

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА АВТОМАТИКИ ТА УПРАВЛІННЯ У ТЕХНІЧНИХ СИСТЕМАХ

Проектування мікропроцесорних систем:

Проектування мікропроцесорних систем на базі AVR-мікроконтролерів:

Периферійні модулі AVR-мікроконтролерів: Навчальний посібник
для студентів напряму підготовки 6.050201 «Системна інженерія»

Київ НТУУ “КПІ” 2012

Проектування мікропроцесорних систем: Проектування мікропроцесорних систем на базі AVR-мікроконтролерів: Периферійні модулі AVR-мікроконтролерів: Навчальний посібник для студентів напряму підготовки 6.050201 «Системна інженерія» кафедри Автоматики та управління у технічних системах / Укл.: А.О. Новацький– К: НТУУ „КПІ”, 2012– 470с.

Укладач: А.О. Новацький
Відповідальний за випуск: Л.Ю. Юрчук
Рецензенти: А.Ю.Дорошенко, О.А.Чемерис

ЗМІСТ

АНОТАЦІЯ.....	14
ВСТУП.....	15
СПИСОК СКОРОЧЕНЬ.....	21
1 ПОРТИ ВВЕДЕННЯ/ВИВЕДЕННЯ.....	25
1.1 ПРИЗНАЧЕННЯ ТА МІСЦЕ ПРИСТРОЇВ ВВЕДЕННЯ/ВИВЕДЕННЯ ДАНИХ У МІКРОПРОЦЕСОРНИХ СИСТЕМАХ	25
1.2 ПАРАЛЕЛЬНЕ ТА ПОСЛІДОВНЕ ВВЕДЕННЯ/ВИВЕДЕННЯ	26
1.3 ПОРТИ ВВЕДЕННЯ/ВИВЕДЕННЯ AVR-МІКРОКОНТРОЛЕРІВ	27
1.3.1 ЗАГАЛЬНІ ВІДОМОСТІ	27
1.3.2 ЗВЕРНЕННЯ ДО ПОРТІВ ВВЕДЕННЯ/ВИВЕДЕННЯ.....	29
1.3.3 КОНФІГУРУВАННЯ ПОРТІВ ВВЕДЕННЯ/ВИВЕДЕННЯ.....	33
1.4 АПАРАТНИЙ МОДУЛЯТОР.....	41
2 ПОСЛІДОВНИЙ СИНХРОННО/АСИНХРОННИЙ ІНТЕРФЕЙС	48
2.1 ЗАГАЛЬНІ ВІДОМОСТІ.....	48
2.2 ВИКОРИСТАННЯ МОДУЛІВ USART/UART	51
2.2.1 ОПИС СТРУКТУРИ МОДУЛЯ.....	51
2.2.2 ШВИДКІСТЬ ПРИЙОМУ/ПЕРЕДАЧІ	60
2.2.3 ФОРМАТ КАДРУ	67
2.2.4 ПЕРЕДАЧА ДАНИХ.....	70
2.2.5 ПРИЙОМ ДАНИХ	73

2.3 МУЛЬТИПРОЦЕСОРНИЙ РЕЖИМ РОБОТИ.....	80
2.4 РОБОТА МОДУЛЯ USART В РЕЖИМІ MSPi	82
2.4.1 ЗАГАЛЬНІ ВІДОМОСТІ	82
2.4.2 КЕРУВАННЯ МОДУЛЕМ USART В РЕЖИМІ MSPi.....	82
2.4.3 ІНІЦІАЛІЗАЦІЯ РЕЖИМУ MSPi	87
2.4.4 ПЕРЕДАЧА ДАНИХ В РЕЖИМІ MSPi.....	89
2.5 ФОРМУВАННЯ СИНХРОЧАСТОТ ДЛЯ ПЕРЕДАВАЧА ТА ПРИЙМАЧА ПОСЛІДОВНОГО ПОРТУ	91
2.5.1 МІКРОКОНТРОЛЕРИ СІМЕЙСТВА MCS-51	91
2.5.2 РОЗРАХУНОК ШВИДКОСТІ ПЕРЕДАЧІ ІНФОРМАЦІЇ ТА ТРИВАЛОСТІ ОДНОГО БІТА ДЛЯ МОДУЛЯ USART МІКРОКОНТРОЛЕРІВ AVR.....	95
3 ПОСЛІДОВНИЙ СИНХРОННИЙ ІНТЕРФЕЙС TWI (I ² C)	99
3.1 ХАРАКТЕРИСТИКА ІНТЕРФЕЙСУ TWI (I ² C) НА АПАРАТНОМУ РІВНІ.....	99
3.1.1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ІНТЕРФЕЙСУ I ² C	99
3.1.2 АРХІТЕКТУРА ІНТЕРФЕЙСУ TWI МІКРОКОНТРОЛЕРІВ СІМЕЙСТВА AVR НА АПАРАТНОМУ РІВНІ.....	105
3.1.2.1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА.....	105
3.1.2.2 ПРИНЦИПИ ОБМІНУ ДАНИМИ ШИНОЮ TWI	107
3.1.2.2.1 ЗАГАЛЬНІ ВІДОМОСТІ.....	107
3.1.2.2.2 ФОРМАТ АДРЕСНОГО ПАКЕТА.....	110
3.1.2.2.3 ФОРМАТ ПАКЕТА ДАНИХ	112

3.1.3 ОПИС ТИПОВОГО МОДУЛЯ TWI.....	114
3.1.3.1 СТРУКТУРА МОДУЛЯ.....	114
3.1.3.2 ВИВОДИ SCL I SDA	115
3.1.3.3 КОНТРОЛЕР ШВИДКОСТІ ПЕРЕДАЧІ (BIT RATE GENERATOR).....	116
3.1.3.4 БЛОК ШИННОГО ІНТЕРФЕЙСУ(BIT INTERFACE UNIT).....	118
3.1.3.5 БЛОК КОНТРОЛЮ АДРЕСИ(ADDRESS MATCH UNIT).....	119
3.1.3.6 БЛОК КЕРУВАННЯ (CONTROL UNIT)	121
3.1.4 ВЗАЄМОДІЯ ПРИКЛАДНОЇ ПРОГРАМИ З ІНТЕРФЕЙСОМ TWI..	123
3.1.5 МІКРОКОНТРОЛЕРИ AVR, ЯКІ МАЮТЬ ІНТЕРФЕЙС TWI.....	128
3.1.6 ПРИСТРОЇ, ЯКІ ПІДТРИМУЮТЬ ІНТЕРФЕЙС TWI.....	129
3.2 ПРОГРАМУВАННЯ ІНТЕРФЕЙСУ TWI	130
3.2.1 РЕЖИМИ РОБОТИ МОДУЛЯ TWI.....	130
3.2.1.1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА	130
3.2.1.2 РЕЖИМ «ВЕДУЧИЙ ПЕРЕДАВАЧ»	131
3.2.1.3 РЕЖИМ «ВЕДУЧИЙ ПРИЙМАЧ»	137
3.2.1.4 РЕЖИМ «ВЕДЕНИЙ ПРИЙМАЧ»	142
3.2.1.5 РЕЖИМ «ВЕДЕНИЙ ПЕРЕДАВАЧ».....	147
3.2.1.6 КОМБІНУВАННЯ РІЗНИХ РЕЖИМІВ	152
3.2.1.7 АРБІТРАЖ.....	153
3.2.1.8 ПАРАМЕТРИ ІНТЕРФЕЙСУ	154

3.2.2 СХЕМИ АЛГОРИТМІВ РОБОТИ МОДУЛЯ TWI.....	155
3.2.3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ РОБОТИ МОДУЛЯ TWI.....	163
3.2.3.1 ПРОГРАМА РОБОТИ У РЕЖИМІ «ВЕДУЧИЙ ПЕРЕДАВАЧ» ..	163
3.2.3.2 ПРОГРАМА РОБОТИ У РЕЖИМІ «ВЕДУЧИЙ ПРИЙМАЧ»	164
3.2.3.3 ПРОГРАМА РОБОТИ У РЕЖИМІ «ВЕДЕНИЙ ПРИЙМАЧ»	166
3.2.3.4 ПРОГРАМА РОБОТИ У РЕЖИМІ «ВЕДЕНИЙ ПЕРЕДАВАЧ» ..	168
4 ПОСЛІДОВНИЙ ПЕРИФЕРІЙНИЙ ІНТЕРФЕЙС SPI.....	172
4.1 ЗАГАЛЬНІ ВІДОМОСТІ	172
4.2 ВИКОРИСТАННЯ МОДУЛЯ SPI.....	173
4.2.1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА АРХІТЕКТУРИ МОДУЛЯ SPI.	173
4.2.2 РЕЖИМИ ПЕРЕДАЧІ ДАНИХ	181
4.2.3 ВИКОРИСТАННЯ ВИВОДУ $\overline{SS}$	186
4.3 РЕЖИМ ПРОГРАМУВАННЯ ПАМ'ЯТІ ПОСЛІДОВНИМ КАНАЛОМ	187
4.4 ПЕРИФЕРІЙНІ ПРИСТРОЇ З SPI-ІНТЕРФЕЙСОМ	190
4.5 УНІВЕРСАЛЬНИЙ ПОСЛІДОВНИЙ ІНТЕРФЕЙС USI	193
4.5.1 ЗАГАЛЬНІ ВІДОМОСТІ	193
4.5.2 ВИКОРИСТАННЯ МОДУЛЯ УНІВЕРСАЛЬНОГО ПОСЛІДОВНОГО ІНТЕРФЕЙСУ	195
4.5.3 РЕЖИМИ РОБОТИ УНІВЕРСАЛЬНОГО ПОСЛІДОВНОГО ІНТЕРФЕЙСУ	200
4.5.3.1 ТРЬОХПРОВІДНИЙ РЕЖИМ.....	201

4.5.3.2 ДВОПРОВІДНИЙ РЕЖИМ	206
4.5.3.3 АЛЬТЕРНАТИВНЕ ВИКОРИСТАННЯ МОДУЛЯ УНІВЕРСАЛЬНОГО ПОСЛІДОВНОГО ІНТЕРФЕЙСУ	210
5 ТАЙМЕРИ	213
5.1 ВОСЬМИРОЗРЯДНІ ТАЙМЕРИ.МОДУЛЯТОР. ВАРТОВИЙ ТАЙМЕР	213
5.1.1 ОСОБЛИВОСТІ АРХІТЕКТУРИ ТАЙМЕРІВ AVR- МІКРОКОНТРОЛЕРІВ.....	213
5.1.1.1 ЗАГАЛЬНІ ВІДОМОСТІ.....	213
5.1.1.2 ПРИЗНАЧЕННЯ ВИВОДІВ ТАЙМЕРІВ/ЛІЧИЛЬНИКІВ	215
5.1.1.3 ПЕРЕРИВАННЯ ВІД ТАЙМЕРІВ/ЛІЧИЛЬНИКІВ	215
5.1.1.4 ПОПЕРЕДНІ ДІЛЬНИКИ ТАЙМЕРІВ/ЛІЧИЛЬНИКІВ	223
5.1.1.4.1 КЕРУВАННЯ ПОПЕРЕДНІМИ ДІЛЬНИКАМИ	226
5.1.1.4.2 ВИКОРИСТАННЯ ЗОВНІШНЬОГО ТАКТОВОГО СИГНАЛУ	227
5.2 ВОСЬМИРОЗРЯДНІ ТАЙМЕРИ/ЛІЧИЛЬНИКИ	228
5.2.1 РЕЖИМИ РОБОТИ	238
5.2.1.1 РЕЖИМ NORMAL	239
5.2.1.2 РЕЖИМ СТС (СКИДАННЯ ЗА ЗБІГОМ/СНОР ON TIMER CONCIDENTENCE).....	241
5.2.1.3 РЕЖИМ «ШВИДКОДІЮЧИЙ ШІМ» (FAST PWM).....	243
5.2.1.4 РЕЖИМ «ШІМ З КОРЕКЦІЄЮ ФАЗИ І ЧАСТОТИ» (PHASE CORRECT PWM)	246
5.2.2 АСИНХРОННИЙ РЕЖИМ	249

5.3 МОДУЛЯТОР	255
5.4 16 - РОЗРЯДНІ ТАЙМЕРИ/ЛІЧИЛЬНИКИ	258
5.4.1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА	258
5.4.2 ЗВЕРНЕННЯ ДО 16-РОЗРЯДНИХ РЕГІСТРІВ	269
5.4.3 КЕРУВАННЯ ТАКТОВИМ СИГНАЛОМ	270
5.4.4 РЕЖИМИ РОБОТИ	271
5.4.4.1 РЕЖИМ NORMAL	272
5.4.4.2 РЕЖИМ «СКИДАННЯ ЗА ЗБІГОМ»(SNOP ON TIME CONFIDENCE)	274
5.4.4.3 РЕЖИМ «ШВИДКОДІЮЧИЙ ШІМ»(FAST PWM).....	276
5.4.4.4 РЕЖИМ «ШІМ З КОРЕКЦІЄЮ ФАЗИ» (PHASE CORRECT PWM)	280
5.4.4.5 РЕЖИМ «ШІМ З КОРЕКЦІЄЮ ФАЗИ І ЧАСТОТИ»(PHASE AND FREQUENCY CORRECT PWM)	284
5.5 ВАРТОВИЙ ТАЙМЕР	287
5.5.1 СТАНДАРТНИЙ ВАРТОВИЙ ТАЙМЕР	287
5.5.2 РОЗШИРЕНИЙ ВАРТОВИЙ ТАЙМЕР.....	292
6 СИСТЕМА ПЕРЕРИВАНЬ	298
6.1 СИСТЕМА ПЕРЕРИВАНЬ AVR-МІКРОКОНТРОЛЕРІВ.....	298
6.1.1 ЗАГАЛЬНІ ВІДОМОСТІ	298
6.1.2 ТАБЛИЦЯ ВЕКТОРІВ ПЕРЕРИВАНЬ	298
6.1.3 ОБРОБКА ПЕРЕРИВАНЬ	303
6.1.4 ЗОВНІШНІ ПЕРЕРИВАННЯ	308

6.1.5 ОСОБЛИВОСТІ ВИКОРИСТАННЯ МОДУЛЯ ПЕРЕРИВАНЬ В МІКРОКОНТРОЛЕРАХ ХМЕГА	322
6.1.5.1 ЗАГАЛЬНІ ВІДОМОСТІ.....	322
6.1.5.2 ПРИНЦИП ДІЇ	323
6.1.5.3 ЗАТРИМКА РЕАГУВАННЯ НА ПЕРЕРИВАННЯ	325
6.1.5.4 РІВНІ ПЕРЕРИВАНЬ	326
6.1.5.5 ПРІОРИТЕТИ ПЕРЕРИВАНЬ.....	327
6.1.5.5.1 СТАТИЧНЕ ЗАДАННЯ ПРІОРИТЕТІВ	327
6.1.5.5.2 ДИНАМІЧНЕ ЗАДАННЯ ПРІОРИТЕТІВ	328
6.1.5.6 ОПИС КЕРУЮЧИХ РЕГІСТРІВ	329
6.1.5.6.1 РЕГІСТР СТАТУСУ(STATUS-РЕГІСТР)	329
6.1.5.6.2 РЕГІСТР ПРІОРИТЕТУ ПЕРЕРИВАНЬ(INTPRI-РЕГІСТР)	330
6.1.5.6.2 РЕГІСТР КЕРУВАННЯ(CTRL-РЕГІСТР).....	331
6.1.5.7 ТАБЛИЦЯ ВЕКТОРІВ ПЕРЕРИВАНЬ МІКРОКОНТРОЛЕРА ХМЕГА А3.....	332
6.1.5.8 КІЛЬКІСТЬ ПЕРЕРИВАНЬ В МІКРОКОНТРОЛЕРАХ ХМЕГА СІМЕЙСТВ А1, А2, А3.....	334
7 АНАЛОГОВИЙ КОМПАРАТОР	336
7.1 ЗАГАЛЬНІ ВІДОМОСТІ	336
7.2 АНАЛОГОВІ КОМПАРАТОРИ НА ІНТЕГРАЛЬНІЙ МІКРОСХЕМІ ОПЕРАЦІЙНОГО ПІДСИЛЮВАЧА.....	337
7.2.1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ІНТЕГРАЛЬНОЇ МІКРОСХЕМИ ОПЕРАЦІЙНОГО ПІДСИЛЮВАЧА.....	337

7.2.2 АНАЛОГОВИЙ КОМПАРАТОРИ ДЛЯ ПОРІВНЯННЯ ОДНОПОЛЯРНИХ НАПРУГ НА ІНТЕГРАЛЬНІЙ МІКРОСХЕМІ ОПЕРАЦІЙНОГО ПІДСИЛЮВАЧА	343
7.2.3 АНАЛОГОВИЙ КОМПАРАТОР ДЛЯ ПОРІВНЯННЯ РІЗНОПОЛЯРНИХ НАПРУГ НА ІНТЕГРАЛЬНІЙ МІКРОСХЕМІ ОПЕРАЦІЙНОГО ПІДСИЛЮВАЧА	345
7.3 АНАЛОГОВИЙ КОМПАРАТОР У СКЛАДІ AVR- МІКРОКОНТРОЛЕРА	346
7.3.1 ЗАГАЛЬНІ ВІДОМОСТІ	346
7.3.2 ФУНКЦІОНУВАННЯ КОМПАРАТОРА	348
8 АНАЛОГО-ЦИФРОВИЙ ПЕРЕТВОРЮВАЧ (АЦП)	355
8.1 ЗАГАЛЬНІ ОСОБЛИВОСТІ МОДУЛЯ АЦП	355
8.2 ФУНКЦІОНУВАННЯ МОДУЛЯ АЦП	356
8.3 ПРОГРАМУВАННЯ МОДУЛЯ АЦП	358
8.4 ФОРМУВАННЯ ТАКТОВОЇ ЧАСТОТИ АЦП	365
8.5 ЧАСОВІ ДІАГРАМИ РОБОТИ АЦП	367
8.6 КЕРУВАННЯ ВХІДНИМ МУЛЬТИПЛЕКСОРОМ	372
8.7 ЗБЕРЕЖЕННЯ РЕЗУЛЬТАТУ ПЕРЕТВОРЕННЯ	378
8.8 РЕКОМЕНДАЦІЇ З ПЕРЕМІКАННЯ ВХІДНИХ КАНАЛІВ	380
8.9 ОСОБЛИВОСТІ ПІДКЛЮЧЕННЯ ДЖЕРЕЛА ОПОРНОЇ НАПРУГИ	381
8.10 ОСОБЛИВОСТІ СХЕМИ АНАЛОГОВОГО ВХОДУ	381
8.11 РЕЗУЛЬТАТ ПЕРЕТВОРЕННЯ АЦП	383
8.12 ПІДВИЩЕННЯ ТОЧНОСТІ ПЕРЕТВОРЕННЯ	386

8.12.1 ЗАГАЛЬНІ РЕКОМЕНДАЦІЇ	386
8.12.2 ПРИДУШУВАЧ ШУМІВ	388
8.12.3 КОМПЕНСАЦІЯ ЗСУВУ	389
8.12.4 ВИЗНАЧЕННЯ ПОХИБОК АНАЛОГО-ЦИФРОВОГО ПЕРЕТВОРЕННЯ.....	389
8.13 ОСНОВНІ ПАРАМЕТРИ РОБОТИ АЦП	392
9 СПЕЦІАЛЬНІ РЕЖИМИ РОБОТИ МІКРОКОНТРОЛЕРА	396
9.1 ТАКТУВАННЯ.....	396
9.1.1 ЗАГАЛЬНІ ВІДОМОСТІ	396
9.1.2 ТАКТОВИЙ ГЕНЕРАТОР	396
9.1.2.1 ГЕНЕРАТОР З ЗОВНІШНІМ РЕЗОНАТОРОМ.....	399
9.1.2.2 НИЗЬКОЧАСТОТНИЙ КВАРЦОВИЙ ГЕНЕРАТОР	403
9.1.2.3 ЗОВНІШНІЙ СИГНАЛ СИНХРОНІЗАЦІЇ	403
9.1.2.4 ГЕНЕРАТОР З ЗОВНІШНІМ RC-ЛАНЦЮГОМ	404
9.1.2.5 ВНУТРІШНІЙ КАЛІБРОВАНИЙ RC-ГЕНЕРАТОР	405
9.1.2.6 ВНУТРІШНІЙ RC-ГЕНЕРАТОР НА 128 КГЦ.....	408
9.1.2.7 КЕРУВАННЯ ТАКТОВОЮ ЧАСТОТОЮ	409
9.2 РЕЖИМ ЗНИЖЕНОГО ЕНЕРГОСПОЖИВАННЯ	412
9.2.1 ЗАГАЛЬНІ ВІДОМОСТІ	412
9.2.2 КЕРУВАННЯ РЕЖИМАМИ ЗНИЖЕНОГО ЕНЕРГОСПОЖИВАННЯ(СПЛЯЧИМИ РЕЖИМАМИ).....	412
9.2.3 КЕРУВАННЯ ТАКТОВИМИ СИГНАЛАМИ ОКРЕМИХ МОДУЛІВ.....	420

