трансивер на базі NRF905</i> Інтерфейс підключення до контролера – SPI Функція автоматичного визначення контрольної суми Напруга живлення: 1.9...3.6 В Вихідна потужність досягає 10 дБ

## 5 Універсальний послідовний інтерфейс USI

### 5.1 Загальні відомості

Модуль універсального послідовного інтерфейсу (Universal Serial Interface – USI) реалізований, наприклад в моделях ATmega165x та ATmega325x/3250/645x/6450x. Цей модуль є свого роду “напівфабрикатом”, який надає базові апаратні ресурси, що необхідні для здійснення обміну послідовним каналом. Використовуючи даний модуль, можна досягти суттєво більшої швидкості передачі та отримати більш компактний код, ніж при чисто програмній реалізації різноманітних протоколів обміну [2].

Спрощену структурну схему модулю USI наведено на рисунку 12.

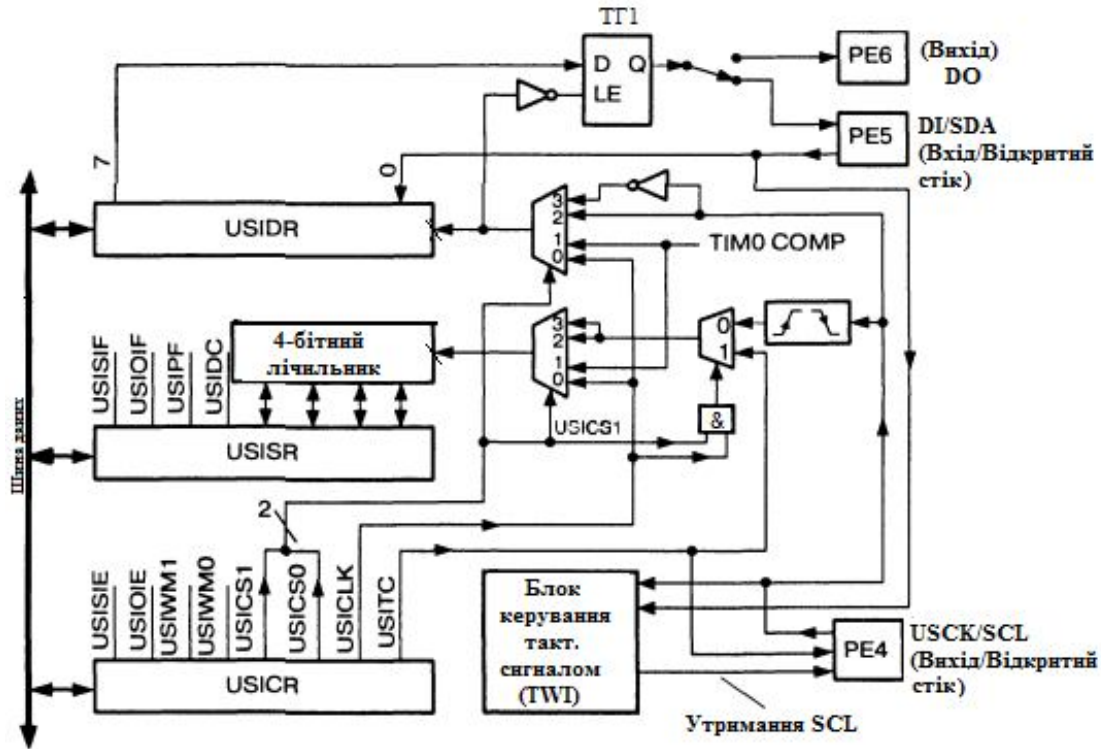


Рисунок 12 – Структурна схема модуля USI

Модуль USI використовує три лінії введення/виведення мікроконтролера:

- PE6 - вихід даних (DO). Використовується у трьохпровідному режимі(SPI);
- PE5 - вхід даних (DI)/лінія даних (SDA);
- PE4 - вхід/вихід тактового сигналу(USCK)/лінія тактового сигналу (SCL).