9.2.4 ЗАГАЛЬНІ РЕКОМЕНДАЦІЇ ПО ЗНИЖЕННЮ ЕНЕРГОСПОЖИВАННЯ	424
9.3 СКИДАННЯ	427
9.3.1 ЗАГАЛЬНІ ВІДОМОСТІ	427
9.3.2 ОПИС ОКРЕМИХ ВИДІВ СКИДАННЯ.....	431
9.3.2.1 СКИДАННЯ ПО ВВІМКНЕННЮ ЖИВЛЕННЯ.....	431
9.3.2.2 АПАРАТНЕ СКИДАННЯ.....	433
9.3.2.3 СКИДАННЯ ВІД ВАРТОВОГО ТАЙМЕРА	433
9.3.2.4 СКИДАННЯ ПРИ ЗНИЖЕННІ НАПРУГИ ЖИВЛЕННЯ	434
10 ЦИФРО-АНАЛОГОВИЙ ПЕРЕТВОРЮВАЧ	439
10.1 ХАРАКТЕРИСТИКА МОДУЛЯ ЦАП У СКЛАДІ МІКРОКОНТРОЛЕРІВ ХМЕГА.....	439
10.1.1 ЗАГАЛЬНІ ВІДОМОСТІ.....	439
10.1.2 ПРИНЦИП РОБОТИ	439
10.1.3 ДЖЕРЕЛА ОПОРНОЇ НАПРУГИ.....	441
10.1.4 ВИХІДНІ КАНАЛИ	441
10.1.5 РЕЖИМИ РОБОТИ.....	442
10.1.5.1 ОДНОКАНАЛЬНИЙ РЕЖИМ РОБОТИ	442
10.1.5.2 ДВОКАНАЛЬНИЙ РЕЖИМ РОБОТИ	443
10.1.6 ТАКТУВАННЯ МОДУЛЯ ЦАП	446
10.1.7 ОБМЕЖЕННЯ ЧАСОВИХ ХАРАКТЕРИСТИК	447
10.1.7.1 РЕЖИМ ЕНЕРГОЗБЕРЕЖЕННЯ	449

10.1.7.2 СИСТЕМА ПОДІЙ	449
10.1.8 ОПИС РЕГІСТРІВ ДАНИХ, СТАТУСУ ТА КЕРУВАННЯ	450
10.1.8.1 РЕГІСТРИ ДАНИХ	450
10.1.8.2 РЕГІСТРИ КЕРУВАННЯ	451
10.1.8.3 РЕГІСТР КЕРУВАННЯ ПОДІЯМИ ЦАП	454
10.1.8.4 РЕГІСТР КЕРУВАННЯ ЧАСОВИМИ ІНТЕРВАЛАМИ ЦАП....	454
10.1.8.5 РЕГІСТР СТАТУСУ ЦАП	455
10.2 КАЛІБРУВАННЯ	457
10.3 ОПИС РОБОТИ І РОЗРАХУНОК ЦИФРО-АНАЛОГОВИХ ПЕРЕТВОРЮВАЧІВ НА ОСНОВІ РЕЗИСТОРНОЇ МАТРИЦІ R-2R З ПІДСУМОВУВАННЯМ НАПРУГ	459
10.3.1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА	459
10.3.2 ПРИНЦИП ДІЇ ЦАП НА МАТРИЦІ R-2R З ПІДСУМОВУВАННЯМ НАПРУГ	460
10.3.3 РОЗРАХУНОК ЦИФРО-АНАЛОГОВИХ ПЕРЕТВОРЮВАЧІВ НА МАТРИЦІ R-2R З ПІДСУМОВУВАННЯМ НАПРУГ	461
ПРЕДМЕТНИЙ ПОКАЖЧИК	465
СПИСОК ЛІТЕРАТУРИ	468

АНОТАЦІЯ

Навчальний посібник орієнтований на підготовку фахівців з напрямку «Системна інженерія». Цей напрям сьогодні використовується в якості інтеграційної основи при організації та здійсненні діяльності по розробці інформаційних систем широкого класу. Для успішного оволодіння майбутніми спеціалістами цим напрямом необхідно засвоїти принципи побудови та використання мікропроцесорних пристроїв, на основі яких розробляються та функціонують мікропроцесорні та обчислювальні системи. Навчальний посібник охоплює теоретичний матеріал та практичні завдання, необхідні для вивчення базової частини дисципліни «Проектування мікропроцесорних систем». Навчальний посібник складається із десяти розділів, в яких розглядаються наступні периферійні модулі: порти введення/виведення, послідовний синхронно/асинхронний інтерфейс, послідовний синхронний інтерфейс TWI (I^2C), послідовний периферійний інтерфейс SPI, послідовний периферійний інтерфейс USI, таймери, система переривань, аналоговий компаратор, аналого-цифровий перетворювач, цифро-аналоговий перетворювач. В кожному розділі розглядаються основні архітектурні особливості периферійних модулів AVR-мікроконтролерів на апаратному та програмному рівнях, наводяться приклади розрахунку та програмування цих модулів.. Тематика посібника відповідає робочій програмі з дисципліни «Проектування мікропроцесорних систем», яка є важливою дисципліною у навчальному плані підготовки бакалаврів з напрямку 6.050201 «Системна інженерія». Робота може бути корисною студентам відповідних спеціальностей при вивченні дисциплін, пов'язаних із проектуванням та використанням мікропроцесорних систем, а також при виконанні бакалаврських робіт, курсових та дипломних проектів, в яких використовуються мікропроцесорні пристрої.

ВСТУП

AVR-мікроконтролери, мабуть, один із самих цікавих напрямів, що розвиваються корпорацією Atmel. Об'єми продажів AVR у світі подвоюються щорічно, неухильно росте число сторонніх фірм, що випускають програмні й апаратні засоби підтримки розробок для цих мікроконтролерів. Можна вважати, що AVR поступово стає ще одним індустріальним стандартом серед 8-розрядних мікроконтролерів загального призначення. В даний час у виробництві в Atmel Corp. знаходяться три сімейства AVR - "tiny", "mega", "X Mega", що розрізняються об'ємами масивів Flash-, EEPROM- і SRAM-пам'яті, набором периферійних вузлів і побудовою схеми тактирування.

AVR являє собою 8-розрядний RISC мікроконтролер, що має швидке процесорне ядро, Flash-пам'ять програм-ROM, пам'ять даних-SRAM, порти введення/виведення і велику кількість інтерфейсних схем. Гарвардська архітектура AVR реалізує повний логічний і фізичний поділ не тільки адресних просторів, але й інформаційних шин для звернення до ROM і SRAM. Така побудова вже ближче до структури цифрових сигнальних процесорів і забезпечує істотне підвищення продуктивності. Використання однорівневого конвеєра в AVR також помітно скоротило цикл "вибірка - виконання" команди. Наприклад, у стандартних мікроконтролерів сімейства MCS-51 коротка команда виконується за 12 тактів генератора (1 машинний цикл), протягом якого процесор послідовно зчитує код операції і виконує її. У AVR-мікроконтролерах коротка команда в загальному потоці теж виконується за один машинний цикл, але він складає всього один період тактової частоти. Відміною рисою архітектури AVR є регістровий файл швидкого доступу, що містить 32-байтових регістри загального призначення. Шість регістрів файлу можуть використовуватися як три 16-розрядних показники адреси при непрямій адресації даних (X, Y і Z

Pointers), що істотно підвищує швидкість пересилання даних при роботі прикладної програми.

Flash-пам'ять програм AVR може бути завантажена як за допомогою звичайного програматора, так і за допомогою SPI-інтерфейсу, у тому числі безпосередньо на робочій платі - функція ISP. Останні версії кристалів "mega" та "X Mega" 2001-2002 року мають можливість самопрограмування (функція SPM). Усі AVR мають також блок енергонезалежної пам'яті даних EEPROM, доступний програмі мікроконтролера безпосередньо в ході її виконання. EEPROM звичайно використовується для збереження проміжних даних, констант, таблиць перекодувань, каліброваних коефіцієнтів і т.ін. Ця пам'ять може бути завантажена ззовні як через SPI інтерфейс, так і за допомогою звичайного програматора. Два програмованих біти таємності дозволяють захистити ROM і енергонезалежну пам'ять даних EEPROM від несанкціонованого зчитування. Внутрішня оперативна пам'ять SRAM входить у більшості мікроконтролерів. Для деяких мікроконтролерів можлива організація підключення зовнішньої пам'яті даних об'ємом до 64К.

Внутрішній тактовий генератор AVR може запускатися від зовнішнього генератора або кварцового резонатора, а також від внутрішнього або зовнішнього RC-ланцюга. Усі AVR цілком статичні, їх мінімальна робоча частота нічим не обмежена (аж до покрокового режиму). Деякі мікроконтролери мають додатковий блок PLL для апаратного збільшення основної тактової частоти в 16 разів. Ця частота може служити джерелом для одного з таймерів/лічильників мікроконтролера, що значно підвищує точність його роботи.

Мікроконтролери AVR мають від 1 до 6 таймерів/лічильників загального призначення з розрядністю 8 або 16 біт.

Загальні риси всіх таймерів/лічильників наступні:

- наявність програмованого попереднього дільника вхідної частоти з різними градаціями ділення. Відміною рисою є можливість роботи таймерів/лічильників на основній тактовій частоті мікроконтролера без попереднього її зниження, що помітно підвищує точність генерації часових інтервалів системи;

- незалежне функціонування від режиму роботи процесорного ядра мікроконтролера (тобто вони можуть бути як зчитані, так і завантажені новим значенням у будь-який час);

- можливість роботи або від зовнішнього джерела опорної частоти, або як лічильник зовнішніх подій. Верхній частотний поріг визначений у цьому випадку як половина основної тактової частоти мікроконтролера. Вибір перепаду зовнішнього джерела (фронт або зріз) програмується користувачем;

- наявність різних векторів переривань для подій "переповнення вмісту", "захоплення", "порівняння";

- вартовий таймер у AVR має свій власний RC-генератор, частота якого залежить від величини напруги споживання мікроконтролера і від температури. Вартовий таймер містить окремий програмований попередній дільник вхідної частоти, що дозволяє підлаштовувати часовий інтервал переповнення таймера і скидання мікроконтролера. Даний таймер можна програмно відключати під час роботи мікросхеми, як в активному режимі, так і в кожному з режимів зниженого енергоспоживання. В останньому випадку це приводить до значного зниження споживаного струму;

- в AVR-мікроконтролери вбудована система реального часу (RTC). Таймер/лічильник RTC має окремий попередній дільник, що може бути програмним способом підключений або до джерела основної тактової частоти, або до додаткового асинхронного джерела опорної частоти (кварцовий резонатор або зовнішній синхросигнал). Для цієї мети зарезервовані два виводи мікросхеми. Внутрішній осцилятор,

навантажений на лічильний вхід таймера/лічильника RTC, оптимізований для роботи з зовнішнім "годинниковим" кварцовим резонатором 32,768 кГц;

– порти введення/виведення AVR мають значне число незалежних ліній "Вхід/Вихід". Вихідні драйвери забезпечують струменеву навантажувальну здатність 20 мА на лінію порту (втікаючий струм) при максимальному значенні 40 мА, що дозволяє безпосередньо підключати до мікроконтролера світлодіоди і біполярні транзистори. Архітектура побудови портів введення/виведення AVR із трьома бітами контролю/керування (замість двох, як це зроблено в більшості 8-розрядних мікроконтролерів) дозволяє розроблювачеві цілком контролювати процес введення/виведення, усуває необхідність мати копію вмісту порту в пам'яті для безпеки і підвищує швидкість роботи мікроконтролера при роботі з зовнішніми пристроями. Особливу значимість здобуває дана можливість AVR при реалізації систем, що працюють в умовах зовнішніх електричних завад;

– AVR-мікроконтролери містять послідовні інтерфейси: USART/UART; SPI та I2C (TWI), що також розширює можливості їх застосування для обміну з сучасною периферією та іншими мікроконтролерами та персональними комп'ютерами;

– до складу більшості AVR входить аналоговий компаратор. Він має окремий вектор переривання в загальній системі переривань мікроконтролера. Тип перепаду, що викликає запит на переривання при спрацьовуванні компаратора, може бути запрограмований як фронт, зріз або переключення. Важливою апаратною особливістю є те, що логічний вихід компаратора може бути програмним чином підключений до входу одного з 16-розрядних таймерів/лічильників, що працює в режимі захоплення. Це дає можливість вимірювати тривалості аналогових сигналів, а також реалізовувати АЦП двотактного інтегрування;

– аналого-цифровий перетворювач побудований за схемою АЦП послідовного наближення з пристроєм вибірки/збереження. Число незалежних каналів перетворення визначається типом мікроконтролера, розрядність АЦП складає 10 біт. Час перетворення вибирається програмно за допомогою встановлення коефіцієнта дільника частоти, що входить до складу блоку АЦП. Важливою особливістю аналого-цифрового перетворювача є функція придушення шуму при перетворенні, коли на точність не впливають завади, що виникають при роботі процесорного ядра;

– мікроконтролери XМega мають декілька каналів цифрово-аналогового перетворення, що значно розширює можливості їх застосування;

– AVR - мікроконтролери можуть бути переведені програмним шляхом в один із шести режимів зниженого енергоспоживання. Для різних сімейств AVR і різних мікроконтролерів у межах кожного сімейства змінюються кількість і реалізоване сполучення доступних режимів зниженого енергоспоживання;

– система команд AVR досить розвинута і нараховує до 118 (а в останніх розробках і до 141) різних інструкцій. Майже всі команди мають фіксовану довжину в одне слово (16 біт), що дозволяє в більшості випадків поєднувати в одній команді і код операції, і операнд(и). В останніх версіях кристалів "mega" реалізована функція множення. За різноманітністю і кількістю інструкцій AVR більше схожі на CISC-, чим на RISC-процесори. Наприклад, у PIC-контролерів система команд нараховує до 75 різних інструкцій, а в MCS-51 вона складає 111.

AVR функціонують у широкому діапазоні споживаючих напруг від 1,8 до 6,0 Вольт. Температурні діапазони роботи - комерційний і індустріальний.

Корпорація Atmel планує подальший розвиток AVR. Виключення складає лише функціонально-збалансоване сімейство "Classic.

Програмні й апаратні засоби підтримки для AVR завжди розроблялися і розробляються паралельно із самими кристалами і містять у собі компілятори, внутрішньосхемні емулятори, налагоджувачі, програматори і найпростіші налагоджувані плати - конструктори практично на будь-який смак. Активно йде процес співробітництва зі сторонніми фірмами, що випускають програмні засоби проектування і налагодження, операційні системи, різноманітні налагоджувані комплекси і внутрішньосхемні емулятори для AVR.

Все вище перераховане говорить про сучасність AVR-мікроконтролерів і можливість їх широкого застосування при проектуванні мікроконтролерних систем різного призначення.

СПИСОК СКОРОЧЕНЬ

- АК – аналоговий компаратор
- АЛП – арифметико-логічний пристрій
- АЦП – аналого-цифровий перетворювач
- ДК – двійковий (машинний) код
- ДНЛ – диференціальна нелінійність
- ДОН – джерело опорної напруги
- ДРВВ – додаткові регістри введення/виведення
- ЕСППЗП – електрично стираємий програмований постійний
запам'ятовуючий пристрій
- ЗВПР - зовнішній пристрій
- ЗПП – зовнішня пам'ять програм
- ІМС ОП – інтегральна мікросхема операційного підсилювача
- ІНЛ – інтегральна нелінійність
- КЗ – канал зв'язку
- КМОН – комплементарний метал-оксид-напівпровідник
- КОП – код операції команди
- МА – мова асемблера
- МЗР – молодший значущий розряд
- МК – мікроконтролер
- МПЛ – мультиплексор
- МПС – мікропроцесорна система
- ОЗП – оперативний запам'ятовуючий пристрій
- ОП – операційний підсилювач
- ПВВ – порти введення/виведення
- ПВЗ – пристрій вибірки-зберігання
- ПДП – прямий доступ до пам'яті
- ПЗП – постійний запам'ятовуючий пристрій

ППЗП – перепрограммовуваний постійний запам'ятовуючий пристрій

ПП – пам'ять програм

ППР – периферійний пристрій

РВВ – регістр введення/виведення

РЗП – регістр загального призначення

РПП – резидентна пам'ять програм

СЗР – старший значущий розряд

СОЗП – статичний оперативний запам'ятовуючий пристрій

СПД – статична пам'ять даних

ТТЛШ – транзисторно-транзисторна логіка з діодами Шоттки

УСАПП – універсальний синхронно/асинхронний приймач-передавач

ЦАП – цифро-аналоговий перетворювач

ЦПП – центральний процесорний пристрій

ША – шина адреси

ШД – шина даних

ШІМ – широтно-імпульсна модуляція

АСК (acknowledge) – підтвердження

СРНА (Clock Phase) – фаза тактового сигналу

СРОЛ (Clock Polarity) – полярність тактового сигналу

EEPROM-пам'ять (Electrically Erasable Programmable Read-Only Memory) - енергонезалежна пам'ять, що може бути електрично стерта та перепрограмована

ETIMSK (Extended Timer/Counter Interrupt MaSK Register) – розширений регістр маски переривань від таймерів/лічильників

FIFO (First In, First Out) – принцип «першим зайшов, першим вийшов», якому відповідає структура даних «черга»

GTCCR (General Timer/Counter Control Register) – загальний регістр керування таймером/лічильником

IEEE (Institute of Electrical and Electronics Engineers) - Інститут інженерів з електротехніки та електроніки

I2C (InterIC, або IIC) – двонаправлена двопровідна шина для так званого міжмікросхемного застосування

JTAG (Joint Test Action Group) - інтерфейс, призначений для підключення складних цифрових мікросхем або пристроїв рівня друкованої плати до стандартної апаратури тестування і налагодження

MISO (Master In Slave Out) – сигнал входу для ведучого, виходу для веденого

MOSI (Master Out Slave In) – сигнал виходу для веденого, входу для ведучого

NACK (no acknowledge) – не підтвердження

OCR (Output Compare Register) – регістр порівняння

SCK (Serial Clock) – лінія тактового сигналу

SCL (Serial CLock) – послідовна лінія тактування

SDA (Serial DAta) – послідовна лінія даних

SFIOR (Special Function Input Output Register) – регістр спеціальних функцій введення/виведення

SPI (Serial Peripheral Interface) - послідовний периферійний інтерфейс

TAP (Test Access Port) - порт тестового доступу

TCCR (Timer/Counter Clock Control Register) – регістр керування таймером/лічильником

TIFR (Timer/Counter Interrupt Flag Register) – регістр прапорців переривань від таймерів/лічильників

TIMSK (Timer/Counter Interrupt MaSK Register) – регістр маски переривань від таймерів/лічильників

TWI (Two Wire (Serial) Interface) – двопровідний інтерфейс

UDR (Universal Data Register) – буферні регістри приймача і передавача

USART (Universal Synchronous and Asynchronous Receiver and Transmitter) – універсальний синхронний/асинхронний приймач-передавач

WDTCR (Watch Dog Timer Control Register) – реєстр керування вартовим таймером

1 ПОРТИ ВВЕДЕННЯ/ВИВЕДЕННЯ

1.1 Призначення та місце пристроїв введення/виведення даних у мікропроцесорних системах

При розгляді зв'язків між окремими частинами мікропроцесорної системи (МПС) використовують поняття інтерфейс, що є границею між кількома пристроями, наприклад, між мікропроцесором/мікроконтролером (МП/МК) та зовнішнім пристроєм (ЗВПР) .

Під інтерфейсом розуміють сукупність уніфікованих технічних і програмних засобів, необхідних для підключення пристроїв до системи чи однієї системи до іншої. Серед інших властивостей інтерфейсу можна відзначити розв'язування задач синхронізації, вибору напряду передачі даних, а інколи приведення у відповідність рівнів чи форм сигналів.

У даному розділі розглядаються інтерфейси AVR-мікроконтролерів.

Як видно з приведеного вище визначення інтерфейсу, його основною частиною є технічні (апаратні) засоби, що забезпечують зв'язок МК-ра та ЗВПР. Ці засоби у мікропроцесорній техніці часто називають «модулем введення/виведення». Основу цього модуля складають порти введення/виведення інформації, виконані на основі регістрів. Крім наявності необхідних апаратних засобів модуль введення/виведення повинен бути програмовуваним, тобто мати відповідні регістри для запису керуючих слів і регістри, які відображають стан інтерфейсу.

Сучасні МП-ри та МК-ри мають команди обміну даними з необхідною периферією. Введення (читання, прийом) відповідає потоку даних від ЗВПР в МП-р (МК-р), а виведення (запис, передача) – потоку даних із МП-ра (МК-ра) до ЗВПР.

1.2 Паралельне та послідовне введення/виведення

Існує два способи обміну даними між ЗВПР і МПС:

- паралельний, коли одночасно передаються всі біти або декілька біт слова даних;
- послідовний, коли біти слова даних пересилаються по черзі, починаючи, наприклад, з його молодшого розряду.

ЗВПР зв'язується з МПС лініями зв'язку, довжина яких при паралельному обміні обмежена і складає кілька метрів.

При послідовному обміні даними обмежень на довжину ліній зв'язку не накладається. Ця обставина, а також бажання використовувати для дистанційного обміну інформацією між ЗВПР і МПС існуючі канали зв'язку (КЗ), обумовили широке поширення послідовного обміну даними між ЗВПР і МПС чи між декількома МПС.

Обмін інформацією в МП-рі або МК-рі здійснюється в паралельній формі. Тому при послідовному обміні даними необхідно: при передачі даних від МПС до ЗВПР - перетворити дані з паралельної форми в послідовну, а при прийомі інформації від ЗВПР та введенні її у МПС – перетворити з послідовної форми в паралельну.

Для перетворення даних з паралельної форми в послідовну, інформація завантажується у регістр зсуву. Вміст регістра зсуву послідовно зсувається на один розряд при надходженні кожного тактового імпульсу від генератора тактових імпульсів (ГТІ).

Дані на виході такого регістра будуть мати послідовну форму. Часто при послідовній передачі в канал зв'язку (КЗ) першим передається молодший біт слова даних (МЗР), останнім – старший біт (СЗР). Процес перетворення даних з паралельної форми в послідовну показаний на рисунку 1.1.

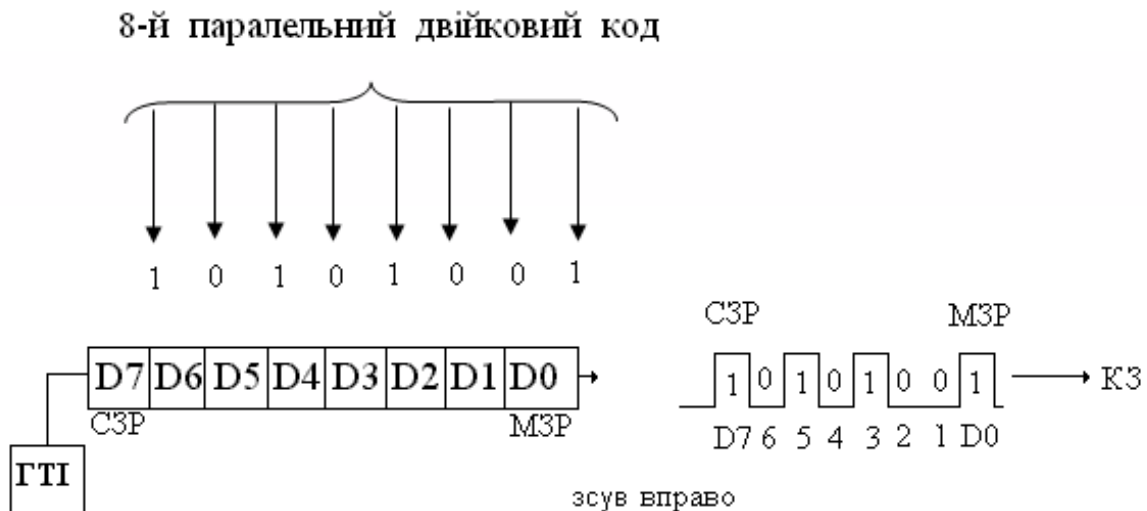


Рисунок 1.1 - Перетворення даних з паралельної форми в послідовну

Для зворотного перетворення даних з послідовної форми в паралельну необхідно виконати дії, зворотні відносно описаного. Дані, що надходять з каналу зв'язку в послідовній формі, вводяться біт за бітом у регістр зсуву. Після заповнення регістра зсуву інформація з нього в паралельній формі передається у МП-р.

Подібні перетворення виконуються в МК-рі при обміні даними через його послідовний порт (рисунок 1.2).

1.3 Порти введення/виведення AVR-мікроконтролерів

1.3.1 Загальні відомості

Як і будь-які інші мікроконтролери, мікроконтролери сімейства AVR мають порти введення/виведення. Кожен порт складається з певного числа виводів, через які мікроконтролер може приймати або передавати цифрові сигнали [1, 2].

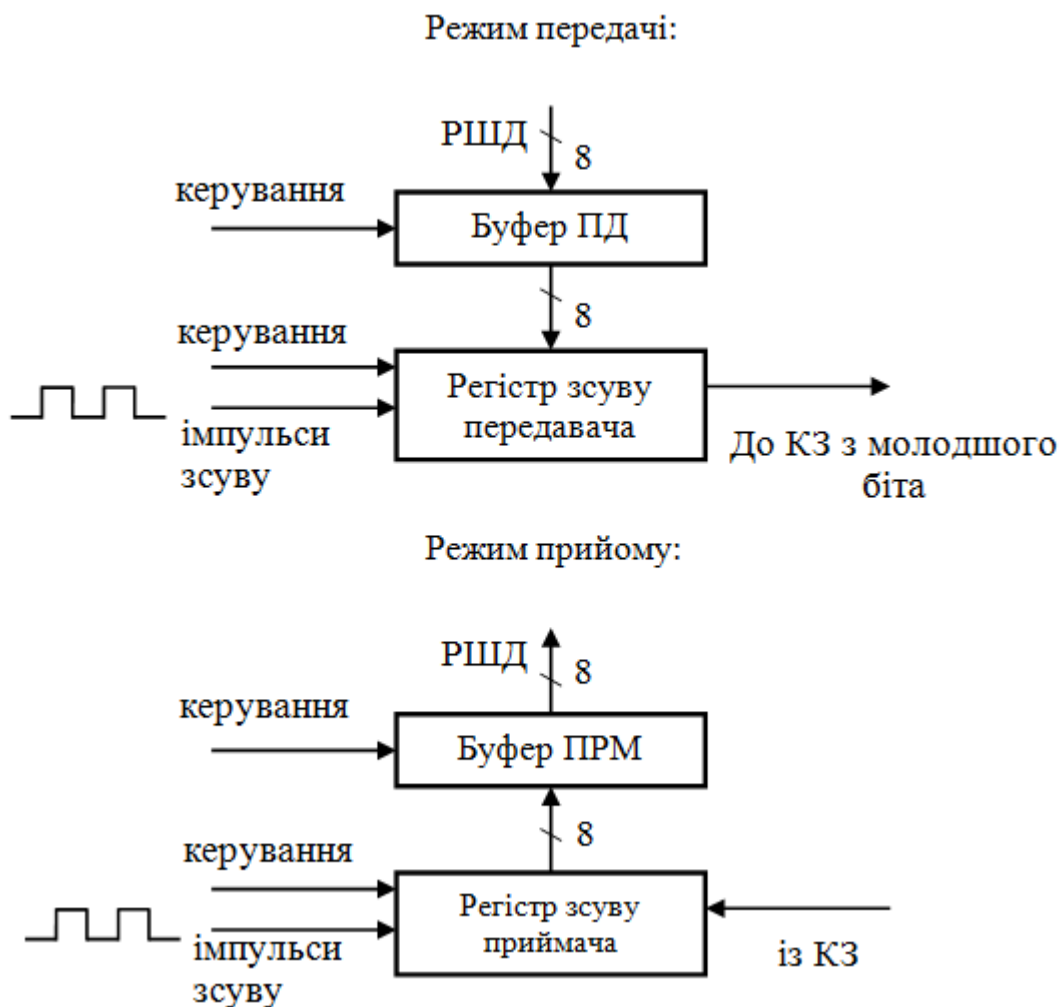


Рисунок 1.2 - Спрощена структура послідовного порту мікроконтролера

Задання напряму передачі даних через будь-який контакт введення/виведення може бути виконано програмно в будь-який момент часу. При цьому деякі моделі сімейства Tiny мають виводи, здатні працювати або тільки як входи, або тільки як виходи. Вхідні буфери портів побудовані за схемою тригера Шмітта. Для більшості ліній, що сконфігуровані як вхідні, також є можливість підключення внутрішнього підтягуючого резистора з опором 35...120 кОм між входом і шиною живлення V_{CC} . Крім того, якщо між входом із задіяним внутрішнім підтягуючим резистором і спільною шиною підключити навантаження, цей вхід може служити джерелом струму.

У портах введення/виведення МК-в сімейства реалізована

функціональність виду «читання/модифікація/запис». Завдяки цьому, використовуючи команди SBI та CBI, можна виконувати операції над будь-яким виводом, не впливаючи на інші виводи порту. Це стосується зміни режиму роботи контакту введення/виведення, зміни вихідного значення і зміни стану внутрішнього підтягуючого резистора (для входів).

Відмінною рисою моделей сімейства Tiny, які випускаються у 8-вивідному корпусі, є сполучення функцій введення/виведення з функціями керування (виводи для підключення зовнішнього резонатора та вивід скидання). Відповідно, в розпорядженні користувача є до 6 ліній введення/виведення, тому що ці моделі можуть бути сконфігуровані для роботи без зовнішнього резонатора і без виводу апаратного скидання.

Мікроконтролери з кожної групи моделей сімейства мають різну кількість портів і, відповідно, контактів введення/виведення (таблиця 1.1).

В усіх МК-х сімейства переважна більшість контактів введення/виведення мають додаткові функції та використовуються периферійними пристроями МК-ів.

1.3.2 Звернення до портів введення/виведення

Звернення до портів виконується через регістри введення/виведення. Під кожен порт (за деякими винятками, наприклад, порт В у моделі ATtiny28x) в адресному просторі введення/виведення зарезервовано по 3 адреси. За цими адресами розміщуються три регістри: регістр даних порту PORTx, регістр напряму даних DDRx (для порту А в моделях ATtiny28x – регістр керування) і регістр виводів порту PINx [1, 2].

Таблиця 1.1 - Порти введення/виведення мікроконтролерів сімейства Mega

Порт введення/виведення	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x, ATmega1280x/2560x
PORTA	●	●	-	●	●	-	●	●	●	●	●	●	●
PORTB	●	●	●	●	●	●	●	●	●	●	●	●	●
PORTC	●	●	7	●	●	7	●	●	●	●	●	●	●
PORTD	●	●	●	●	●	●	●	●	●	●	●	●	●
PORTE	3	-	-	-	●	-	3	-	●	●	●	●	●
PORTF	-	-	-	-	●	-	-	-	●	●	●	●	●
PORTG	-	-	-	-	5	-	-	-	5	6 ¹⁾	6 ¹⁾	6	6
PORTH	-	-	-	-	-	-	-	-	-	-	●	-	●
PORTJ	-	-	-	-	-	-	-	-	-	-	7	-	●
PORTK	-	-	-	-	-	-	-	-	-	-	-	-	●
PORTL	-	-	-	-	-	-	-	-	-	-	-	-	●
Число контактів введення/ виведення	35	32	23	32	53	23	35	32	53	54	69	54	86
¹⁾ - Вивід PG5 - тільки вхід з постійно включеною підтяжкою (суміщений з входом RESET).													

Примітка. Цифра позначає розрядність порту (відмінну від 8).

Оскільки всі виводи порту В моделей ATtiny28x можуть

працювати тільки як входи, цьому порту відповідає тільки один регістр - регістр виводів порту. Адреси всіх регістрів, що відносяться до портів введення/виведення, наведені в таблиці 1.2.

Таблиця 1.2 - Регістри портів введення/виведення

Порт	Регістр	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x	ATmega162x	ATmega64x, ATmega128x	ATmega48x/88x/168x	ATmega164x/324x/644x	ATmega165x, ATmega325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x, ATmega1280x/2560x
A	PORTA	\$1B (\$3B)	-	\$1B (\$3B)			-	\$02 (\$22)					
	DDRA	\$1A (\$3A)	-	\$1A (\$3A)			-	\$01 (\$21)					
	PINA	\$19 (\$39)	-	\$19 (\$39)			-	\$00 (\$20)					
B	PORTB	\$18 (\$38)					\$05 (\$25)						
	DDRB	\$17 (\$37)					\$04 (\$24)						
	PINB	\$16 (\$36)					\$03 (\$23)						
C	PORTC	\$15 (\$35)					\$08 (\$28)						
	DDRC	\$14 (\$34)					\$07 (\$27)						
	PINC	\$13 (\$33)					\$06 (\$26)						
D	PORTD	\$12 (\$32)					\$0B (\$2B)						
	DDRD	\$11 (\$31)					\$0A (\$2A)						
	PIND	\$10 (\$30)					\$09 (\$29)						
E	PORTE	\$07 (\$27)	-	-	\$07 (\$27)	\$03 (\$23)	-	-	\$0E (\$2E)				
	DDRE	\$06 (\$26)	-	-	\$06 (\$26)	\$02 (\$22)	-	-	\$0D (\$2D)				
	PINE	\$05 (\$25)	-	-	\$05 (\$25)	\$01 (\$21)	-	-	\$0C (\$2C)				

Продовження таблиці 1.2

Порт	Регістр	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x	ATmega162x	ATmega64x, ATmega128x	ATmega48x/88x/168x	ATmega164x/324x/644x	ATmega165x, ATmega325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x, ATmega1280x/2560x
F	PORTF	\$03 (\$23)	-	-	-	-	(\$62)	-	-	\$11 (\$31)			
	DDRF	\$02 (\$22)	-	-	-	-	(\$61)	-	-	\$10 (\$30)			
	PINF	\$01 (\$21)	-	-	-	-	\$00 (\$20)	-	-	\$0F (\$2F)			
G	PORTG	-	-	-	-	-	(\$65)	-	-	\$14 (\$34)			
	DDRG	-	-	-	-	-	(\$64)	-	-	\$13 (\$33)			
	PING	-	-	-	-	-	(\$63)	-	-	\$12 (\$32)			
H	PORTH	-	-	-	-	-	-	-	-	-	(\$DA)	-	(\$102)
	DDRH	-	-	-	-	-	-	-	-	-	(\$D9)	-	(\$101)
	PINH	-	-	-	-	-	-	-	-	-	(\$D8)	-	(\$100)
J	PORTJ	-	-	-	-	-	-	-	-	-	(\$DD)	-	(\$105)
	DDRJ	-	-	-	-	-	-	-	-	-	(\$DC)	-	(\$104)
	PINJ	-	-	-	-	-	-	-	-	-	(\$DB)	-	(\$103)
K	PORTK	-	-	-	-	-	-	-	-	-	-	-	(\$108)
	DDRK	-	-	-	-	-	-	-	-	-	-	-	(\$107)
	PINK	-	-	-	-	-	-	-	-	-	-	-	(\$106)
L	PORTL	-	-	-	-	-	-	-	-	-	-	-	(\$10B)
	DDRL	-	-	-	-	-	-	-	-	-	-	-	(\$10A)
	PINL	-	-	-	-	-	-	-	-	-	-	-	(\$109)

Дійсні назви регістрів отримуються підстановкою назви порту замість символу x . Відповідно, регістри порту A називаються PORTA, DDRA, PINA, порту B - PORTB, DDRB, PINB і т. ін.

Загалом кажучи, «регістри» PIN_x , де $x = A/B/D$ (і т. ін.), не є регістрами. За цими адресами здійснюється доступ до фізичних значень сигналів на виводах порту. Відповідно, вони доступні тільки для читання, тоді як інші регістри доступні і для читання, і для запису.

Тим не менш, в нових моделях мікроконтролерів (всі моделі, крім ATmega8515x/8535x, ATmega8x/16x/32x/64x/128x і ATmega162x) запис 1 в біт регістра PIN_x призводить до переключення стану відповідного біта регістра даних $PORT_x$.

Запис у порт означає запис необхідного стану для кожного виводу порту у відповідний регістр даних порту $PORT_x$. А читання стану порту виконується читанням або регістра даних порту $PORT_x$, або регістра виводів порту PIN_x . При читанні регістра виводів порту PIN_x відбувається зчитування сигналів, які присутні на виводах порту. А при читанні регістра даних порту $PORT_x$ відбувається зчитування даних, які перебувають у регістрі-защипці порту (як для вхідних, так і для вихідних контактів). При знаходженні мікроконтролера в стані скидання виводи всіх портів перебувають у третьому стані (стан високого імпедансу Hi-Z).

1.3.3 Конфігурування портів введення/виведення

Порти введення/виведення (і навіть окремі розряди однойменних портів) різних моделей сімейства мають різні можливості конфігурування.

Спрощена структурна схема одного з каналів порту введення/виведення R_{xp} при роботі його у якості цифрового входу/виходу загального призначення наведена на рисунку 1.3 [2].

Кожному виводу порту відповідають три розряди регістрів введення/виведення: PORTxn (регістр PORTx), DDRxn (регістр DDRx) і PINxn (регістр PINx). Дійсні назви розрядів регістрів отримуються підстановкою назви порту замість символу «x» і номера розряду замість символу «n». Порядковий номер виводу порту відповідає порядковому номеру розряду регістрів цього порту. Тому, якщо розрядність порту менше восьми, у регістрах порту використовується відповідне число молодших розрядів. Незадіяні старші розряди регістрів доступні тільки для читання і завжди містять «0».

Розряд DDRxn регістра DDRx визначає напрям передачі даних через контакт введення/виведення. Якщо цей розряд встановлений в «1», то n-й вивід порту є виходом, якщо ж скинутий в «0» - входом.

Розряд PORTxn регістра PORTx виконує подвійну функцію. Якщо вивід функціонує як вихід (DDRxn = «1»), цей розряд визначає стан виводу порту. Якщо розряд встановлений в «1», на виводі встановлюється напруга ВИСОКОГО рівня. Якщо розряд скинутий в «0», на виводі встановлюється напруга НИЗЬКОГО рівня.

Якщо ж вивід функціонує як вхід (DDRxn = «0»), розряд PORTxn визначає стан внутрішнього підтягуючого резистора для даного виводу. При встановленні розряду PORTxn в «1» підтягуючий резистор підключається між виводом мікроконтролера і живленням, за умови, що біт PUD = 0 (див. нижче).

Загалом кажучи, керування підтягуючими резисторами майже у всіх мікроконтролерах сімейства, за винятком моделей ATmega161x, здійснюється на двох рівнях. Загальне керування (для всіх виводів портів) здійснюється розрядом PUD регістра спеціальних функцій SFIOR або регістра керування мікроконтролера MCUCR [1, 2]. У моделях ATmega64x та ATmega128x цей регістр розташовується за адресою \$20 (\$40), а в інших моделях - за адресою \$30 (\$50). Регістр MCUCR розташований за адресою

\$35 (\$55). Формати цих регістрів наведені на рисунку 1.4.

	7	6	5	4	3	2	1	0	
	X	X	X	X	X	PUD	X	X	ATmega8515x/8535x ATmega8x/16x/32x ATmega64x/128x ATmega162x
Читання (R)/Запис (W)	X	X	X	X	X	R/W	X	X	
Початкове значення	0	0	0	0	0	0	0	0	
а)									
	7	6	5	4	3	2	1	0	
	—	—	—	PUD	—	—	IVSEL	IVCE	ATmega48x/88x/168x
Читання (R)/Запис (W)	R	R	R	R/W	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	JTD	—	—	PUD	—	—	IVSEL	IVCE	ATmega164x/324x/644x ATmega165x ATmega325x/3250x ATmega645x/6450x ATmega640x ATmega1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R	R	R/W	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
б)									

Рисунок 1.4 – Регістри керування підтяжкою SFIOR (а) та MCUCR (б)

Якщо розряд PUD скинутий в «0» (початковий стан), стан підтягуючих резисторів буде визначатися станом розрядів PORTxn для кожного входу порту, при умові, що відповідний розряд DDRxn скинутий в «0». Якщо ж розряд PUD встановлений в «1», підтягуючі резистори відключаються від усіх виводів мікроконтролера.

Зверніть увагу, що при переключенні виводу між третім станом (DDRxn = «0», PORTxn = «0») і станом ВИСОКОГО рівня (DDRxn = «1», PORTxn = «1») відбувається перехід через один із проміжних станів: або включається підтягуючий резистор (DDRxn = «0», PORTxn = «1»), або вихід переключається у стан НИЗЬКОГО рівня (DDRxn = «1», PORTxn = «0»). Найбільше використання має, як правило, перший варіант, оскільки для виводів з високоімпедансним станом байдуже, яким чином сформується ВИСОКИЙ рівень. Якщо в якому-небудь випадку це не підходить, користувач може відключити підтягуючі резистори від усіх

портів встановленням в «1» розряду PUD регістра SFIOR.

Аналогічна ситуація виникає і при переключенні між станом із включеним підтягуючим резистором ($DDR_{xn} = \langle 0 \rangle$, $PORT_{xn} = \langle 1 \rangle$) і станом НИЗЬКОГО рівня ($DDR_{xn} = \langle 1 \rangle$, $PORT_{xn} = \langle 0 \rangle$). У цьому випадку проміжним станом є або високоімпедансний стан ($DDR_{xn} = \langle 0 \rangle$, $PORT_{xn} = \langle 0 \rangle$), або стан ВИСОКОГО рівня ($DDR_{xn} = \langle 1 \rangle$, $PORT_{xn} = \langle 1 \rangle$).

Всі можливі сполучення станів керуючих розрядів і, відповідно, конфігурації виводів портів зведені у таблиці 1.3.