У 8-бітному регістрі зсуву (регістр USIDR) утримуються вхідні та вихідні дані. Старший біт регістра зсуву підключається, в залежності від режиму роботи, до однієї з двох ліній даних модуля. Вхідні біти завжди зчитуються з виводу DI, незалежно від конфігураційного модуля. Чотирьохбітний лічильник, наведений на рисунку 12, доступний як для зчитування, так і для запису. Крім того, при його переповненні може

генеруватися переривання. Оскільки регістр зсуву та лічильник використовують один і той самий тактовий сигнал, лічильник може підраховувати число переданих або прийнятих бітів та згенерувати переривання по закінченні процесу обміну. Зверніть увагу, що при використанні зовнішнього тактового сигналу зміна стану лічильника відбувається за кожним фронтом сигналу, тобто лічильник рахує кількість фронтів, а не бітів. Тактовий сигнал може зніматися з входу USCK, з виходу блоку порівняння таймера/лічильника T0 або формуватися програмно.

Блок керування тактовим сигналом використовується у двопровідному режимі та може генерувати переривання при виявленні на шині стану СТАРТ. Крім того, він може формувати на шині цикли очікування, утримуючи на лінії SCL НИЗЬКИЙ рівень після виявлення стану СТАРТ або після переповнення лічильника.

## **5.2 Використання модуля універсального послідовного інтерфейсу**

Взаємодія з модулем USI здійснюється за допомогою трьох регістрів введення/виведення: регістра даних USIDR, розташованого за адресою (\$BA), регістра стану USISR, розташованого за адресою (\$B9) та регістра керування USICR, розташовано за адресою (\$B8).

При зверненні до регістра даних USIDR здійснюється доступ безпосередньо до регістра зсуву модуля. Тактовий сигнал, який використовується для тактування регістра зсуву, визначається станом бітів USICS1:0 регістра USICR. Старший біт регістра даних підключається до вихідного контакту модуля (DO або SDA) через тригер-защіпку ТГ1. При використанні зовнішнього тактового сигналу (USICS1 = 1) защіпка відкрита в першому напівперіоді тактового сигналу, а при використанні



внутрішнього тактового сигналу (USICS1 = 0) - відкрита постійно. Застосування тригера-защипки гарантує, що зчитування даних з входу та видача їх на вихід будуть здійснюватись за протилежними фронтами тактового сигналу.

Формат регістру керування USICR наведений на рисунку 4.13, а опис його бітів наведений в таблиці 9.

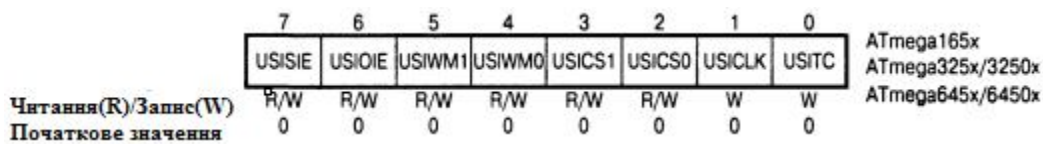


Рисунок 13 – Формат регістра USICR

Таблиця 9 – Біти регістра USICR

Біт	Назва	Опис
7	USISIE	<u>Дозвіл переривання при виявленні стану СТАРТ</u> Якщо в цьому біті записано лог. 1 та прапорець I регістра SREG також встановлено в 1, то переривання від детектору стану СТАРТ дозволено
6	USIOIE	<u>Дозвіл переривання при переповненні лічильника</u> Якщо в цьому біті записано лог. 1 та прапорець I регістра SREG також встановлено в 1, то переривання від лічильника дозволено
5	USIWM1	<u>Режим роботи модуля USI</u>
4	USIWM0	Ці біти визначають функціонування вихідних контактів модуля згідно з таблицею 4.10. Ці біти не впливають на входи тактового сигналу та даних, тому прийом даних, а також тактування регістра зсуву та лічильника зовнішнім тактовим сигналом можуть здійснюватись навіть якщо виходи модуля відключені від контактів введення/виведення мікроконтролера

Продовження таблиці 9

Біт	Назва	Опис
3	USICS1	<u>Вибір тактового сигналу</u> Ці біти визначають джерело тактового сигналу для регістра зсуву та лічильника модуля згідно з таблицею 11
2	USICS0	
1	USICLK	<u>Строб тактового сигналу</u> Призначення цього біта залежить від встановлень бітів USICS1:0 (див. таблицю 11). Якщо обраний режим програмного формування тактового сигналу (USICS1:0 = 00), то встановлення даного біта в 1 викликає зсув вмісту регістра зсуву та інкрементування лічильника. Одразу після встановлення стробу (в тому ж такту) міняється стан виходу модуля. В регістр зсуву заціпається значення, яке було зчитане в попередньому такті. При зчитуванні біта повертається 0. Якщо використовувати зовнішній тактовий сигнал (USICS1 = 1), то біт USICLK використовується в якості селектора – при встановленні його в 1 тактування лічильника здійснюється за допомогою біта USITC
0	USITC	<u>Переключення стану виводу тактового сигналу</u> При записі у цей біт лог. 1 стан лінії USCK/SCL змінюється на протилежний. Якщо формуючий сигнал необхідно вивести назовні, біт DDB4 регістра DDRE повинен бути встановлений в 1. При зчитуванні біта USITC повертається 0. Якщо використовується зовнішній тактовий сигнал (USICS1 = 1) та біт USICLK встановлений в 1, то запис лог. 1 в біт USITC визиває інкрементування лічильника. Завдяки цьому забезпечується раннє виявлення завершення обміну при роботі модуля в якості ведучого