Стан виводу мікроконтролера (незалежно від розряду DDR_{xn}) може бути отриманий шляхом читання розряду PIN_{xn} регістра PIN_x . При цьому варто пам'ятати, що між дійсною зміною сигналу на виводі та зміною розряду PIN_{xn} існує затримка. Ця затримка вноситься вузлом синхронізації, що складається, як показано на рисунку 1.3, з розряду тригера PIN_{xn} і додаткового тригера-защипки. Значення сигналу на виводі мікроконтролера фіксується тригером-защипкою за високим рівнем тактового сигналу і переписується потім у розряд PIN_{xn} за наростаючим фронтом тактового сигналу ($f_{CLK\ I/O}$). Відповідно, величина затримки може становити від 0.5 до 1.5 періоду системного тактового сигналу, як показано на рисунку 1.5, а.

З цієї ж причини між операціями зміни, виведення і повторного зчитування станів виводу необхідно вставляти команду NOP. Оскільки команда OUT записує інформацію в тригер $PORT_{xn}$ за додатним фронтом тактового сигналу, затримка в цьому випадку дорівнює одному періоду тактового сигналу (рисунок 1.5, б)

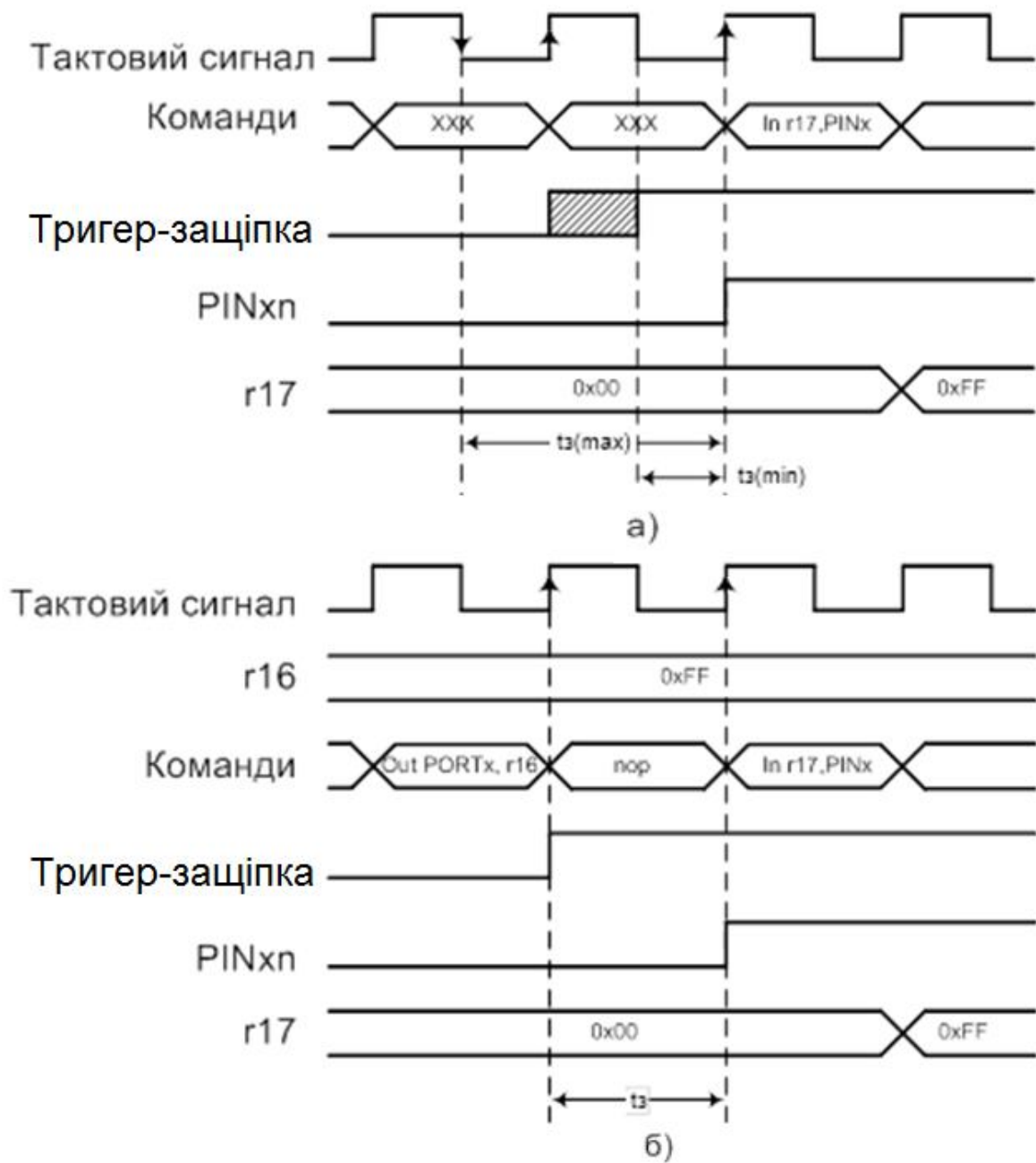


Рисунок 1.5 – Синхронізація при читанні стану виводу:

а) - при зчитуванні стану розряду PINxn; б) - при повторному зчитуванні

Таблиця 1.3 – Конфігурації виводів портів

DDRxn	PORTxn	PUD*	Функція виводу	Резистор	Примітки
0	0	X	Вхід	Відключений	Третій стан (Hi-Z)**
0	1	0	Вхід	Підключений	При підключенні навантаження між виводом і спільним проводом вивід є джерелом струму
0	1	1	Вхід	Відключений	Третій стан (Hi-Z) для виходу
1	0	X	Вихід	Відключений	Вихід скинутий в «0»
1	1	X	Вихід	Відключений	Вихід встановлений в «1»
* Відсутній у моделях ATmega161х. ** Стан виводів порту при скиданні.					

Далі наведений приклад конфігурування одного з портів мікроконтролера. У прикладі виводи 0 і 1 порту В встановлюються в «1», виводи 2 і 3 - в «0». Виводи 4...7 порту конфігуруються як входи, при цьому до виводів 6 і 7 підключаються підтягуючі резистори.

Приклад на асемблері

...

```
ldi r16, (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0)
```

```
ldi r17, (1<<DDRB3) | (1<<DDRB2) | (1<<DDRB1) | (1<<DDRB0)
```

```
out PORTB,r16 ; Задати стан виходів і
```

```
; підтягуючих резисторів
```

```
out DDRB,r17; ; Задати режими роботи виводів
```

```
nor ; Синхронізація
```

```
in r16,PINB ; Прочитати стан виводів порту
```

Приклад на C

```
unsigned char i;
```

...

```

/* Задати стан виходів і підтягуючих резисторів */
/* Задати режими роботи виводів */
PORTB = (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0);
DDRB=(1<<DDRB3) | (1<<DDRB2) | (1<<DDRB1) | (1<<DDRB0);
_NOP();                                /* Синхронізація */
i=PINB;                                /* Прочитати стан виводів порту */
...

```

Відзначимо, що переважна більшість контактів введення/виведення всіх мікроконтролерів сімейства мають додаткові функції і можуть використовуватися різними периферійними пристроями мікроконтролерів. При цьому можливі дві ситуації. В одних випадках користувач повинен самостійно задавати конфігурацію виводу, а в інших - вивід конфігурується автоматично при включенні відповідного периферійного пристрою [1, 2].

1.4 Апаратний модулятор

Апаратний модулятор (Hardware Modulator) призначений для керування виводом RA2 деяких мікроконтролерів, наприклад ATtiny28x. Єдиним завданням модулятора є генерування послідовності імпульсів із заданими параметрами [1, 2].

Структурна схема блоку апаратного модулятора наведена на рисунку 1.6.

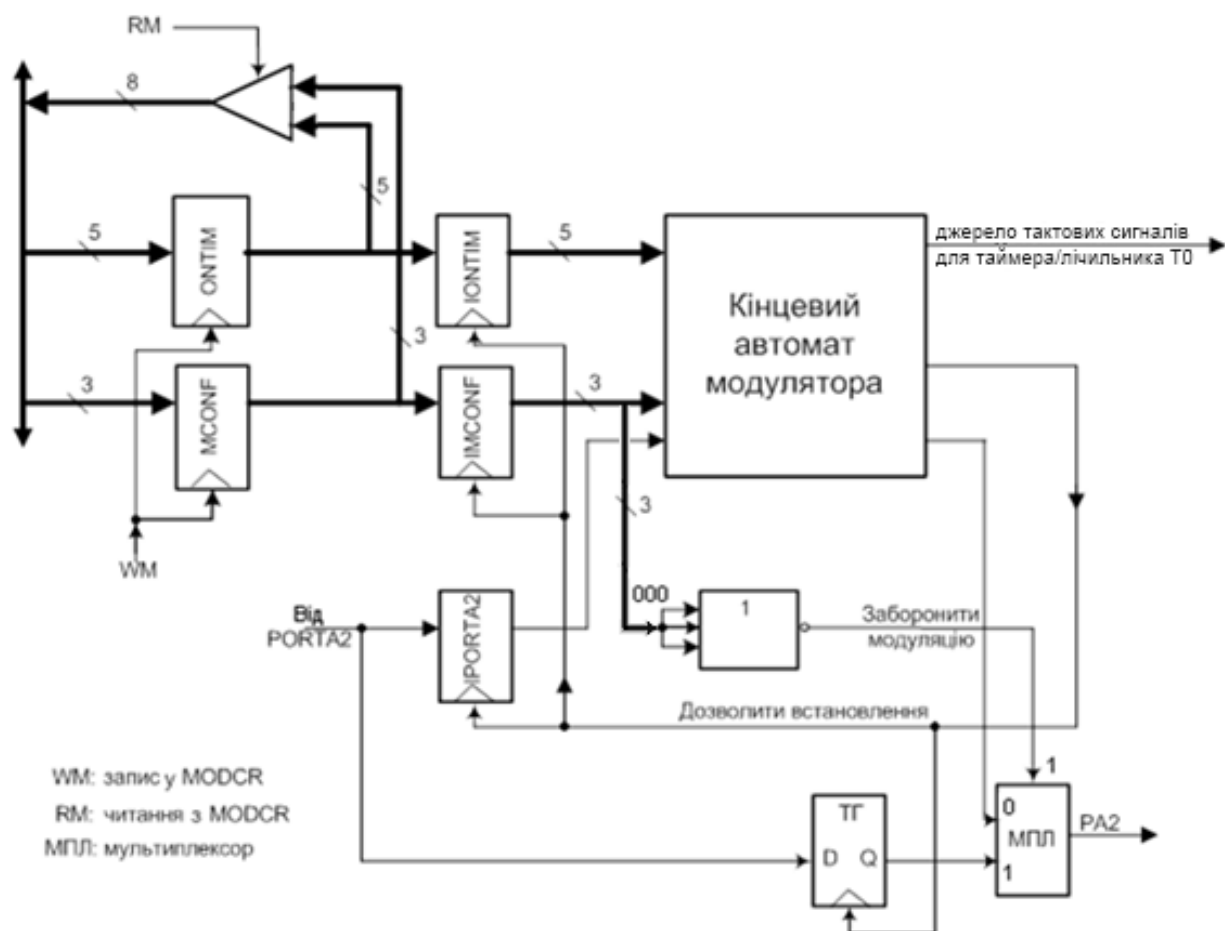


Рисунок 1.6 – Структурна схема апаратного модулятора

Для керування параметрами сигналу, що генерується, призначений регістр MODCR (MODulation Control Register), розташований за адресою \$02. Формат цього регістра наведено на рисунку 1.7.

	7	6	5	4	3	2	1	0
\$02	ONTIM4	ONTIM3	ONTIM2	ONTIM1	ONTIM0	MCONF2	MCONF1	MCONF0
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 1.7 – Формат регістра MODCR

Старші 5 розрядів регістра (ONTIM4...ONTIM0) визначають час знаходження виходу PA2 в активному стані (НИЗЬКИЙ рівень). Число машинних циклів мікроконтролера, протягом яких вивід PA2 буде перебувати в активному стані (формування нульового вихідного імпульсу), на одиницю більше значення, записаного в розрядах ONTIM4...ONTIM0.

Молодші три розряди регістра (MCONF2...MCONF0) визначають співвідношення між довжиною активного і неактивного станів виводу PA2 і відповідно коефіцієнт заповнення сигналу ($\gamma = t_{im}/T$, де T – період, t_{im} - довжина імпульсу), що генерується. Значення коефіцієнта заповнення сигналу для різного вмісту цих розрядів, а також можливі значення періоду сигналу, що генерується, наведені в таблиці 1.4.

Несуча частота (частота сигналу, що генерується) визначається співвідношенням $f_c = f_{CLK}/(T_{ON}+T_{OFF})$, де f_{CLK} - тактова частота мікроконтролера, T_{ON} - час знаходження виводу PA2 в активному стані (лог. 0), T_{OFF} - знаходження виводу PA2 в неактивному стані (лог. 1). При значенні $MCONF = 111$ частота вихідного сигналу дорівнює частоті тактового сигналу мікроконтролера (f_{CLK}).

Таблиця 1.4 - Залежність коефіцієнта заповнення і періоду модуляції від вмісту розрядів MCONF2...MCONF0

MCONF 2...0	Тривалість сигналу		Коефіцієнт заповнення	Період модуляції*		Пояснення
	«0»	«1»		min	max	
000	X	X	1	X	X	Немає модуляції
001	ONTIM +1	ONTIM + 1	0.5	2T _{CLK}	64T _{CLK} ***	
010	ONTIM +1	2(ONTIM+1)	0.33	3T _{CLK}	96T _{CLK}	
011	ONTIM +1	3(ONTIM +1)	0.25	4T _{CLK}	128T _{CLK}	
100	2(ONTIM +1)	ONTIM +1	0.67	3T _{CLK}	96T _{CLK}	
101	3(ONTIM + 1)	ONTIM +1	0.75	4T _{CLK}	128T _{CLK}	
110	Зарезервовано					
111	X	X	**	1T _{CLK}	1T _{CLK}	Сигнал високої частоти
* Мінімальний період модуляції відповідає ONTIM=0, а максимальний - ONTIM=31. ** Залежить від параметрів тактового сигналу мікроконтролера, тому що сигнал на виході PA2 ідентичний йому. *** T_{CLK} - період тактового сигналу мікроконтролера.						

Для прикладу в таблиці 1.5 та на рисунку 1.8 приводяться значення розглянутих параметрів для деяких значень частот тактового сигналу і сигналу, що генерується.

Таблиця 1.5 – Типові конфігурації модулятора

Частота резонатора [МГц]	Несуча частота [КГц]	Похибка встановлення частоти [%]	Коефіцієнт заповнення	Значення параметрів	
				ONTIM	MCONF
0.455	38	0.2	0.25	2	011
0.455	38	0.2	0.33	3	010

Продовження таблиці 1.5

Частота резонатора [МГц]	Несуча частота [КГц]	Похибка встановлення частоти [%]	Коефіцієнт заповнення	Значення параметрів	
				ONTIM	MCONF
0.455	38	0.2	0.50	5	001
0.455	38	0.2	0.67	3	100
0.455	38	0.2	0.75	2	101
1	38	1.2	0.50	12	001
1.8432	38	1.1	0.25	11	011
1.8432	38	1.1	0.33	15	010
1.8432	38	1.1	0.50	23	001
2	38	1.2	0.25	12	011
2	38	1.2	0.50	25	001
2.4576	38	1.1	0.50	31	001
3.2768	38	2.0	0.25	21	011
4	38	1.2	0.25	25	011
0.455	455	0.0	≈0.50	X	111
1	455	9.9	0.50	0	001
1.82	455	0.0	0.25	0	011
1.82	455	0.0	0.50	1	001
1.8432	455	1.3	0.25	0	011
1.8432	455	1.3	0.50	1	001
2	455	9.9	0.25	0	011
2	455	9.9	0.50	1	001
2.4576	455	10.0	0.33	1	010
2.4576	455	10.0	0.50	2	001
3.2768	455	10.0	0.25	1	011
3.2768	455	10.0	0.50	3	001
3.64	455	0.0	0.25	1	011
3.64	455	0.0	0.50	3	001
4	455	9.9	0.25	1	011
4	455	9.9	0.50	3	001

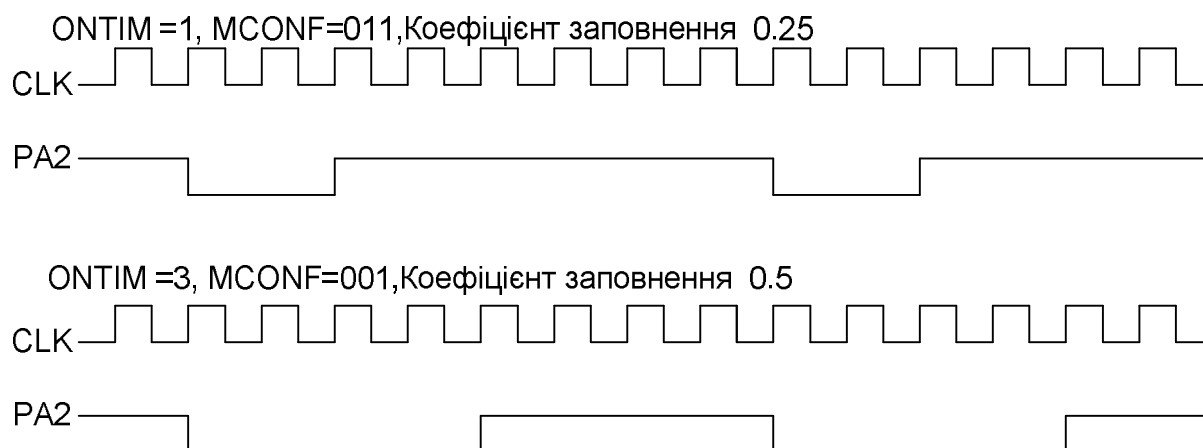


Рисунок 1.8 - Приклади сигналів, що формуються апаратним модулятором

Зрозуміло, що стан виходу PA2 може визначатися не тільки апаратним модулятором. Всі можливі умови функціонування виходу PA2 наведені в таблиці 1.6.

Таблиця 1.6 - Функціонування виходу PA2

PORTA2	MCONF2...MCONF0	Стан виходу PA2
0	000	0
0	001...111	Визначається модулятором
1	X	1

Апаратний модулятор має ще одну корисну функцію. Крім керування станом виводу PA2 мікроконтролера, вихідний сигнал апаратного модулятора може використовуватися в якості вхідного для таймера/лічильника T0. У цьому випадку таймер/лічильник може використовуватися як для обмеження тривалості посилки, так і для задання інтервалу між послідовними посилками. Останнє можливо завдяки тому, що сигнал з виходу модулятора може подаватися на вхід таймера/лічильника T0, навіть якщо розряд PORTA2 встановлений в «1» і генерація відсутня.

Для використання таймера/лічильника T0 в описаному режимі він повинен бути сконфігурований таким чином, щоб за його переповненням змінювався стан виходу PA2 [1, 2].

Перед запуском модулятора варто занести необхідні значення в регістр MODCR. Далі запуск може бути здійснений двома способами: скиданням розряду PORTA2 регістра PORTA або заданням конфігурації таймера/лічильника T0 таким чином, щоб він скинув вихід PA2 при переповненні.

Зупинити генерацію можна також двома способами: програмно встановивши відповідні біти або за допомогою таймера (встановлення виходу PA2 при переповненні).

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Яким чином виконується звернення до портів введення/виведення?
- 2) Які три регістри пов'язані з кожним портом введення/виведення?
- 3) Що означають x та n у позначеннях $PORTx_n$, $DDRx_n$ і $PINx_n$?
- 4) Як здійснюється загальне керування резисторами, що підтягують?
- 5) Яким чином може бути отриманий стан виводу мікроконтролера?
- 6) Чим пояснюється наявність затримки між дійсною зміною сигналу на виводі порту та зміною розряду $PINx_n$?
- 7) Для чого призначений апаратний модулятор?
- 8) Розкажіть про регістр, який призначений для керування параметрами сигналу, що генерується апаратним модулятором.
- 9) В якості чого може використовуватись вихідний сигнал апаратного модулятора?
- 10) Яким чином можна зупинити генерацію сигналу апаратним модулятором?

ЛІТЕРАТУРА [1...6, 8...10]

2 ПОСЛІДОВНИЙ СИНХРОННО/АСИНХРОННИЙ ІНТЕРФЕЙС

2.1 Загальні відомості

Значна частина мікроконтролерів сімейства AVR мають у своєму складі модулі або універсального асинхронного (UART), або універсального синхронно/асинхронного (USART) приймача-передавача (УСАПП). У деяких моделях міститься по декілька таких модулів [1, 2].

Відзначимо, що модулі USART, які мають всі мікроконтролери сімейства Mega, при роботі в асинхронному режимі сумісні з модулями UART як за розміщенням розрядів керуючих регістрів, так і за функціонуванням. Невеликі відмінності є тільки в роботі схеми буферизації блоку приймача у назві (але не в призначенні) деяких розрядів керуючих регістрів. В деяких нових моделях модулі USART можуть використовуватись в якості ведучого шини SPI (таблиця 2.1) [2].

Таблиця 2.1 – Модулі USART в мікроконтролерах сімейства Mega

Модуль приймача- передавача	ATmega8515x/ 8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48/x88x/ 168x	ATmega162x	ATmega164x/324x/ 644x	ATmega165x, ATmega325x/350x, ATmega645x/6450x	ATmega640x/ 1280x/1281x	ATmega2560x/ 2561x
USART0	•	•	•	• ¹⁾	•	• ¹⁾	•	• ¹⁾	• ¹⁾
USART1			•		•	• ¹⁾		• ¹⁾	• ¹⁾
USART2								• ¹⁾	
USART3								• ¹⁾	
¹⁾ Може працювати в режимі SPI									

Всі модулі приймачів-передавачів забезпечують дуплексний обмін по послідовному каналу, при цьому швидкість передачі даних може змінюватись у доволі широких межах. У модулях UART пакет може бути 8-ми чи 9-ти розрядним, а в модулях USART його довжина може складати від 5-ти до 9-ти розрядів. Ще однією особливістю модулів USART є наявність схем формування та контролю парності.

Модулі USART/UART, що реалізовані у мікроконтролерах сімейства, можуть виявляти наступні нетипові ситуації:

- переповнення;
- помилку кадрування;
- некоректний старт-біт.

Для зменшення ймовірності збоїв у модулях також реалізована така корисна функція, як фільтрація завад.

Для взаємодії з програмою у модулях передбачені 3 переривання, запит на генерацію яких формується при виникненні наступних подій: «передачу завершено», «регістр даних передавача пустий» та «прийом завершено».

Зазвичай, виводи мікроконтролера, що використовуються модулями USART/UART, є лініями портів введення/виведення загального призначення. Як приклад, виводи мікроконтролерів, що використовуються модулями USART сімейства Mega, наведені у таблиці 2.2. Там же вказані функції цих виводів.

Таблиця 2.2 – Виводи, що використовуються модулями USART

Назва	ATmega8515x	ATmega8535x	ATmega8x	ATmega16/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x, ATmega325x/350x, ATmega645x/6450x	ATmega640x/1280x/2560x	ATmega1281x/2561x	Опис
RXD	PD0	PD0	PD0	PD0	-	PD0	-	-	PE0	-	-	Вхід USART/ USART 0
RXD0	-	-	-	-	PE0	-	PD0	PD0	-	PE0	PE0	
TXD	PD1	PD1	PD1	PD1	-	PD1	-	-	PE1	-	-	Вихід USART/ USART 0
TXD0	-	-	-	-	PE1	-	PD1	PD1	-	PE1	PE1	
XCK	PD4	PB0	PD4	PB0	-	PD4	PD4	-	PE2	-	-	Вхід/вихід зовнішнього тактового сигналу USART/ USART 0
XCK0	-	-	-	-	PE2	-	-	PB0	-	PE2	PE2	
RXD1	-	-	-	-	PD2	-	PB2	PD2	-	PD2	PD2	Вхід USART1
TXD1	-	-	-	-	PD3	-	PB3	PD3	-	PD3	PD3	Вихід USART1
XCK1	-	-	-	-	PD5	-	PD2	PD4	-	PD5	PD5	Вхід/вихід зовнішнього тактового сигналу USART1
RXD2	-	-	-	-	-	-	-	-	-	PH0	-	Вхід USART2
TXD2	-	-	-	-	-	-	-	-	-	PH1	-	Вихід USART2
XCK2	-	-	-	-	-	-	-	-	-	PH2	-	Вхід/вихід зовнішнього тактового сигналу USART2
RXD3	-	-	-	-	-	-	-	-	-	PJ0	-	Вхід USART3
TXD3	-	-	-	-	-	-	-	-	-	PJ1	-	Вихід USART3
XCK3	-	-	-	-	-	-	-	-	-	PJ2	-	Вхід/вихід зовнішнього тактового сигналу USART3

2.2 Використання модулів USART/UART

2.2.1 Опис структури модуля

Спрощена структурна схема одного модуля USART/UART наведена на рисунку 2.1. Елементи схеми, виділені на рисунку сірим кольором, є тільки у складі модулів USART [2].

Як показано на рисунку, модуль складається з трьох частин: блоку тактування, блоку передавача та блоку приймача. Блок тактування модулів USART включає схему синхронізації, яка використовується при роботі у синхронному режимі і контролер швидкості передачі. У модулях UART блок тактування складається тільки з контролера швидкості передачі.

Блок передавача містить однорівневий буфер UDR, регістр зсуву, схему формування біта парності (тільки USART) і схему керування. Блок приймача, у свою чергу, включає схеми відновлення тактового сигналу і даних, схему контролю парності (тільки USART), дворівневий (USART) чи однорівневий (UART) буфер UDR, регістр зсуву, а також схему керування.

Буферні регістри приймача і передавача розміщуються за однією адресою простору регістрів введення/виведення і позначаються як регістр даних UDR (Universal Data Register) (UDR_n, де $n = 0/1$). У цьому регістрі зберігаються молодші 8 розрядів даних, які приймаються чи передаються. Під час читання виконується звернення до буферного регістра UDR приймача, а у разі запису – до буферного регістра UDR передавача. Розміщення регістрів даних UDR для деяких моделей мікроконтролерів наведено у таблиці 2.3.

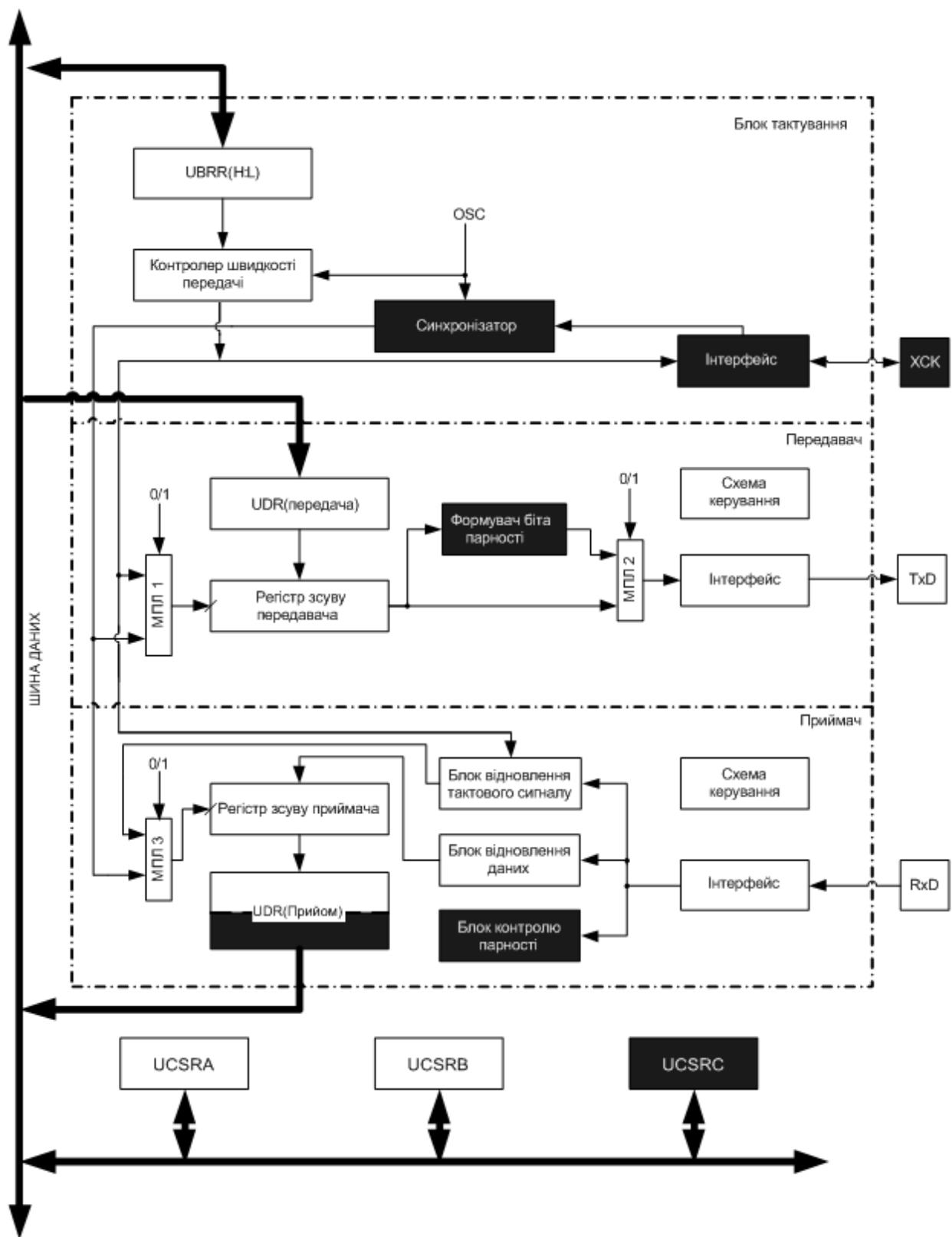


Рисунок 2.1 – Спрощена структурна схема модуля USART/UART

Таблиця 2.3 – Розміщення регістрів даних модулів USART/UART деяких мікроконтролерів сімейства AVR

Регістр	Адреса	ATmega8515x/8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x, ATmega325x/350x, ATmega645x/6450x	ATmega640x/1280x/1281x	ATmega2560x/2561x	Опис
UDR	\$0C (\$2C)	•	•								Регістр даних USART
	(\$C6)							•			
UDR0	\$0C (\$2C)			•		•					Регістр даних USART0
	(\$C6)				•		•		•	•	
UDR1	\$03 (\$23)					•					Регістр даних USART1
	(\$9C)			•							
	(\$CE)								•	•	
UDR2	(\$D6)								•		Регістр даних USART2
UDR3	(\$136)								•		Регістр даних USART3

У модулях USART буфер приймача є дворівневим (FIFO-буфер), зміна стану якого відбувається при будь-якому зверненні до регістра UDR. У зв'язку з цим, не варто використовувати регістр UDR в якості операндів команд типу «читання/модифікація/запис» (SBI та CBI).

Окрім того, треба бути дуже обережними при використанні команд перевірки SBIC та SBIS, оскільки вони також змінюють стан буфера приймача.

Для керування модулями USART використовуються три регістри: UCSRA (UCSRnA), UCSRB (UCSRnB) та UCSRC (UCSRnC), де n = 0, 1, 2 або 3. Адреси цих регістрів деяких мікроконтролерів сімейства AVR вказані у таблиці 2.4.

Формат регістрів UCSRA (UCSRnA), UCSRB (UCSRnB) та UCSRC (UCSRnC) наведений на рисунках 2.2...2.4, а значення розрядів цих регістрів описано у таблицях 2.5...2.7 відповідно.

	7	6	5	4	3	2	1	0
	RXC _n	TXC _n	UDRE _n	FE _n	DOR _n	UPE _n	U2X _n	MPCM _n
Зчитування(R)/Запис(W)	R	R/W	R	R	R	R	R/W	R/W
Початкове значення	0	0	1	0	0	0	0	0

Рисунок 2.2 – Формат регістрів UCSRA (UCSRnA)

	7	6	5	4	3	2	1	0
	RXCIE _n	TXCIE _n	UDRIEE _n	RXEN _n	TXEN _n	UCSZ _{n2}	RXB8 _n	TXB8 _n
Зчитування(R)/Запис(W)	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Початкове значення	0	0	0	0	0	0	1	0

Рисунок 2.3 – Формат регістрів UCSRB (UCSRnB)

	7	6	5	4	3	2	1	0	
	—	UMSEL _n	UPM _{n1}	UPM _{n0}	USBS _n	UCSZ _{n1}	UCSZ _{n0}	UCPOL _n	ATmega64x ATmega128x
Зчитування(R)/Запис(W)	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	1	1	0	

	7	6	5	4	3	2	1	0	
	URSEL _n	UMSEL _n	UPM _{n1}	UPM _{n0}	USBS _n	UCSZ _{n1}	UCSZ _{n0}	UCPOL _n	ATmega8515x/8535x ATmega8x/16x/32x ATmega162x ATmega165x/325x/3250x ATmega645x/6450x
Зчитування(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	1	1	0	

	7	6	5	4	3	2	1	0	
	UMSEL _{n1}	UMSEL _{n0}	UPM _{n1}	UPM _{n0}	USBS _n	UCSZ _{n1}	UCSZ _{n0}	UCPOL _n	ATmega48x/88x/168x ATmega164x/324x/644x ATmega640x/1280x/1281x ATmega2560x/2561x
Зчитування(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	1	1	0	

Рисунок 2.4 – Формат регістрів UCSRC (UCSRnC)

Таблиця 2.4 – Регістри керування та стану модулів USART

Регістр	Адреса	ATmega8515x/8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x, ATmega325x/3250x, ATmega645x/6450x	ATmega640x/1280x/2560x	ATmega1281x/2561x	Опис
UCSRA	\$0B (\$2B)	•	•								Регістр керування А
UCSRB	\$0A(\$2A)	•	•								Регістр керування В
UCSRC	\$20(\$40)	•	•								Регістр керування С
UCSR0A	\$0B (\$2B)			•		•					Регістр А керування USART0
	(\$C0)				•		•	•	•	•	
UCSR0B	\$0A(\$2A)			•		•					Регістр В керування USART0
	(\$C1)				•		•	•	•	•	
UCSR0C	\$20 (\$40)					•					Регістр С керування USART0
	(\$95)			•							
	(\$C2)				•		•	•	•	•	
UCSR1A	\$02(\$22)					•					Регістр А керування USART1
	(\$9B)			•							
	(\$C8)						•		•	•	
UCSR1B	\$01(\$21)					•					Регістр В керування USART1
	(\$9A)			•							
	(\$C9)						•		•	•	
UCSR1C	\$3C (\$5C)					•					Регістр С керування USART1
	(\$9D)			•							
	(\$CA)						•		•	•	
UCSR2A	(\$D0)								•		Регістр А керування USART2
UCSR2B	(\$D1)								•		Регістр В керування USART2
UCSR2C	(\$D2)								•		Регістр С керування USART2
UCSR3A	(\$130)								•		Регістр А керування USART3
UCSR3B	(\$131)								•		Регістр В керування USART3
UCSR3C	(\$132)								•		Регістр С керування USART3

Таблиця 2.5 – Розряди регістрів UCSRA, UCSR0A та UCSR1A деяких мікроконтролерів сімейства AVR

Розряд	Назва	Опис
7	RXC (RXCn)	<u>Прапорець завершення прийому</u> Прапорець встановлюється в «1» при пересиланні прийнятого слова з регістра зсуву приймача в регістр даних UDR і утримується в 1 при наявності непрочитаних даних у буфері приймача (регістр даних UDR). Скидається прапорець апаратно після очищення буфера (у UART - після читання регістра даних). Якщо розряд RXCIE (RXCIEн) регістра UCSRB (UCSRnB) встановлено, то при встановленні прапорця генерується запит на переривання «прийом завершено»
6	TXC (TXCn)	<u>Прапорець завершення передачі</u> Прапорець встановлюється в «1» після передачі усіх розрядів пакету із регістра зсуву передавача, за умови, що у регістр даних UDR не було завантажено нового значення. Якщо розряд TXCIE (TXCIEн) регістра UCSRB (UCSRnB) встановлено, то при встановленні прапорця генерується переривання «передачу завершено». Прапорець скидається апаратно при виконанні підпрограми обробки переривання чи програмно, записуючи у нього лог. 1
5	UDRE (UDREн)	<u>Прапорець очищення регістра даних</u> Даний прапорець встановлюється в «1» при пустому буфері передавача (після пересилання байта із регістра даних UDR у регістр зсуву передавача). Встановлений прапорець означає, що у регістр даних можна завантажити нове значення. Якщо розряд UDRIE регістра UCR (UCSRB) встановлено, генерується запит на переривання «регістр даних пустий». Прапорець скидається апаратно, при записі у регістр даних
4	FE (FEн)	<u>Прапорець помилки кадрування</u> Прапорець встановлюється в «1» при виявленні помилки кадрування, тобто якщо перший стоп-біт прийнятої послідовності дорівнює «0». Прапорець скидається при прийомі стоп-біта, який дорівнює «1»
3	OR (ORн) DOR (DORн)	<u>Прапорець переповнення</u> В USART прапорець встановлюється в «1», якщо у момент виявлення нового старт-біта в регістрі зсуву приймача знаходиться останнє прийняте слово, а буфер приймача заповнений (два значення). В UART прапорець встановлюється в «1», якщо нові дані будуть поміщені у регістр зсуву приймача до того, як із регістра даних буде зчитано попереднє слово. Прапорець скидається при пересиланні прийнятих даних із регістра зсуву приймача у буфер

Продовження таблиці 2.5

Розряд	Назва	Опис
2	UPE (UPE _n)	<u>Прапорець помилки контролю біта парності</u> Прапорець встановлюється в «1», якщо в даних, що знаходяться в буфері приймача, виявлена помилка контролю парності. При вимкненому контролі парності цей розряд постійно скинутий в «0»
1	U2X (U2X _n)	<u>Подвоєння швидкості обміну</u> Якщо цей розряд встановлено в «1», коефіцієнт ділення попереднього дільника контролера швидкості передачі зменшується з 16 до 8, подвоюючи тим самим швидкість асинхронного обміну по послідовному каналу. У USART розряд U2X (U2X _n) використовується тільки при асинхронному режимі роботи. У синхронному режимі він має бути скинутий
0	MPCM (MPCM _n)	<u>Режим мультипроцесорного обміну</u> Розряд MPCM використовується у режимі мультипроцесорного обміну. Якщо він встановлений в «1», ведений мікроконтролер очікує прийому кадру, що містить адресу. Кадри, що не містять адреси пристрою, ігноруються (див. підрозділ 2.3)

Примітка: n = 0, 1, 2 або 3

Таблиця 2.6 – Розряди регістрів UCSRB (UCSR_nB)

Розряд	Назва	Опис
7	RXCIE (RXCIE _n)	<u>Дозвіл переривання після завершення прийому</u> Якщо даний розряд встановлено в «1», то при встановленні прапорця RXC (RXC _n) регістра UCSRA (UCSR _n A) генерується переривання «прийом завершено» (якщо прапорець I регістра SREG встановлено в «1»)
6	TXCIE (TXCIE _n)	<u>Дозвіл переривання після завершення передачі</u> Якщо даний розряд встановлено в «1», то при встановленні прапорця TXC (TXC _n) регістра UCSRA (UCSR _n A) генерується переривання «передачу завершено» (якщо прапорець I регістра SREG встановлено в «1»)
5	UDRIE (UDRIE _n)	<u>Дозвіл переривання при очищенні регістра даних UART.</u> Якщо даний розряд встановлено в «1», то при встановленні прапорця UDRE у регістрі UCSRA (UCSR _n A) генерується переривання «регістр даних пустий» (якщо прапорець I регістра SREG встановлено в «1»)

Продовження таблиці 2.6

Розряд	Назва	Опис
4	RXEN (RXENn)	<u>Дозвіл прийому</u> Якщо даний розряд встановлюється в одиницю, то дозволяється робота приймача та перевизначається функціонування виводу RXD (RXDn). При скиданні розряду RXEN (RXENn) робота приймача забороняється, а його буфер очищується. Значення прапорців RXC (RXCn), DOR/OR, FE (Fen) при цьому стають недійсними
3	TXEN (TXENn)	<u>Дозвіл передачі</u> При встановленні цього розряду в «1» дозволяється робота передавача UART/USART та перевизначається функціонування виводу TXD (TXDn). Якщо розряд скидається в «0» під час передачі, відключення передавача відбудеться тільки після закінчення передачі даних, що знаходяться у регістрі зсуву та буфері передавача
2	CHR9 (CHR9n) UCSZ2 (UCSZn2)	<u>Формат пакетів</u> Цей розряд використовується для задання розміру слів даних, що передаються по послідовному каналу. У модулях USART він використовується разом з розрядами UCSZ1:0 (UCSZn1:0) регістра UCSRC (UCSRnC). У модулях UART, якщо розряд CHR9 (CHR9n) встановлено в «1», здійснюється передача та прийом 9-розрядних даних, якщо скинуто - 8-розрядних
1	RXB8 (RXB8n)	<u>8-й розряд даних, що приймаються</u> При використанні 9-розрядних слів даних цей розряд містить значення старшого розряду прийнятого слова. У випадку USART вміст цього розряду має бути зчитано до читання регістра даних UDR
0	TXB8 (TXB8n)	<u>8-й розряд даних, що передаються</u> При використанні 9-розрядних слів даних, вміст цього розряду є старшим розрядом слова, що передається. Необхідне значення має бути занесене у цей розряд до завантаження байта даних у регістр UDR

Примітка: n = 0, 1, 2 або 3.

Зверніть увагу на те, що в моделях ATmega48x/88x/168x, ATmega164x/324x/644x і ATmega640x/1280x/1281x/2560x/2561x для задання режиму роботи використовується не один розряд регістра UCSRnC, а два – UMSELn1:0. Залежність режиму роботи модуля USART від встановлення цих розрядів наведена в таблиці 2.8.