Таблиця 10 – Задання режиму роботи модуля USI

USIWM1	USIWM0	Опис
0	0	Виходи модуля та детектор стану СТАРТ відключені. Виводи PE4...PE6 працюють як лінії введення/виведення загального призначення

Продовження таблиці 10

USIWM1	USIWM0	Опис
0	1	<p><u>Трьохпровідний режим</u>                      Використовуються виводи DO, DI та USCK.                      Значення на виході даних (DO) заміщує значення відповідного біта регістра порту. При цьому напрям передачі даних все одно визначається регістром DDRE. Якщо лінія порту налаштована на вхід, керування підтяжкою здійснюється відповідним бітом регістра PORTE.                      Вхід даних (DI) та лінія тактового сигналу (USCK) не впливає на нормальне функціонування порту мікроконтролера. При роботі в якості ведучого генерація тактових імпульсів здійснюється переключенням стану біта регістра PORTE (для цього можна використовувати біт USITC регістра USICR). При цьому вивід мікроконтролера має бути виходом (DDRE4 = 1)</p>
1	0	<p><u>Двопровідний режим</u>                      Використовуються виводи SDA та SCL.                      Обидва виводи є двонаправленими та підключені до виходів драйверами з відкритим стоком. Ці драйвери вмикаються встановленням в 1 відповідно бітів DDRE5 та DDRE4 регістра DDRE. Після ввімкнення драйвер лінії SDA формує на ній НИЗЬКИЙ рівень, якщо на виході регістра зсуву або в біті PORTE5 присутній 0. В іншому випадку лінія звільняється.                      Аналогічно, драйвер лінії SCL формує на ній НИЗЬКИЙ рівень або за сигналом від детектора стану СТАРТ (для звільнення лінії слід скинути прапорець USISIF регістра USISR), або якщо біт PORTE4 скинутий в 0. Таким чином, внутрішні підтягуючі резистори на лініях PE4 та PE5 в цьому режимі відімкнені. Стан контактів, відповідний лініям SDA та SCL, може бути зчитаний звичайним чином</p>
1	1	<p><u>Двопровідний режим</u>                      Використовуються виводи SDA та SCL.                      Цей режим аналогічний попередньому, за виключенням того, що НИЗЬКИЙ рівень на лінію SCL також видається при переповненні лічильника. В цьому стані лінія утримується до скидання прапорця переповнення USIOIF регістра USISR</p>

Таблиця 11 – Вибір тактових сигналів модуля USI

USICS1	USICS0	USICLK	Джерело тактових імпульсів регістра зсуву	Джерело тактових імпульсів 4-бітного лічильника
0	0	0	Тактовий сигнал відсутній	Тактовий сигнал відсутній
0	0	1	Програмний строб (USICLK)	Програмний строб (USICLK)
0	1	X	Блок порівняння таймера/лічильника T0	Блок порівняння таймера/лічильника T0
1	0	0	Зовнішній сигнал, наростаючий фронт	Зовнішній сигнал, обидва фронти
1	1	0	Зовнішній сигнал, спадаючий фронт	Зовнішній сигнал, обидва фронти
1	0	1	Зовнішній сигнал, наростаючий фронт	Програмний строб (USITC)
1	1	1	Зовнішній сигнал, спадаючий фронт	Програмний строб (USITC)

Формат регістра стану USISR наведений на рисунку 14, а опис його бітів наведено в таблиці 12.

	7	6	5	4	3	2	1	0
	USISIF	USIOIF	USIPF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0
Читання(R)/Запис(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Початкове положення	0	0	0	0	0	0	0	0

ATmega165x  
ATmega325x/3250x  
ATmega645x/6450x

Рисунок 14 - Формат регістра USISR

Таблиця 12 – Біти регістра USISR

Біт	Назва	Опис
7	USISIF	<u>Прапорець переривання за виявленням стану СТАРТ</u> В двопровідному режимі цей прапорець встановлюється в 1 при виявленні на шині стану СТАРТ. В інших режимах прапорець встановлюється при інкрементуванні 4-бітного лічильника. Якщо біт USISIE регістра USICR та прапорець I регістра SREG встановлені, то при встановленні даного прапорця генерується переривання (це переривання виводить мікроконтролер з будь якого «сплячого» режиму). Прапорець скидається програмно записом у нього лог. 1. Одночасно зі скиданням прапорця звільняється лінія SCL
6	USIOIF	<u>Прапорець переривання за переповненням лічильника</u> Цей прапорець встановлюється в 1 при переповненні 4-бітного лічильника (при переході від 15 до 0). Якщо біт USIOIE регістра USICR та прапорець I регістра SREG встановлені, то при встановленні даного прапорця генерується переривання, яке може використовуватись для виводу мікроконтролера зі «сплячого» режиму Idle. Прапорець скидається програмно записом у нього лог. 1. Одночасно зі скиданням прапорця звільняється лінія SCL
5	USIPF	<u>Прапорець виявлення стану СТОП</u> У двопровідному режимі цей прапорець встановлюється в 1 при виявленні на шині стану СТОП. Прапорець скидається програмно записом у нього лог. 1
4	USIDC	<u>Прапорець колізії при виведенні даних</u> Цей прапорець встановлюється в 1, якщо значення 7-го біта регістра зсуву відмінне від стану виводу мікроконтролера (дійсно тільки у двопровідному режимі). Даний прапорець використовується для реалізації арбітражу
3	USICNT3	<u>4-бітний лічильник</u> Ці біти відображають поточне значення лічильника. Це значення можна зчитувати та міняти в будь-який момент часу.
2	USICNT2	Інкрементування лічильника проводиться за кожним імпульсом, який формується або апаратно (детектором фронтів зовнішнього сигналу або виходом блоку співпадіння таймера/лічильника T0), або програмно (за допомогою бітів USICLK або USITC регістра USICR).
1	USICNT1	
0	USICNT0	Джерело тактового сигналу визначається станом бітів USICS1:0 та USICLK регістру керування USICR ( таблиця 11)

### 5.3 Режими роботи універсального послідовного інтерфейсу

Як вже було сказано, модуль USI має два основних режими роботи:

– трьохпроводний режим;

– двопровідний режим.

Окрім того, окремі вузли модуля можуть використовуватись також для інших цілей, не обов'язково пов'язаних з передачею даних.

### **5.3.1 Трьохпровідний режим**

У трьохпровідному режимі модуль USI може використовуватись для обміну шиною SPI в режимах 0 або 1. Принципи роботи цієї шини були детально розглянуті в основній частині цієї лекції. Основна відмінність модуля USI від стандартного модуля SPI полягає у відсутності в модулі USI входу  $\overline{SS}$  для вибору веденого пристрою. Однак при необхідності підтримка цього сигналу може бути легко реалізована програмно. Структура модулів USI при роботі в трьохпровідному режимі наведена на рисунку 15.

Регістри зсуву обох модулів (ведучого та веденого) з'єднуються між собою таким чином, щоб після восьми імпульсів на лінії USCK вміст регістру зсуву ведучого опинився в регістрі зсуву веденого, та навпаки. По цьому ж тактовому сигналу інкрементується 4-бітний лічильник модуля USI. Відповідно, для визначення моменту завершення обміну можна використовувати прапорець переривання за переповненням лічильника (USIOF). Тактовий сигнал формується ведучим пристроєм. Стан виводу USCK змінюється переключенням біта PORTE4, або ж за допомогою запису лог. 1 в біт USITC регістра USICR.