Таблиця 2.7 – Розряди регістрів UCSRC (UCSRnC)

Розряд	Назва	Опис		
7	URSELn1	<u>Режим роботи USART</u> (моделі 48х/88х/168х, 164х/324х/644х і 640х/1280х/1281х/2560х/2561х) Разом з розрядом UMSELn0 визначає режим роботи модуля USART		
	URSEL* (URSELn)	<u>Вибір регістра</u> Цей розряд визначає, у який з регістрів модуля здійснюється запис. Якщо розряд встановлено в «1», звернення здійснюється до регістра UCSRC (UCSRnC). Якщо ж розряд скинуто в «0», звернення здійснюється до регістра UBRRH (UBRRnH). Детальніше - див. наступний підрозділ		
6	UMSEL (UMSELn)	<u>Режим роботи USART</u> Якщо розряд скинуто в «0», модуль USART працює в асинхронному режимі. Якщо розряд встановлено в «1», то модуль USART працює у синхронному режимі		
	UMSELn0	<u>Режим роботи USART</u> (моделі 48х/88х/168х, 164х/324х/644х і 640х/1280х/1281х/2560х/2561х) Разом з розрядом UMSELn1 визначає режим роботи модуля USART		
5	UPM1 (UPMn1)	<u>Режим роботи схеми контролю та формування парності</u> 5-й та 4-й розряди визначають функціонування схем контролю та формування парності (таблиця 2.12)		
4	UPM0 (UPMn0)			
3	USBS (USBSn)	<u>Кількість стоп-бітів</u> Цей розряд визначає кількість стоп-бітів, що пересилаються передавачем. Якщо розряд скинуто в «0», передавач посилає 1 стоп-біт, якщо встановлено в «1», то 2 стоп-біти. Для приймача вміст цього розряду не має значення		
2	UCSZ1 (UCSZn1)	<u>Формат кадру</u> Разом з розрядом UCSZ2 (UCSZn2) ці розряди визначають кількість розрядів даних та посилки (розмір слова) (таблиця 2.11)		
1	UCSZ0 (UCSZn0)			
0	UCPOL (UCPOLn)	<u>Полярність тактового сигналу</u> Значення цього розряду визначає момент видачі та зчитування даних на виводах модуля. Розряд використовується тільки при роботі у синхронному режимі. При роботі в асинхронному режимі він має бути скинутий у «0»		
		UCPOL (UCPOLn)	Видача даних на вивід TXD (TXDn)	Зчитування даних з виводу RXD (RXDn)
		0	Наростаючий фронт ХСК (ХСКn)	Спадаючий фронт ХСК (ХСКn)
		1	Спадаючий фронт ХСК (ХСКn)	Наростаючий фронт ХСК (ХСКn)

* – зарезервований у моделях ATmega 64х/128х

Примітка: n = 0, 1, 2 чи 3.

Таблиця 2.8 – Керування режимом роботи модулів USART моделей
 ATmega48x/88x/168x, ATmega164x/324x/644x і
 ATmega640x/1280x/1281x/2560x/2561x

UMSELn1	UMSELn0	Режим роботи
0	0	Асинхронний
0	1	Синхронний
1	0	Зарезервовано
1	1	Ведучий шини SPI

2.2.2 Швидкість прийому/передачі

В асинхронному режимі, а також у синхронному режимі при роботі у якості ведучого, швидкість прийому та передачі даних задається контролером швидкості передачі, що функціонує як дільник системного тактового сигналу з програмованим коефіцієнтом ділення. Коефіцієнт визначається вмістом регістра контролера UBRR (UBRRn). У блок приймача сформований сигнал надходить одразу, а у блок передавача - через додатковий дільник, коефіцієнт ділення якого (2, 8 чи 16) залежить від режиму роботи модуля USART/UART.

На рисунку 2.5 у якості прикладу наведено структурну схему блока синхронізації модуля УСАПП для мікроконтролера Mega 128 [1, 2,13].

Опис сигналів:

- txclk – синхронізація передавача (внутрішній сигнал);
- rxclk – синхронізація приймача (внутрішній сигнал);
- xski – вхідний сигнал від виводу ХСК (внутрішній сигнал), який використовується для синхронної підлеглої (веденої) роботи;
- xsko – вихідний сигнал синхронізації до виводу ХСК (внутрішній сигнал). Використовується у ведучому синхронному режимі;

– f_{CLK} – частота системного тактового генератора.

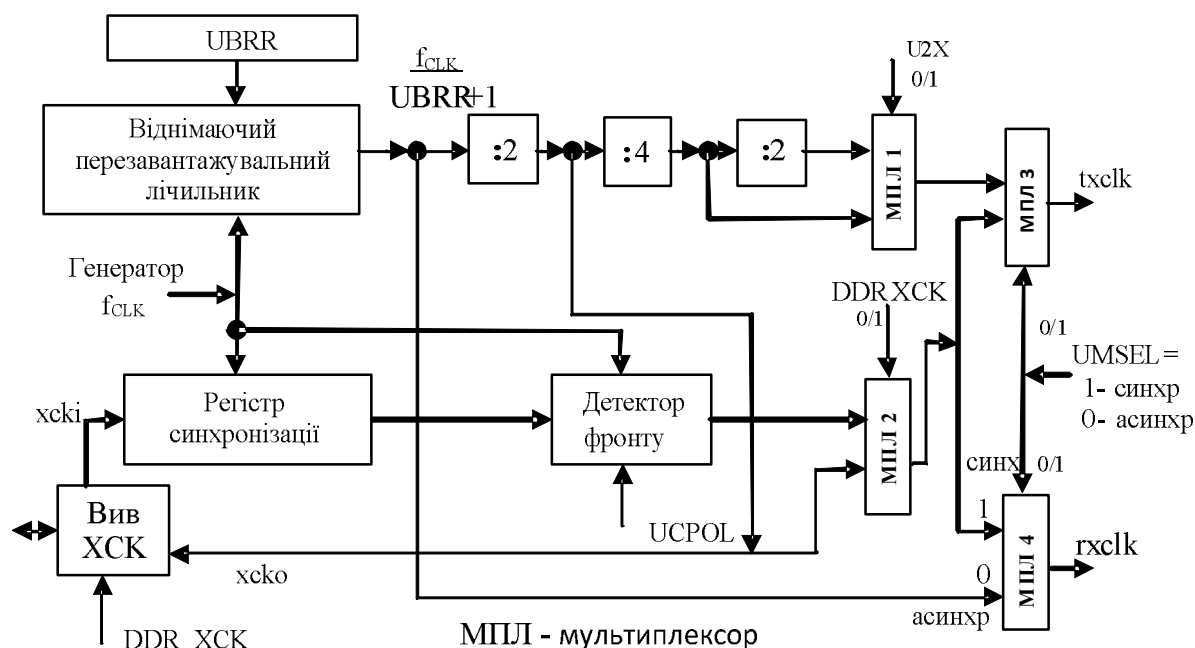


Рисунок 2.5 – Структурна схема блоку синхронізації УСАПП

УСАПП підтримує чотири режими роботи синхронізації: нормальна асинхронна, асинхронна з подвоєнням швидкості, ведуча синхронна та підлегла (ведена) синхронна. Біт $UMSEL$ у регістрі керування та статусу ($UCSRC$) дозволяє обрати асинхронну (0) чи синхронну (1) роботу. Подвоєння швидкості (тільки у асинхронному режимі) керується бітом $U2X$ у регістрі $UCSRA$. При використанні синхронного режиму ($UMSEL = 1$) відповідний біт у регістрі напряму даних для виводу XCK (DDR_XCK) задає, чи буде синхронізація внутрішньою: $DDR\_XCK = 1$ (ведучий режим) чи зовнішньою: $DDR\_XCK = 0$ (ведений режим). Вивід XCK активний тільки при використанні синхронного режиму.

Внутрішня синхронізація використовується для асинхронного та ведучого синхронного режимів роботи. Зовнішня синхронізація використовується у синхронному підлеглому (веденому) режимі роботи.

Регістр UBRR є 12-розрядним та фізично розміщений у двох регістрах введення/виведення. Адреси і назви цих регістрів для деяких моделей мікроконтролерів наведено у таблиці 2.9.

Потрібно мати на увазі, що у моделях ATmega8x, ATmega8515x, ATmega16x, ATmega162x, ATmega32x та ATmega323x регістр UBRRH (UBRRnH) розміщується за тією ж адресою, що й регістр керування UCSRC (UCSRnC). Через це при зверненні за цими адресами необхідно виконати ряд додаткових дій для вибору конкретного регістра.

Таблиця 2.9 – Розміщення регістрів контролера швидкості передачі

Модель	Регістри	Адреса
ATmega8515x/8535x ¹⁾	UBRRH:UBRRL	\$20 (\$40):\$09 (\$29)
ATmega8x/16x/32x ¹⁾	UBRRH:UBRRL	\$20 (\$40):\$09 (\$29)
ATmega64x/128x	UBRR0H:UBRR0L	(\$90):\$09 (\$29)
	UBRR1H:UBRR1L	(\$98):(\$99)
ATmega48x/88x/168x	UBRR0H:UBRR0L	(\$C5):(\$C4)
ATmega162x ¹⁾	UBRR0H:UBRR0L	\$20 (\$40):\$09 (\$29)
	UBRR1H:UBRR1L	\$3C (\$5C):\$00 (\$20)
ATmega164x/324x/644x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)
ATmega165x/325x/3250x/645x/6450x	UBRR0H:UBRR0L	(\$C5):(\$C4)
¹⁾ У цих моделях регістр UBRRH (UBRRnH) розміщується за тією ж адресою, що й регістр UCSRC (UCSRnC)		

Продовження таблиці 2.9

Модель	Регістри	Адреса
ATmega640x/1280x/2560x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)
	UBRR2H:UBRR2L	(\$D5):(\$D4)
	UBRR3H:UBRR3L	(\$135):(\$134)
ATmega1281x/2561x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)

При записі реєстр визначається станом старшого розряду значення, що записується у реєстр UCSRC (UCSRnC), URSEL (URSELn). Якщо цей розряд скинуто у «0», змінюється вміст реєстра UBRRH (UBRRnH). Якщо ж старший розряд значення встановлено в «1», змінюється вміст реєстра керування UCSRC (UCSRnC). Наведені нижче фрагменти програм ілюструють сказане.

Приклад на асемблері

```

...
;Записати 2 у UBRRH
ldi r16, 0x02
out UBRRH, r16

;Встановити розряди URSEL, USBS та UCSZ1 реєстра UCSRC
ldi r16, (1<<URSEL) | (1<<USBS) | (1<<UCSZ1)
out UCSRC, r16
...

```

Приклад на С

```
...
/* Записати 2 у UBRRH */
UBRRH = 0x02
/* Встановити розряди URSEL, USBS та UCSZ1 регістра UCSRC
*/
UCSRC = (1<<URSEL) | (1<<USBS) | (1<<UCSZ1);
...
```

Для вибору регістра при читанні використовується відповідна часова послідовність. При першому зверненні за вказаними адресами повертається значення регістра UBRRH (UBRRHn). При повторному зверненні за цими адресами у наступному машинному циклі повертається значення регістра UCSRC (UCSRnC), як показано у наведених нижче прикладах. Переривання при виконанні цієї послідовності команд мають бути заборонені.

Приклад на асемблері

```
USART_ReadUCSRC:    ;Прочитати регістр UCSRC
in r16,UBRRH        ; r16=UBRRH
in r16,UCSRC        ; Значення регістра UCSRC повертається
;у регістр r16
ret ;Повернення з підпрограми.
```

Приклад на С

```
unsigned char USART_ReadUCSRC(void)
{
    unsigned char ucsrc;
    ucsrc = UBRRH;
```

```

    ucsrc = UCSRC;
    return ucsrc;
}

```

При роботі в асинхронному режимі швидкість обміну визначається не тільки вмістом регістра UBRR, але й станом розряду U2X (U2Xn) регістра UCSRA (UCSRnA). Якщо цей розряд встановлено в «1», коефіцієнт ділення попереднього дільника зменшується у два рази, а швидкість обміну відповідно подвоюється. При роботі у синхронному режимі цей розряд має бути скинуто.

Отже, швидкість обміну визначається наступними формулами:

- асинхронний режим (звичайний, U2Xn = «0»)

$$BAUD = f_{clk} / (16(UBRR+1));$$

- асинхронний режим (пришвидшений, U2Xn = «1»)

$$BAUD = f_{clk} / (8(UBRR+1));$$

- синхронний режим (ведучий)

$$BAUD = f_{clk} / (2(UBRR+1)),$$

де BAUD - швидкість передачі у бодах,

f_{clk} - тактова частота мікроконтролера,

UBRR - вміст регістра контролера швидкості передачі (0...4095).

В якості прикладу в таблиці 2.10 наведено значення регістра UBRR, що дозволяють отримати стандартні для асинхронного режиму швидкості передачі при використанні різних резонаторів, а також величини похибок значень, що отримуються, відносно стандартних швидкостей.

При максимальному значенні швидкості UBRR = 0, похибка = 0.0%. Рекомендується використовувати значення регістра UBRR, при яких отримувана швидкість передачі відрізняється від необхідного значення менше ніж на 0.5% (у таблиці 2.10 виділені жирним шрифтом). Значення,

що дають більше відхилення, також можна використовувати, проте слід мати на увазі, що при цьому знижується завадозахищеність лінії передачі.

При роботі в синхронному режимі в якості веденого швидкість прийому та передачі визначається частотою сигналу, що поступає на вивід ХСК (ХСК_n). Частота цього сигналу має задовольняти виразу $f_{\text{ХСК}} < f_{\text{CLK}}/4$. Це обмеження пов'язано з тим, що сигнал з виводу ХСК (ХСК_n) спочатку синхронізується з тактовою частотою мікроконтролера, а потім проходить через детектор фронтів. Затримка сигналу при проходженні цих вузлів дорівнює двом тактам.

Таблиця 2.10 - Приклад значень регістра UBRR

Швидкість [бод]	1.0000 МГц				1.8432 МГц				2.0000 МГц				3.6864 МГц				4.0000 МГц				7.3728 МГц			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]
2400	25	0.2	51	0.2	47	0.0	95	0.0	51	0.2	103	0.2	95	0.0	191	0.0	103	0.2	207	0.2	191	0.0	383	0.0
4800	12	0.2	25	0.2	23	0.0	47	0.0	25	0.2	51	0.2	47	0.0	95	0.0	51	0.2	103	0.2	95	0.0	191	0.0
9600	6	7.5	12	0.2	11	0.0	23	0.0	12	0.2	25	0.2	23	0.0	47	0.0	25	0.2	51	0.2	47	0.0	95	0.0
14400	3	7.8	8	-3.5	7	0.0	15	0.0	8	-3.5	16	2.1	15	0.0	31	0.0	16	2.1	34	-0.8	31	0.0	63	0.0
19200	2	7.8	6	-7.0	5	0.0	11	0.0	6	-7.0	12	0.2	11	0.0	23	0.0	12	2.2	25	0.2	23	0.0	47	0.0
28800	1	7.8	3	8.5	3	0.0	7	0.0	3	8.5	8	-3.5	7	0.0	15	0.0	8	-3.5	16	3.1	15	0.0	31	0.0
38400	1	22.9	2	8.5	2	0.0	5	0.0	2	8.5	6	-7.0	5	0.0	11	0.0	6	-7.0	12	0.2	11	0.0	23	0.0
57600	0	7.8	1	8.5	1	0.0	3	0.0	1	8.5	3	8.5	3	0.0	7	0.0	3	8.5	8	-3.5	7	0.0	15	0.0
76800	—	—	1	-18.6	1	-25.0	2	0.0	1	-18.6	2	8.5	2	0.0	5	0.0	2	8.5	6	-7.0	5	0.0	11	0.0
115200	—	—	0	8.5	0	0.0	1	0.0	0	8.5	1	8.5	1	0.0	3	0.0	1	8.5	3	8.5	3	0.0	7	0.0
230400	—	—	—	—	—	—	0	0.0	—	—	—	—	0	0.0	1	0.0	0	8.5	1	8.5	1	0.0	3	0.0
250000	—	—	—	—	—	—	—	—	—	—	0	0.0	0	-7.8	1	-7.8	0	0.0	1	0.0	1	-7.8	3	-7.8
0.5M	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	-7.8	—	—	0	0.0	0	-7.8	1	-7.8
1.0M	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	-7.8
Max ¹⁾	62.5K	125K	115.2K	230.4K	125K	250K	230.4K	460.8K	250K	0.5M	460.8K	921.6K												

¹⁾ При UBRR = 0 і похибці 0.0%

Продовження таблиці 2.10

Швидкість [бод]	8.0000 МГц				11.0592 МГц				14.7456 МГц				16.0000 МГц				18.4320 МГц				20.0000 МГц			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]
2400	207	0.2	416	-0.1	287	0.0	575	0.0	383	0.0	767	0.0	416	-0.1	832	0.0	479	0.0	959	0.0	520	0.0	1041	0.0
4800	103	0.2	207	0.2	143	0.0	287	0.0	191	0.0	383	0.0	207	0.2	416	-0.1	239	0.0	479	0.0	259	0.2	520	0.0
9600	51	0.2	103	0.2	71	0.0	143	0.0	95	0.0	191	0.0	103	0.2	207	0.2	119	0.0	239	0.0	129	0.2	259	0.2
14400	34	-0.8	68	0.6	47	0.0	95	0.0	63	0.0	127	0.0	68	0.6	138	-0.1	79	0.0	159	0.0	86	-0.2	173	-0.2
19200	25	0.2	51	0.2	35	0.0	71	0.0	47	0.0	95	0.0	51	0.2	103	0.2	59	0.0	119	0.0	64	0.2	129	0.2
28800	16	2.1	34	-0.8	23	0.0	47	0.0	31	0.0	63	0.0	34	-0.8	68	0.6	39	0.0	79	0.0	42	0.9	86	-0.2
38400	12	0.2	25	0.2	17	0.0	35	0.0	23	0.0	47	0.0	25	0.2	51	0.2	29	0.0	59	0.0	32	-1.4	64	0.2
57600	8	-3.5	16	2.1	11	0.0	23	0.0	15	0.0	31	0.0	16	2.1	34	0.8	19	0.0	39	0.0	21	-1.4	42	0.9
76800	6	-7.0	12	0.2	8	0.0	17	0.0	11	0.0	23	0.0	12	0.2	25	0.2	14	0.0	29	0.0	15	1.7	32	-1.4
115200	3	8.5	8	-3.5	5	0.0	11	0.0	7	0.0	15	0.0	8	-3.5	16	2.1	9	0.0	19	0.0	10	-1.4	21	-1.4
230400	1	8.5	3	8.5	2	0.0	5	0.0	3	0.0	7	0.0	3	8.5	8	-3.5	4	0.0	9	0.0	4	8.5	10	-1.4
250000	1	0.0	3	0.0	2	-7.8	5	-7.8	3	-7.8	6	5.3	3	0.0	7	0.0	4	-7.8	8	2.4	4	0.0	9	0.0
0.5M	0	0.0	1	0.0	—	—	2	-7.8	1	-7.8	3	-7.8	1	0.0	3	0.0	—	—	4	-7.8	—	—	4	0.0
1.0M	—	—	0	0.0	—	—	—	—	0	-7.8	1	-7.8	0	0.0	1	0.0	—	—	—	—	—	—	—	—
Max ¹⁾	0.5M	1.0M	691.2K	1.3824M	921.6K	1.8432M	1.0M	2.0M	1.152M	2.304M	1.25M	2.5M												
¹⁾ При UBRR = 0 і похибці 0.0%																								

¹⁾ При UBRR = 0 і похибці 0.0%

2.2.3 Формат кадру

Під кадром у даному випадку розуміється сукупність одного слова (байта) даних і додаткової інформації (рисунок 2.6). Кадр починається із старт-біта, за яким слідує молодший розряд слова даних. Після старшого розряду слова даних слідує один або два стоп-біта. Якщо включена схема формування біта парності, то він розміщується між старшим розрядом слова даних і першим стоп-бітом.

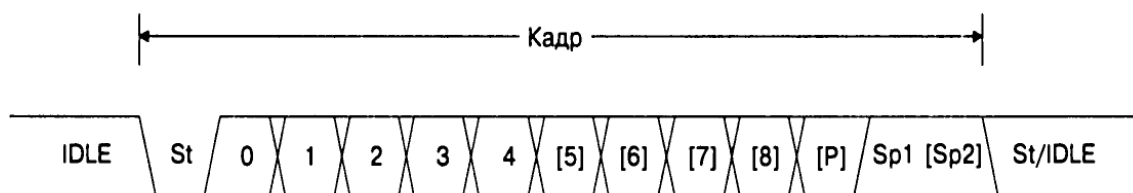


Рисунок 2.6 – Формат кадру

На рисунку 2.6 використані наступні скорочення:

- St – старт-біт, завжди має низький рівень;
- 0...8 – номер біта даних;
- P – біт паритету: парність або непарність;
- Sp1, Sp2 – стоп-біт, завжди має високий рівень;
- IDLE – стан очікування, в якому призупинена передача лінією RxD чи TxD. У стані очікування на лінії має бути високий рівень.