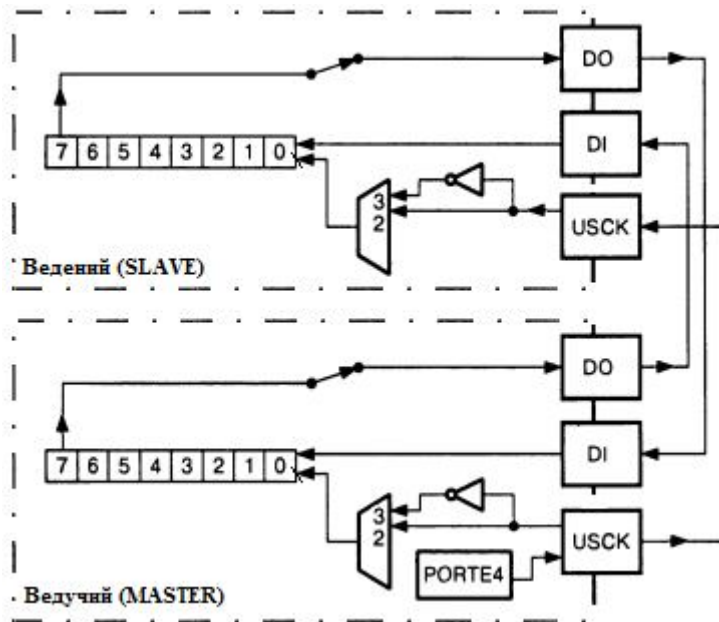


Рисунок 15 - Модуль USI в трьохпровідному режимі

Часові діаграми обміну в трьохпровідному режимі представлені на рисунку 16.

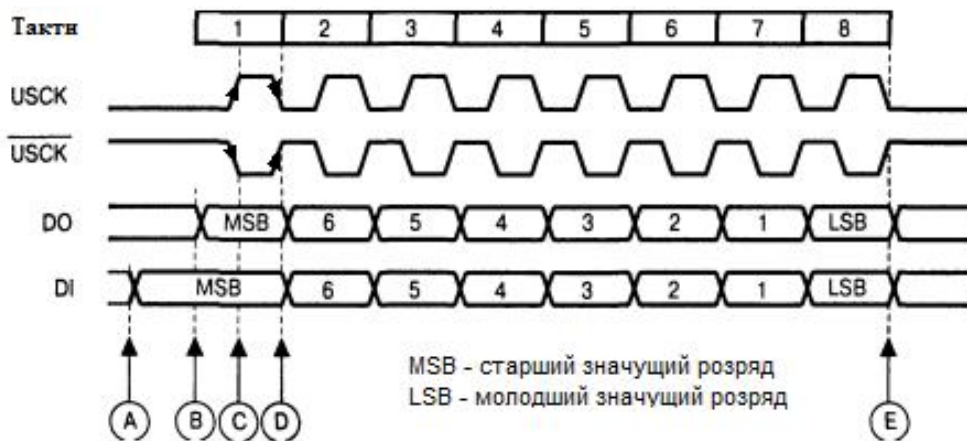


Рисунок 16 – Часові діаграми обміну в трьохпровідному режимі

Верхній графік сигналу USCK відповідає 0-му режиму зовнішнього тактового сигналу ( $USICS0 = 0$ ). В цьому режимі дані, що поступають на вхід DI, зчитуються за наростаючим фронтом сигналу USCK, а зміна стану виходу DO (зсув вмісту регістра зсуву) виконується за спадаючим

фронтом. Нижній графік сигналу USCK відповідає 1-му режиму зовнішнього тактового сигналу ( $USICSO = 1$ ). В цьому режимі призначення фронтів тактового сигналу змінюється на протилежні: вхідні дані зчитуються за спадаючим фронтом, а вихідні дані виводяться за наростаючим фронтом. Вказані режими відповідають 0-му та 1-му режимам SPI (див. основну частину лекції).

Таким чином, обмін в режимі SPI виконується у декілька етапів:

1) Обидва пристрої готують вихідні дані, вмикають вихідні драйвери та скидають вміст своїх 4-бітних лічильників (точки А та В на рисунку 4.16). Підготовка даних полягає в завантаженні їх в регістр зсуву, а вихідні драйвери вмикаються встановленням відповідного біта регістра напряму передачі даних. Послідовність подій А та В може бути будь-якою, важливо тільки, щоб інтервал між останньою подією та початком зчитування даних був не менше половини періоду сигналу USCK.

2) Ведучий формує імпульси тактового сигналу, програмно змінюючи стан лінії USCK (точки С та D на рисунку 16). Значення біта даних на входах (DI) ведучого та веденого пристрою зчитується за першим фронтом сигналу USCK (точка С), а стан виходу (DO) змінюється за протилежним фронтом (точка D). Інкрементування лічильника проводиться за кожним фронтом.

3) Для передачі байта 2-й етап повторюється 8 раз.

4) Після 8-го імпульсу тактового сигналу відбувається переповнення лічильника, що вказує на завершення обміну, при цьому прийнятий байт знаходиться в регістрі даних USIDR.

На завершення наведемо приклади трьох підпрограм, демонструючих використання модуля USI у трьохпровідному режимі.

Перша підпрограма реалізує передачу одного байта даних ведучим пристроєм. Байт, що передається, міститься в регістрі r16, по закінченні



обміну в цьому ж регістрі знаходиться прийнятий байт. Передбачається, що виводи DO та USCK вже налаштовані на вихід.

### SPITransfer:

```
sts    USIDR,r16          ; Завантажуємо регістр зсуву
ldi    r16,(1<<USIOIF)   ; Скидаємо прапорець переповнення
lei    USUSR,r16         ; та лічильник
SPITransfer_loop:
ldi    r16,(1<<USIWM0) | (1<<USICS1) | (1<<USICLK) | (1<<USITC)
sts    USICR,r16         ; Задаємо режим та переключаємо USCK
lds    r16,USISR
sbrs   r16,USIOIF        ; Перевіряємо прапорець переповнення
rjmp   SPITransfer_loop
lds    r16,USIDR         ; Запам'ятовуємо прийнятий біт
ret
```

Друга підпрограма виконує ті ж самі функції, але при цьому забезпечується максимально можлива швидкість передачі ( $f_{SCK} = f_{CLK}/4$ ).

### SPITransfer\_Fast:

```
sts    USIDR,r16
ldi    r16,(1<<USIWM0) | (1<<USICS0) | (1<<USITC)
ldi    r17,(1<<USIWM0) | (1<<USICS1) | (1<<USITC) | (1<<USICLK)
sts    USICR,r16         ; MSB
sts    USICR,r17
sts    USICR,r16
sts    USICR,r17
sts    USICR,r16
sts    USICR,r17
sts    USICR,r16
```

```

sts    USICR,r17
sts    USICR,r16
sts    USICR,r17
sts    USICR,r16
sts    USICR,r17
sts    USICR,r16
sts    USICR,r17
sts    USICR,r16    ;LSB
sts    USICR,r17
lds    r16,USIDR
ret

```

Третя підпрограма демонструє роботу модуля USI в якості веденого SPI-пристрою:

Init:

```

ldi    r16,(1<<USIWM0) | (1<<USICS1)
sts    USUCR,r16    ; Виконується тільки один раз
...

```

SlaveSPITransfer:

```

sts    USIDR,r16
ldi    r16,(1<<USIOIF)
sts    USISR,r16

```

SlaveSPITransfer\_loop:

```

lds    r16,USISR
sbrs  r16,USIOIF
rjmp  SlaveSPITransfer_loop
lds    r16,USIDR
ret

```

Як і в першій підпрограмі, передбачається, що виводи DO та USCK вже налаштовані на вихід. Байт, що передається, міститься в регістрі r16, по закінченні обміну в цьому ж регістрі знаходиться байт, посланий ведучим.

### **5.3.2 Двопровідний режим**

У двопровідному режимі модуль USI може використовуватися для обміну шиною TWI (I<sup>2</sup>C), принципи роботи та протокол якої були детально розглянуті у розділі 3. На відміну від стандартних приймачів-передавачів TWI, в модулі USI відсутнє обмеження швидкості наростання вихідних та фільтрація вхідних сигналів. Структура модулів USI при роботі у двопровідному режимі наведена на рисунку 17.

Головна відмінність між ведучим та веденим пристроями полягає в тому, що тактовий сигнал задіяний тільки у ведучому. Формування імпульсів тактового сигналу здійснюється програмною зміною біта PORTE4 регістра PORTE, в той час як зсув вмісту регістра зсуву здійснюється апаратно в обох пристроях (в двопровідному режимі ця операція виконується тільки за спадаючим фронтом тактового сигналу). Оскільки ведений пристрій може формувати цикли очікування, утримуючи на лінії SCL НИЗЬКИЙ рівень, ведучий пристрій завжди повинен контролювати стан лінії SCL після формування наростаючого фронту. Завершення циклу обміну можна визначити за переповненням 4-бітного лічильника.