Формат кадру визначається кількома розрядами регістрів UCSRB (UCSRnB) та UCSRC (UCSRnC). Розмір слова даних в USART визначається розрядами UCSZ2...UCSZ0 (UCSZn2...UCSZn0) відповідно до таблиці 2.11.

Таблиця 2.11 – Визначення розміру слова даних у модулях USART

UCSZ2 (UCSZn2)	UCSZ1 (UCSZn1)	UCSZ0 (UCSZn0)	Розмір слова даних
0	0	0	5 розрядів
0	0	1	6 розрядів
0	1	0	7 розрядів
0	1	1	8 розрядів
1	0	0	Зарезервовано
1	0	1	Зарезервовано
1	1	0	Зарезервовано
1	1	1	9 розрядів

Примітка: n = 0, 1, 2 чи 3.

У модулях UART слово даних може бути тільки 8- або 9-розрядним, що визначається станом прапорця CHR9 (CHR9n) регістра UCSRB (UCSRnB). Якщо цей прапорець скинутий в «0», розмір слова дорівнює 8 розрядам, якщо встановлений в «1» — 9 розрядам.

Вибір кількості стоп-бітів в модулях USART здійснюється за допомогою розряду USBS (USBSn) регістра UCSRC (UCSRnC). Якщо цей розряд скинутий в «0», блок передавача формує 1 стоп-біт у кінці посилки. Якщо розряд встановлений в «1», блок передавача формує 2 стоп-біти. Слід зазначити, що приймачем другий стоп-біт ігнорується, і, відповідно, помилки кадрування виявляються тільки для першого стоп-біта.

Розряди UPM1:UPM0 (UPMn1:UPMn0) регістра UCSRC (UCSRnC) визначають функціонування схеми контролю парності модулів USART відповідно до таблиці 2.12.

Таблиця 2.12 – Керування контролем парності

UPM1 (UPMn1)	UPM0 (UPMn0)	Режим роботи
0	0	Відключений
0	1	Зарезервовано
1	0	Включений, перевірка на парність (even parity)
1	1	Включений, перевірка на непарність (odd parity)

Примітка: n = 0, 1, 2 чи 3.

Значення біта парності отримується шляхом виконання операції «виключаючого АБО» над усіма розрядами слова даних, яке передається. Якщо використовується перевірка на непарність (odd parity), отриманий результат інвертується:

$$P_{EVEN} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0,$$

$$P_{ODD} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1.$$

Якщо контроль парності включений, біт парності, як вже було сказано, вставляється передавачем між старшим розрядом даних, які передаються, і першим стоп-бітом.

2.2.4 Передача даних

Робота передавача дозволяється встановленням в «1» розряду TXEN (TXENn) регістра UCSRB (UCSRnB). При встановленні цього розряду вивід TXD (TXDn) підключається до передавача USART/UART і починає функціонувати, як вихід, незалежно від значень у регістрах керування портом. Якщо використовується синхронний режим роботи, треба програмно перевизначити функціонування виводу XCK (XCKn) (введення або виведення).

Передача ініціюється записом даних, які передаються, у буферний регістр передавача - регістр даних UDRn. Після цього дані пересилаються з регістра UDRn у регістр зсуву передавача. Одночасно, якщо використовуються 9-розрядні дані, значення розряду TXB8 (TXB8n) регістра UCSRB (UCSRnB) копіюється у 9-й розряд регістра зсуву. При цьому можливі два варіанти:

- запис у регістр UDR здійснюється в той момент, коли передавач знаходиться у стані очікування (попередні дані вже передані). У цьому випадку дані пересилаються у регістр зсуву одразу ж після запису в регістр UDR;

- запис у регістр UDR здійснюється під час передачі. В цьому випадку дані пересилаються у регістр зсуву після передачі останнього стоп-біта поточного кадру.

Очевидно, що 9-й розряд даних повинен бути завантажений в розряд TXB8 (TXB8n) до запису байта даних в регістр даних.

Після пересилання слова даних в регістр зсуву, прапорець UDRE (UDREN) регістра UCSRA (UCSRnA) встановлюється в «1», що означає готовність передавача до отримання нового байта даних. У цьому стані прапорець залишається до наступного запису в буфер. Одночасно з пересиланням у регістрі формується службова інформація - старт-біт, можливий біт парності (тільки в USART), а також один або два стоп-біти.

Після завантаження регістра зсуву його вміст починає зсуватися вправо і поступати на вивід TXD (TXDn) у порядку, розглянутому в підрозділі 2.2.3. Швидкість зсуву визначається налаштуванням контролера тактових сигналів. При роботі в синхронному режимі зміна стану виводу TXD (TXDn) відбувається за одним з фронтів сигналу XCK (XCKn). Якщо розряд UCPOL (UCPOLn) регістра UCSRC (UCSRnC) скинутий в «0», зміна стану виводу відбувається за наростаючим фронтом сигналу XCK (XCKn), якщо ж встановлений в «1» - за спадаючим фронтом сигналу, як показано на рисунку 2.7.

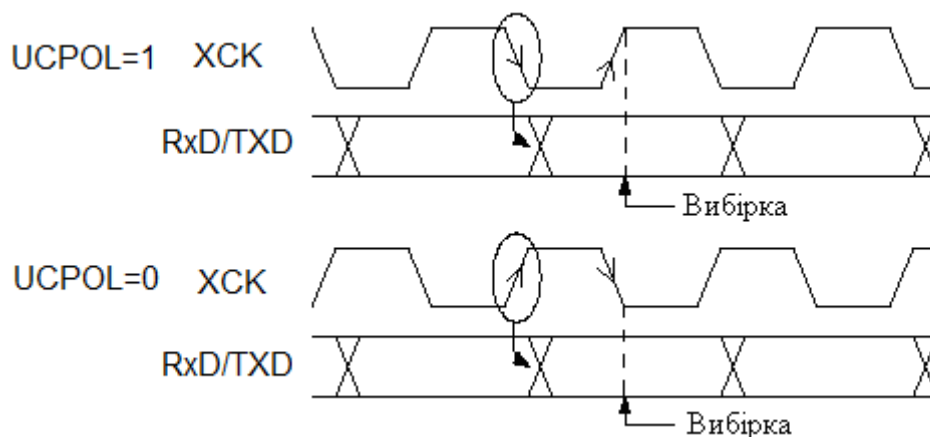


Рисунок 2.7 – Часові діаграми для синхронного режиму роботи USART

Якщо під час передачі у регістр UDR було записано нове слово даних, то після передачі останнього стоп-біта попереднього слова нове слово пересилається в регістр зсуву. Якщо ж до моменту закінчення

передачі кадру такого запису виконано не було, то встановлюється прапорець переривання «Передача завершена» TXC (TXCn) регістра UCSRA (UCSRnA). Скидання прапорця здійснюється апаратно при вході у підпрограму обробки відповідного переривання або програмно, записом в цей розряд лог. 1.

Відключення передавача здійснюється скиданням розряду TXEN (TXENn) регістра UCSRB (UCSRnB). Якщо у момент виконання цієї команди здійснювалася передача, скидання розряду відбудеться тільки після завершення поточної та відкладеної передач, тобто після очищення буферного регістра та регістра зсуву передавача. При вимкненому передавачеві вивід TXD (TXDn) може використовуватися, як лінія введення/виведення загального призначення.

Нижче наведений приклад підпрограми передачі інтерфейсом USART/UART. Ця підпрограма чекає на очищення буфера передавача, а потім завантажує в нього нове значення.

Приклад на асемблері

```
USART_Transmit:
;Чекати очистки буфера передавача
sbis UCSRA,UDRE
rjmp USART_Transmit
;Скопіювати 9-й розряд даних з R17 в TXB8
cbi UCSRB,TXB8
sbrc R17,0
sbi UCSRB,TXB8
;Завантажити байт даних у буфер, почати передачу
out UDR,R16
ret
```

Приклад на C

```
void USART_Transmit(unsigned int data)
{
    /* чекати очистки буфера передавача */
    while ( !( UCSRA & (1<<UDRE)) )
    /* скопіювати 9-й розряд даних з R17 в TXB8 */
    UCSRB &= ~(1<<TXB8);
    if (data & 0x100)
    UCSRB |= (1<<TXB8) ;
    /* завантажити байт даних у буфер, почати передачу */
    UDR = data
}
```

2.2.5 Прийом даних

Робота приймача дозволяється встановленням розряду RXEN (RXENn) регістра UCSRB (UCSRnB). При встановленні цього розряду вивід RXD (RXDn) підключається до приймача USART/UART і починає функціонувати як вхід, незалежно від значень регістрів керування портом. Якщо використовується синхронний режим роботи, треба програмно перевизначити функціонування виводу XCK (XCKn).

Прийом даних починається відразу ж після виявлення приймачем коректного старт-біта. Кожен розряд вмісту кадру потім зчитується з частотою, яка визначається контролером швидкості передачі або тактовим сигналом XCK (XCKn). Зчитані розряди даних послідовно розміщуються у регістр зсуву приймача до виявлення першого стоп-біта кадру. Після цього вміст регістра зсуву пересилається у буфер приймача, з якого прийняте значення може бути отримане шляхом читання регістра даних модуля. При використанні 9-розрядних слів даних значення старшого розряду може бути визначене за станом прапорця RX8 (RX8n) регістра UCSRB (UCSRnB). Причому у модулях USART вміст старшого розряду даних

(регістр UCSRB/UCSRnB) має бути зчитаний до звернення до регістра даних. Це пов'язано з тим, що прапорець RX8 (RX8n) відображає значення старшого розряду слова даних кадру, який знаходиться на верхньому рівні буфера приймача, стан якого при читанні регістра даних зміниться.

Якщо була включена схема контролю парності (тільки USART), вона обчислює біт парності для всіх розрядів прийнятого слова даних і порівнює його з прийнятим бітом парності. Результат перевірки запам'ятовується у буфері приймача разом з прийнятим словом даних і стоп-бітами. Наявність або відсутність помилки контролю парності може бути потім визначена за станом прапорця UPE (UPEn) регістра UCSRA (UCSRnA). Цей прапорець встановлюється в «1», якщо наступне слово, яке може бути прочитане з буфера, має помилку контролю парності. При вимкненому контролі парності прапорець UPE (UPEn) завжди читається як «0».

Блок приймача модулів USART/UART в регістрі UCSRA(UCSRnA) має ще два прапорці, які показують стан обміну: прапорець помилки кадрування FE (FEn) і прапорець переповнення DOR (DORn) / OR (ORn). Прапорець FE (FEn) встановлюється в «1», якщо значення першого стоп-біта прийнятого кадру не відповідає потрібному, тобто дорівнює «0».

Прапорці DOR (DORn) в USART та OR (ORn) в UART відображають втрату даних через переповнення буфера приймача. В UART прапорець встановлюється в «1», якщо до моменту закінчення прийому кадру (заповнення регістра зсуву приймача) дані попереднього кадру не були зчитані з регістра даних. В USART прапорець встановлюється в «1» у разі прийому старт-біта нового кадру при заповнених буфері та регістрі зсуву приймача. Встановлений прапорець DOR (DORn) / OR (ORn) означає, що між минулим байтом, зчитаним з регістра UDR, та байтом, зчитаним у даний момент, відбулася втрата одного або декількох кадрів.

Слід мати на увазі, що обробка описаних прапорців в модулях UART і USART дещо відрізняється. У модулях UART прапорець помилки кадрів FE (FEn) має бути прочитаний перед зверненням до регістра даних, а прапорець переповнення OR (ORn) — після звернення до цього регістра.

У модулях USART всі прапорці помилок буферизуються разом із словом даних, тобто відповідні розряди регістра UCSRA (UCSRnA) відносяться до кадру, слово даних якого буде прочитано при наступному зверненні до регістра даних UDR (UDRn). Тому стан цих прапорців має бути зчитаний перед зверненням до регістра даних. Крім того, для сумісності з майбутніми пристроями рекомендується при записі в регістр UCSRA (UCSRnA) скидати відповідні цим прапорцям розряди в «0». Для індикації стану приймача в модулях USART/UART використовується прапорець переривання «Прийом завершений» RXC (RXCn) регістра UCSRA (UCSRnA). Цей прапорець встановлюється в «1» за наявності в буфері приймача непрочитаних даних. У модулях UART цей прапорець скидається після читання регістра даних, а у модулях USART - при звільненні буфера (після зчитування всіх даних, які знаходяться у ньому).

Вимкнення приймача здійснюється скиданням розряду RXEN (RXENn) регістра UCSRB (UCSRnB). На відміну від передавача, приймач вимикається відразу ж після скидання розряду, тобто кадр, який приймається у цей момент, втрачається. У модулях USART, окрім того, при вимкненні приймача очищується його буфер, тобто втрачаються також всі непрочитані дані. При вимкненому приймачі вивід RXD (RXDn) може використовуватися як лінія введення/виведення загального призначення.

Приклад підпрограми прийому з дев'ятьма бітами даних інтерфейсом USART наведений нижче. Як і в попередньому прикладі, тут використовується опитування прапорця переривання.

Приклад на асемблері

```
USART_Receive:
;Чекати завантаження даних у буфер приймача
sbis UCSRA,RXC
rjmp USART_Receive
;Прочитати прапорці стану та 9-й розряд даних
in r18,UCSRA
in r17,UCSRB
;Прочитати молодший байт даних
in r16,UDR
;У випадку помилки повернути -1
andi r18, (1«FE) | (1«DOR) | (1«UPE)
breq no_error
ldi r17,HIGH(-1)
ldi r18,LOW(-1)
no_error:
;Виділити 9-й розряд даних
lsr r17
andi r17,0x01
ret
```

Приклад на C

```
unsigned int USART_Receive(void)
{
    unsigned char status, resh, resl
    /* Чекати заповнення буфера приймача */
    while (! (UCSRA & (1«RXC) ) )
    /* Прочитати 9-й розряд даних та прапорці стану */
    status = UCSRA;
    resh = UCSRB;
    /* Прочитати молодший байт даних */
    resl = UDR;
```



```

/* У випадку помилки повернути -1*/
if (status & (1«FE) | (1«DOR) | (1«UPE) )
return -1;
/* Виділити 9-й розряд даних */
resh = (resh»1) & 0x01;
return ( (resh<<8) | resl);
}

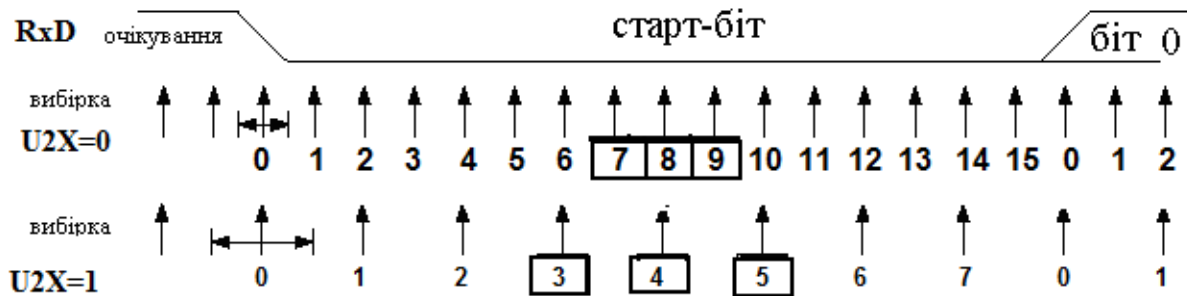
```

Власне прийом всіх розрядів кадру здійснюється по-різному, залежно від режиму роботи модуля. При роботі модуля USART у синхронному режимі стан виводу RXD (RXDn) читається за одним з фронтів сигналу XCK (XCKn). Якщо розряд UCPOL (UCPOLn) регістра UCSRC (UCSRnC) скинуто в «0», зчитування стану виводу RXD відбувається за спадаючим фронтом сигналу XCK (XCKn), якщо ж встановлено в «1» - за наростаючим фронтом сигналу. Іншими словами, зчитування даних з виводу RXD (RXDn) і їх видача на вивід TXD (TXDn) відбуваються за протилежними фронтами (рисунок 2.7).

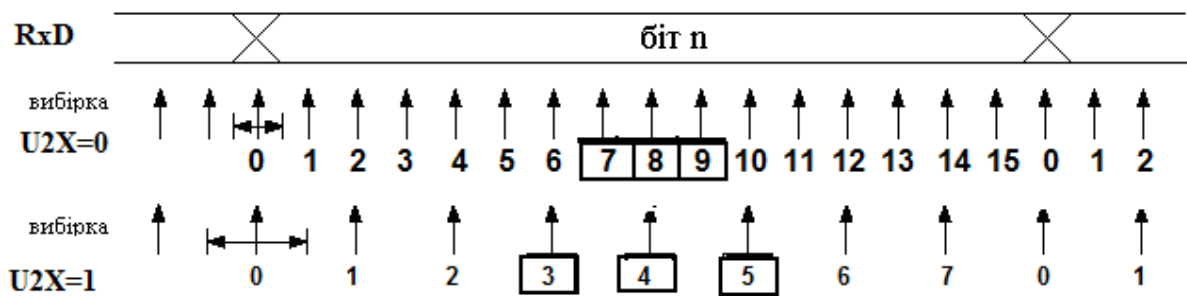
Для забезпечення прийому в асинхронному режимі роботи використовуються схеми відновлення тактового сигналу і даних. Схема відновлення тактового сигналу призначена для синхронізації внутрішнього тактового сигналу, який формується контролером швидкості передачі, і кадрів, які поступають на вивід RXD (RXDn) мікроконтролера. Схема відновлення даних здійснює читання та фільтрацію кожного розряду кадру, що приймається.

Схема відновлення тактового сигналу здійснює опитування входу приймача (вибірку) з метою визначення старт-біта кадру (рисунок 2.8). Частота опитування залежить від стану розряду U2X (U2Xn) регістра UCSRA (UCSRnA). У звичайному режимі (при U2Xn = «0») частота опитування у 16 разів перевищує швидкість передачі даних, а у прискореному режимі (при U2X = «1») - у 8 разів. Горизонтальні стрілки

на рисунку 2.8 ілюструють можливий відхід синхронізації у процесі вибірки (опитування). Як видно з рисунка 2.8, більш високу розсинхронізацію у часі має прискорений обмін (біт U2X=1).



а)



б)

Рисунок 2.8 – Розпізнавання розрядів кадру:

а) - старт-біт; б) - інші розряди

Виявлення зміни сигналу на виводі RXD (RXDn) з лог. 1 (режим очікування) на лог. 0 інтерпретується як можлива поява переднього фронту старт-біта. Коли перехід сигналу на вході RXD з «1» в «0» виявлений, негайно скидається лічильник-дільник на 16 у ланцюзі формування сигналу «Синхр. RXD». В результаті цього відбувається суміщення моментів переповнення цього лічильника-дільника з межами зміни бітів кадру, що приймається, на вході RXD. Шістнадцять станів лічильника-

дільника ділять час, протягом якого кожен біт посилки, що приймається, присутній на вході RXD, на 16 фаз (вибірок), з 0-ї по 15-у для кожного біта. Після цього у нормальному режимі перевіряється значення 7-ої, 8-ої і 9-ої вибірок вхідного сигналу, а в прискореному режимі - 3-ої, 4-ої і 5-ої вибірок (рисунок 2.8, а). Якщо значення хоча б двох вибірок із вказаних дорівнює лог. 1, старт-біт вважається помилковим (завада), а приймач переходить до очікування наступної зміни вхідного сигналу з лог. 1 на лог. 0. Інакше вважається, що виявлено старт-біт нової послідовності, з яким синхронізується внутрішній тактовий сигнал приймача. Після цього починає працювати схема відновлення даних.

Рішення про значення прийнятого розряду ухвалюється також за результатами 7-ої, 8-ої і 9-ої (3-й, 4-й і 5-й) вибірок вхідного сигналу (рисунок 2.8, б). Станом розряду вважається логічне значення, яке було отримане щонайменше у двох з трьох вибірок. Процес розпізнавання повторюється для всіх розрядів кадру, що приймається, включаючи перший стоп-біт.

Із сказаного випливає, що старт-біт нового кадру може передаватись відразу ж після останньої вибірки, яка використовується для визначення значення першого стоп-біта. У звичайному режимі роботи формування старт-біта може початися в момент А, а в прискореному режимі - в момент В (рисунок 2.9). Момент С, позначений на рисунку, визначає максимальну тривалість стоп-біта. Використання більш раннього виявлення старт-біта впливає на робочій діапазон синхрочастот приймача (допустиме розходження синхрочастоти передавача та приймача при фіксованій довжині посилки) [1, 2].