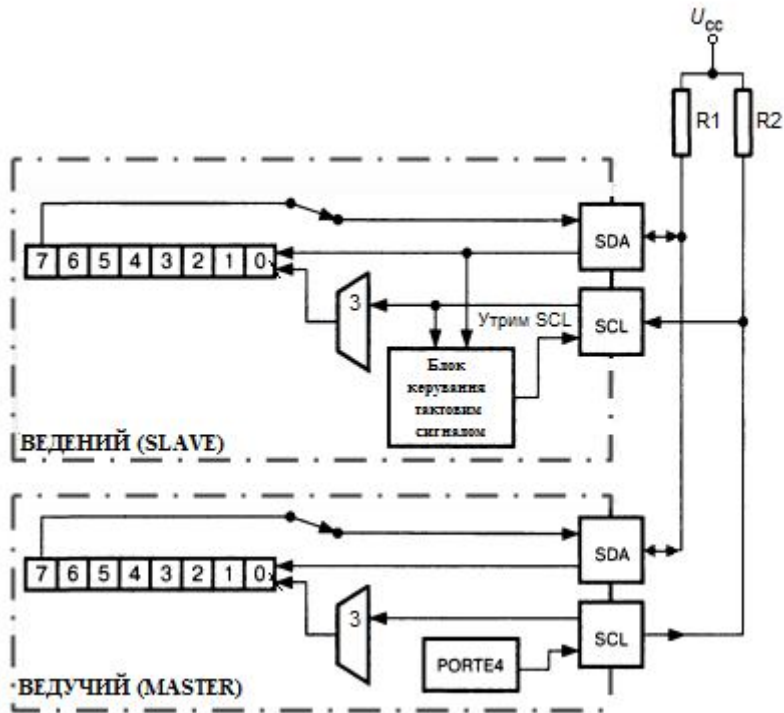


Рисунок 17 – Модулі USI в двопровідному режимі

Зверніть увагу, що модулі USI реалізують обмін тільки на фізичному рівні, а для формування потоку даних необхідно використовувати протокол, подібний розглянутому в [2].

Часові діаграми обміну в двопровідному режимі представлені на рисунку 18.

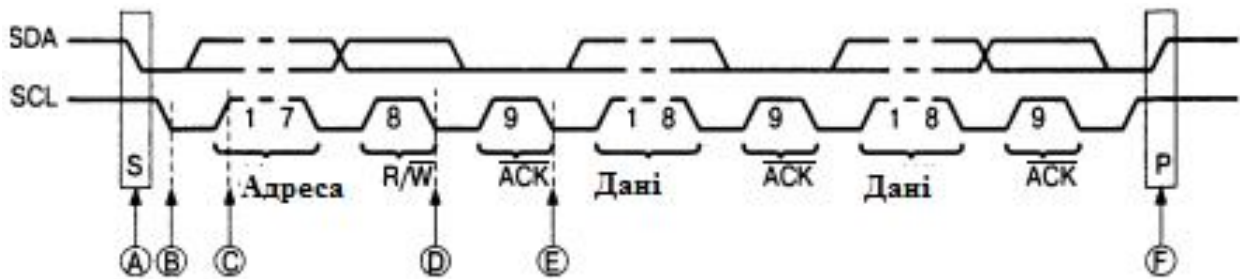


Рисунок 18 – Часові діаграми обміну в двопровідному режимі

Обмін в режимі TWI виконується в декілька етапів:

1) Ведучий пристрій формує на шині стан СТАРТ, видаючи на лінію SDA НИЗЬКИЙ рівень при ВИСОКОМУ рівні на лінії SCL (точка А на рисунку 18). Видача на лінію SDA НИЗЬКОГО рівня може бути здійснена або записом в регістр зсуву значення зі скинутим 7-м бітом, або скиданням 5-го біта регістра PORTE. Зрозуміло, що біт DDRE7 регістра DDRE при цьому повинен бути встановлений в 1. Цей стан детектується схемою виявлення стану СТАРТ веденого модуля (рисунок 19), в результаті чого встановлюється прапорець USISIF та, при необхідності, генерується переривання.

2) Окрім встановлення прапорця, схема виявлення стану СТАРТ утримує на лінії SCL НИЗЬКИЙ рівень після формування на ній ведучим спадаючого фронту (точка В). Це дозволяє виконати веденому вихід зі «сплячого» режиму або завершити виконання інших операцій перед ініціалізацією регістра зсуву для прийому адреси, яка полягає у скиданні прапорця USISIF та лічильника.

3) Ведучий видає на лінію SDA перший біт байта та звільняє лінію SCL, формуючи на ній наростаючий фронт (точка С). За цим фронтом ведений пристрій зчитує значення біта та поміщає його в регістр зсуву.

4) Після передачі восьми бітів (адреса веденого пристрою плюс біт напряму передачі даних) відбувається переповнення лічильника веденого пристрою, в результаті чого останній виставляє на лінію SCL НИЗЬКИЙ рівень (точка D). Якщо адреса, послана ведучим, не співпадає з адресою веденого, то ведений звільняє лінію SCL та переходить до очікування появи на шині нового стану СТАРТ.

5) Якщо ж ведений розпізнає свою адресу, то під час передачі біта підтвердження він посилає на лінію SDA НИЗЬКИЙ рівень, після чого знову утримує на лінії SCL напругу НИЗЬКОГО рівня (тобто перед звільненням лінії SCL в момент D необхідно записати в лічильник значення 14). В залежності від значення біта напряму передачі даних або

ведучий, або ведений пристрій вмикає вихідні буфери. Якщо даний біт встановлений, виконується операція читання (лінією SDA керує ведений пристрій). Після видачі підтвердження ведений пристрій знову може виставити на лінії SCL НИЗЬКИЙ рівень (точка E).

б) Далі виконується пересилання у заданому напрямі потрібного числа байтів даних до тих пір, поки ведучий пристрій не сформує на шині стан СТОП або ПОВСТАРТ.

Якщо з якоїсь причини модуль не може продовжувати прийом даних, то після прийому останнього байта він формує непідтвердження (NACK), видаючи НИЗЬКИЙ рівень на лінію SDA під час передачі біта підтвердження.

На рисунку 19 наведена функціональна схема вузла виявлення стану СТАРТ модуля USI.

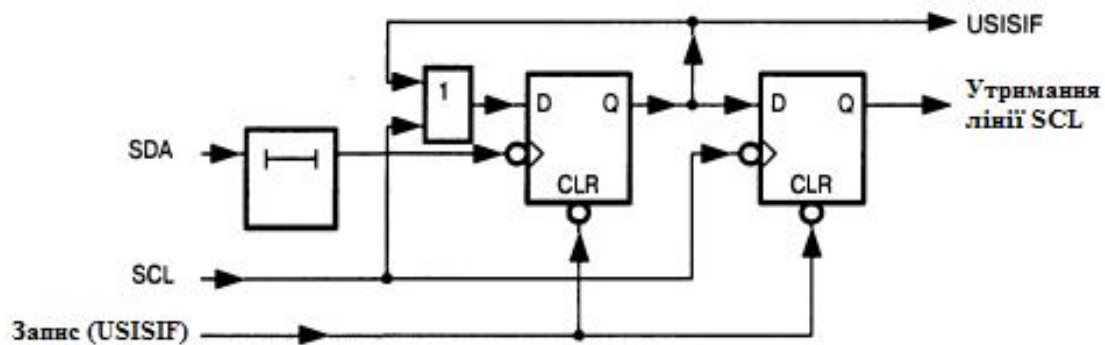


Рисунок 19 – Схема виявлення стану СТАРТ

Оскільки цей вузол є асинхронним, він може використовуватися для виведення мікроконтролера зі «сплячих» режимів, таких як POWERDOWN. Єдине, на що потрібно звернути увагу - час запуску тактового генератора не повинен перевищувати максимально можливу тривалість сигналу НИЗЬКОГО рівня на лінії SCL, якщо ця величина нормується у використовуваному протоколі.

### **5.3.3 Альтернативне використання модуля універсального послідовного інтерфейсу**

Як вже зазначалося, модуль USI може використовуватись не тільки для організації обміну по дво - або трьохпровідній лінії. Нижче перелічені варіанти альтернативного використання вузлів модуля USI.

#### Напівдуплексний асинхронний приймач-передавач

Використовуючи регістр зсуву модуля в трьохпровідному режимі, можна реалізувати додатковий модуль UART, більш компактний та швидкодіючий, ніж при чисто програмній його реалізації.

#### 4-бітний лічильник

Вбудований 4-бітний лічильник модуля USI можна використовувати як автономний лічильник з можливістю генерації переривання за переповненням.

#### 12-бітний таймер/лічильник

Об'єднавши 4-бітний лічильник модуля USI з таймером/лічильником T0, отримаємо 12-бітний лічильник.

#### Формування зовнішнього переривання за фронтом

При записі в лічильник максимального значення (\$0F) він може використовуватись для генерації зовнішнього переривання. Цей режим роботи задається встановленням біта USICS1 регістра USICR в 1.

#### Формування програмного переривання

Переривання за переповненням 4-бітного лічильника можна використовувати в якості програмного переривання, яке формується стробом тактового сигналу.