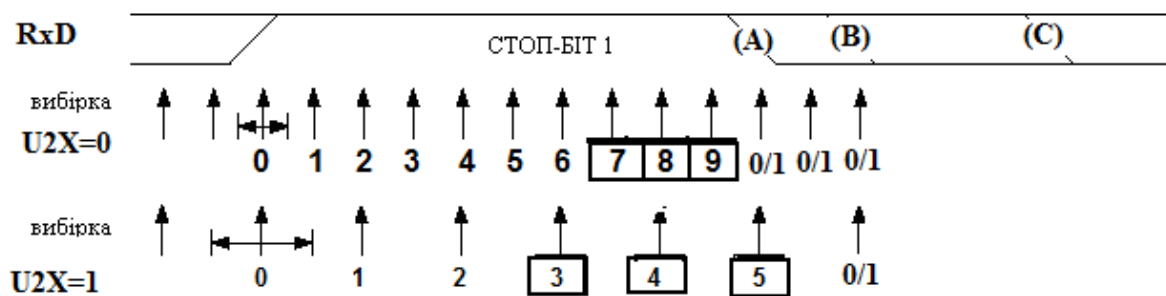


Рисунок 2.9 – Розпізнавання стоп-біта та наступного старт-біта

2.3 Мультипроцесорний режим роботи

Режим мультипроцесорного обміну дозволяє здійснювати зв'язок між декількома веденими мікроконтролерами і одним ведучим. У цьому режимі кожен ведений мікроконтролер має свою унікальну адресу. Якщо ведучий мікроконтролер хоче що-небудь передати, то він посилає адресний байт, який визначає, до якого з мікроконтролерів він збирається звернутися. Якщо який-небудь з ведених мікроконтролерів розпізнав свою адресу, він переходить у режим прийому даних і відповідно приймає подальші байти як дані. Решта ведених мікроконтролерів ігнорують байти, що приймаються, до посилки ведучим нового адресного байта. Увімкнення режиму фільтрації кадрів, що приймаються і не містять адреси, здійснюється встановленням в «1» розряду MPCM (MPCMn) регістра UCSRA (UCSRnA).

У мікроконтролері, який виконує роль ведучого, повинен бути встановлений режим передачі 9-розрядних даних. При передачі адресного байта старший (8-й) розряд повинен встановлюватися в «1», а при передачі байтів даних він повинен скидатися в «0».

У ведених мікроконтролерах механізм прийому залежить від режиму роботи приймача. Якщо приймач налаштований на прийом 5...8-розрядних даних, то ідентифікація вмісту кадру (адрес/дані) здійснюється за 1-м стоп-

бітом. При прийомі 9-розрядних даних ідентифікація вмісту кадру здійснюється за старшим (8-м) розрядом слова даних. Якщо вказані розряди встановлені в «1», то кадр містить адресу, якщо в «0» - кадр містить дані.

Для здійснення обміну даними в багатопроцесорному режимі виконуються наступні дії.

1) Всі ведені мікроконтролери перемикаються в режим мультипроцесорного обміну встановленням в «1» розряду MPCM (MPCMn) регістра UCSRA (UCSRnA).

2) Ведучий мікроконтролер посилає адресний кадр, а всі ведені мікроконтролери його приймають. Відповідно, у кожному з ведених мікроконтролерів встановлюється прапорець RXC (RXCn) регістра UCSRA (UCSRnA).

3) Кожен з ведених мікроконтролерів читає вміст регістра даних. Мікроконтролер, адреса якого співпала з адресою, посланою ведучим, скидає в «0» розряд MPCM (MPCMn).

4) Адресований мікроконтролер починає приймати кадри, що містять дані. Якщо приймач веденого мікроконтролера налаштований на прийом 5...8-розрядних даних, то буде генеруватися помилка кадрування, оскільки перший стоп-біт буде рівний «0» (в USART цього можна уникнути, використовуючи два стоп-біти). У решті ведених мікроконтролерів розряд MPCM (MPCMn) встановлений в «1», через це кадри даних будуть ігноруватися.

5) Після прийому останнього байта даних адресований мікроконтролер встановлює в «1» розряд MPCM (MPCMn) і знову чекає прихід кадру з адресою. Процес повторюється з пункту 2.

2.4 Робота модуля USART в режимі MSPI

2.4.1 Загальні відомості

В ряді нових моделей мікроконтролерів, а саме в ATmega48x/88x/168x, ATmega164x/324x/644x і ATmega640x/1280x/1231x/2560x/2561x, модулі USART можуть працювати у якості ведучого шини SPI (див. розділ 4). В цьому режимі, який називається MSPI (Master SPI Mode), модулі USART підтримують усі чотири режими SPI (0, 1, 2 і 3). При цьому доступні як подвійна буферизація даних при прийомі і передачі, так і більшість переривань модуля. Функції виводів модуля USART в режимі MSPI вказані у таблиці 2.13 [2].

Таблиця 2.13 – Виводи USART в режимі MSPI

USART	SPI	Опис
TXDn	MOSI	Тільки вихід ведучого
RXDn	MISO	Тільки вхід ведучого
XCKn	SCK	Вихід тактового сигналу
-	$\overline{SS}$	Не підтримується

Зверніть увагу, що напрям передачі даних виводу XCKn необхідно задавати програмно (відповідний розряд регістра DDRx повинен бути встановлений в 1). Причому конфігурування виводу бажано виконувати до дозволу роботи приймача і передавача USART.

2.4.2 Керування модулем USART в режимі MSPI

Керування модулями USART в режимі MSPI здійснюється за допомогою тих же регістрів UCSRnA, UCSRnB і UCSRnC, що і в звичайному режимі, змінюється тільки призначення (і назва) деяких

розрядів регістрів UCSRnC. Розряди цих регістрів, які використовуються в режимі MSPI, наведені на рисунках 2.10...2.12, а їх опис – у таблицях 2.14...2.16 відповідно.

	7	6	5	4	3	2	1	0
	RXC _n	TXC _n	UDRE _n	—	—	—	—	—
Зчитування(R)/Запис(W)	R/W	R/W	R/W	R	R	R	R	R
Початкове значення	0	0	1	0	0	0	0	0

Рисунок 2.10 – Регістри UCSRnA в режимі MSPI

Таблиця 2.14 – Розряди регістрів UCSRnA в режимі MSPI

Розряд	Назва	Опис
7	RXC _n	<u>Прапорець завершення прийому</u> Прапорець встановлюється в «1» при пересиланні прийнятого слова з регістра зсуву приймача в регістр даних UDR і утримується в 1 при наявності непрочитаних даних у буфері приймача (регістр даних UDR). Скидається прапорець апаратно після очищення буфера (в UART - після читання регістра даних). Якщо розряд RXCIE (RXCIE _n) регістра UCSRB (UCSRnB) встановлено, то при встановленні прапорця генерується запит на переривання «прийом завершено»
6	TXC _n	<u>Прапорець завершення передачі</u> Прапорець встановлюється в «1» після передачі усіх розрядів пакету із регістра зсуву передавача, за умови, що у регістр даних UDR не було завантажено нового значення. Якщо розряд TXCIE (TXCIE _n) регістра UCSRB (UCSRnB) встановлено, то при встановленні прапорця генерується переривання «передачу завершено». Прапорець скидається апаратно при виконанні підпрограми обробки переривання чи програмно записом у нього лог. 1
5	UDRE _n	<u>Прапорець очищення регістра даних</u> Даний прапорець встановлюється в «1» при пустому буфері передавача (після пересилання байта із регістра даних UDR у регістр зсуву передавача). Встановлений прапорець означає, що у регістр даних можна завантажити нове значення. Якщо розряд UDRIE регістра UCR (UCSRB) встановлено, генерується запит на переривання «регістр даних пустий». Прапорець скидається апаратно, при записі у регістр даних
4..0	-	Не використовуються

Примітка: n = 0, 1, 2 або 3.

	7	6	5	4	3	2	1	0
	RXCIE _n	TXCIE _n	UDRIE _n	RXEN _n	TXEN _n	—	—	—
Зчитування(R)/Запис(W)	R/W	R/W	R/W	R/W	R/W	R	R	R
Початкове значення	0	0	1	0	0	0	1	0

Рисунок 2.11 – Регістри UCSRnB в режимі MSPI

Таблиця 2.15 – Розряди регістрів UCSRnB в режимі MSPI

Розряд	Назва	Опис
7	RXCIE _n	<u>Дозвіл переривання після завершення прийому</u> Якщо даний розряд встановлено в «1», то при встановленні прапорця RXC (RXC _n) регістра UCSRA (UCSRnA) генерується переривання «прийом завершено» (якщо прапорець I регістра SREG встановлено в «1»)
6	TXCIE _n	<u>Дозвіл переривання після завершення передачі</u> Якщо даний розряд встановлено в «1», то при встановленні прапорця TXC (TXC _n) регістра UCSRA (UCSRnA) генерується переривання «передачу завершено» (якщо прапорець I регістра SREG встановлено в «1»)
5	UDRIE _n	<u>Дозвіл переривання при очищенні регістра даних USART</u> Якщо даний розряд встановлено в «1», то при встановленні прапорця UDRE у регістрі UCSRA (UCSRnA) генерується переривання «регістр даних пустий» (якщо прапорець I регістра SREG встановлено в «1»)
4	RXEN _n	<u>Дозвіл прийому</u> Якщо даний розряд встановлюється в одиницю, то дозволяється робота приймача та перевизначається функціонування виводу RXD (RXD _n). При скидані розряду RXEN (RXEN _n) робота приймача забороняється, а його буфер очищається. Значення прапорців RXC (RXC _n), DOR/OR, FE (FE _n) при цьому стають недійсними
3	TXEN _n	<u>Дозвіл передачі</u> При встановленні цього розряду в «1» дозволяється робота передавача UART/USART та перевизначається функціонування виводу TXD (TXD _n). Якщо розряд скидається в «0» під час передачі, відключення передавача відбудеться тільки після закінчення передачі даних, що знаходяться у регістрі зсуву та буфері передавача
2..0	-	Не використовуються

Примітка: n = 0, 1, 2 або 3.

	7	6	5	4	3	2	1	0
	UMSELn1	UMSELn0	—	—	—	UDORDn	UCPHAn	UCPOLn
Зчитування(R)/Запис(W)	R/W	R/W	R	R	R	R/W	R/W	R/W
Початкове значення	0	0	0	0	0	1	1	0

Рисунок 2.12 – Регістри UCSRnC в режимі MSPI

Таблиця 2.16 – Розряди регістрів UCSRnC в режимі MSPI

Розряд	Назва	Опис															
7	URSELn1	<u>Режим роботи USART</u> Ці розряди визначають режим роботи модуля USART:															
6	UMSELn0	<table><tr><th>UMSELn1</th><th>UMSELn0</th><th>Режим роботи</th></tr><tr><td>0</td><td>0</td><td>Асинхронний</td></tr><tr><td>0</td><td>1</td><td>Синхронний</td></tr><tr><td>1</td><td>0</td><td>Зарезервовано</td></tr><tr><td>1</td><td>1</td><td>Ведучий шини MSPI</td></tr></table>	UMSELn1	UMSELn0	Режим роботи	0	0	Асинхронний	0	1	Синхронний	1	0	Зарезервовано	1	1	Ведучий шини MSPI
UMSELn1	UMSELn0	Режим роботи															
0	0	Асинхронний															
0	1	Синхронний															
1	0	Зарезервовано															
1	1	Ведучий шини MSPI															
5..3	-	Не використовуються															
2	UDORDn	<u>Формат передачі даних</u> Якщо цей розряд встановлено в 1, дані передаються від молодшого біта до старшого. Якщо розряд скинуто в 0, порядок передачі розрядів змінюється на протилежний - від старшого до молодшого біта															
1	UCPHAn	<u>Фаза тактового сигналу</u> Якщо цей біт скинуто в 0, дані зі входу RxDn зчитуються за переднім фронтом імпульсів тактового сигналу (наростаючий фронт при UCPOLn = 0 і спадаючий при UCPOLn = 1). Якщо ж цей біт встановлено в 1, дані з входу RxDn зчитуються за заднім фронтом імпульсів тактового сигналу (спадаючий фронт при UCPOLn = 0 і наростаючий при UCPOLn = 1)															
0	UCPOLn	<u>Полярність тактового сигналу</u> Значення цього розряду визначає стан лінії SCK в режимі очікування. Якщо розряд скинуто в 0, генеруються імпульси додатної полярності, а в режимі очікування на виводі XCKn формується НИЗЬКИЙ рівень. Якщо розряд встановлено в 1, генеруються імпульси від'ємної полярності, а в режимі очікування на виводі XCKn формується ВИСОКИЙ рівень															

Примітка: n = 0, 1, 2 або 3.

Як видно з таблиці 2.16, перемикання модуля USART в режим MSPI здійснюється встановленням в 1 обох розрядів UMSELn1 і UMSELn0 регістра UCSRnC. Формування тактового сигналу в режимі MSPI здійснюється аналогічно синхронному режиму USART. Таким чином, швидкість передачі в режимі MSPI визначається виразом:

$$BAUD = \frac{f_{CLK}}{2 \cdot (UBRR + 1)},$$

де BAUD - швидкість передачі, біт/с;

f_{CLK} - тактова частота мікроконтролера;

UBRR - вміст регістра контролера швидкості передачі (0 ... 4095).

Як було сказано вище, модуль USART підтримує всі чотири режими SPI, що розрізняються фазою та полярністю тактового сигналу XCKn (SCK) відносно послідовних даних. В даному випадку фаза тактового сигналу визначається розрядом UCPHAn, а полярність – розрядом UCPOLn регістра UCSRnC. Відповідність між станом цих розрядів і реалізованим режимом SPI вказано у таблиці 2.17. Часові діаграми передачі даних для всіх чотирьох режимів наведені на рисунку 2.13.

Таблиця 2.17 – Значення розрядів UCPHAn і UCPOLn регістрів UCSRnC у різних режимах SPI

UCPOLn	UCPHAn	Режим SPI	Передній фронт	Задній фронт
0	0	0	Зчитування даних (наростаючий фронт)	Встановлення (передача) даних (спадаючий фронт)
0	1	1	Встановлення (передача) даних (наростаючий фронт)	Зчитування даних (спадаючий фронт)
1	0	2	Зчитування даних (спадаючий фронт)	Встановлення (передача) даних (наростаючий фронт)
1	1	3	Встановлення (передача) даних (спадаючий фронт)	Зчитування даних (наростаючий фронт)

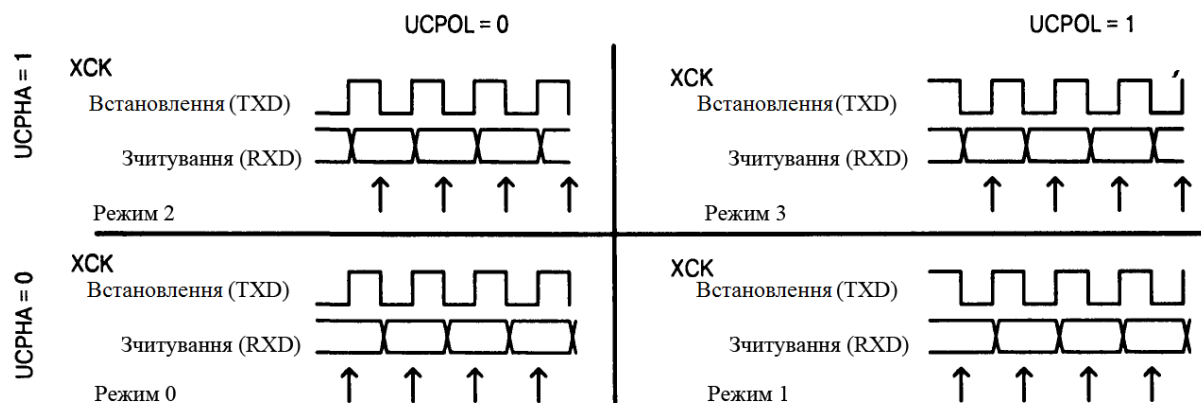


Рисунок 2.13 – Передача даних в режимі MSPI

Формат передачі 8-бітових даних, а саме порядок передачі бітів, визначається станом розряду UDORDn регістра UCSRnC. Якщо цей біт скинутий в 0, першим передається старший біт вмісту регістра зсуву передавача, а якщо встановлено в 1 - молодший біт. Завдяки подвійній буферизації даних в модулі USART, можна легко здійснити передачу і 16-бітних даних, записуючи обидва байта поспіль у регістр даних UDRn. Закінчення передачі 16-бітного значення можна буде визначити за встановленням прапорця переривання «передача завершена».

2.4.3 Ініціалізація режиму MSPI

Ініціалізація режиму MSPI модуля USART повинна бути виконана до початку обміну шиною. Власне ініціалізація полягає у заданні необхідної швидкості передачі, переведенні виводу XCKn в режим виходу, встановленні формату передачі даних, а також включенні передавача і приймача USART (якщо не потрібне отримання даних від ведених пристроїв, то приймач можна і не включати). Якщо при обробці USART використовуються переривання, то під час ініціалізації всі переривання повинні бути заборонені (прапорець I регістра SREG повинен бути скинутий в 0). Зверніть увагу, що включення передавача USART необхідно

виконувати при скинутому регістрі контролера швидкості передачі UBRRn. Відповідно запис в регістр UBRRn необхідного значення слід проводити після дозволу передавача, але до початку обміну шиною.

При реініціалізації модуля (зміні швидкості передачі, режиму роботи або формату даних), перш ніж змінювати вміст регістрів, необхідно переконатися, що всі дані передані, проконтролювавши стан прапорців TXCn і RXCn. Нижче наведено простий приклад підпрограми ініціалізації (використовується опитування прапорців переривань). Параметром підпрограми є значення швидкості передачі даних, які передаються (у прикладі на асемблері у регістровій парі R17:R16).

Приклад на асемблері

```
USART_MSPIM_Init:
    clr r18
    out UBRRnH, r18
    out UBRRnL, r18; Перемикаємо XCKn на вихід
    sbi XCKn_DDR, XCKn; Перемикаємося в режим MSPI (0-й
    ;режим SPI)
    ldi r18, (1<<UMSELn1) | (1<<UMSELn0) | (0<<UCPHAn) |
    (0<<UCPOLn)
    out UCSRnC, r18
    ; Включаємо приймач і передавач
    ldi x18, (1<<RXENn) | (1<<TXENn)
    out UCSRnB, r18; Встановлюємо швидкість передачі
    out UBRRnH, r17
    out UBRRnL, r16
    ret
```

Приклад на C

```
void USART_MSPIM_Init (unsigned int baud)
```

```

{
    UBRRn = 0;
    /* Перемикаємо XCKn на вихід */
    XCKn_DDR |= (1<<XCKn);
    /*Перемикаємося в режим MSPI (0-й режим SPI) */
    UCSRnC = (1<<UMSELn1) |
    (1<<UMSELn0) | (0<<UCPHAn) | (0<<UCPOLn)
    /* Включаємо приймач і передавач */
    UCSRnB = (1<<RXENn) | (1<<TXENn)
    /* Встановлюємо швидкість передачі */
    UBRRn = baud;
}

```

2.4.4 Передача даних в режимі MSPI

Для передачі даних в режимі MSPI необхідно дозволити роботу передавача USART, тобто повинен бути встановлений розряд TXEn регістра UCSRnB. Якщо передбачається отримання даних від ведених пристроїв, то слід включити і приймач, встановивши прапорець RXEn регістра UCSRnB. Процес обміну ініціюється записом байта в регістр даних UDRn. Причому це стосується як прийому, так і передачі даних, оскільки тактовий сигнал формується передавачем. Дані, записані в регістр UDRn, перезавантажуються з буфера передавача у регістр зсуву при його готовності до пересилання наступного кадру. Зверніть увагу на те, що для кожного переданого байта регістр даних UDRn повинен бути прочитаний тільки один раз. Вхідний буфер USART у режимі MSPI функціонує так само, як і в звичайному режимі, тобто при переповненні втрачається останній прийнятий байт, а не перший.

Нижче наведено простий приклад підпрограми передачі даних в режимі MSPI, що використовує опитування прапорців переривань UDREn і

RXCn. Ця підпрограма чекає на очищення буфера передавача, завантажує в нього нове значення, після чого очікує заповнення буфера приймача.

Приклад на асемблері

```
USART_MSPIM_Transfer:
    ; Чекаємо очищення буфера передавача
    sbis UCSRnA, UDREn
    rjmp USART_MSPIM_Transfer
    ;Завантажуємо дані в буфер, ініціюючи передачу
    out UDRn, r16
    ; Чекаємо прийому даних
USART_MSPIM_wait_RX:
    sbis UCSRnA, RXCn
    rjmp USART_MSPIM_Wait_RX
    ; Прочитуємо прийняте значення з буфера
    in r16,UDRn
    ret
```

Приклад на С

```
unsigned char USART_MSPIM_Transfer (unsigned char data)
{
    /* Чекаємо очищення буфера передавача */
    while (!(UCSRnA & (1<<UDREn)))
        ;
    /*Завантажуємо дані в буфер, ініціюючи передачу */
    UDRn = data;
    /* Чекаємо прийому даних */
    While (!(UCSRnA & (1<<RXCn)))
        ;
    /* Прочитуємо прийняте значення з буфера і повертаємо
    його */
    return UDRn;
}
```

2.5 Формування синхрочастот для передавача та приймача послідовного порту

2.5.1 Мікроконтролери сімейства MCS-51

Для пояснення формування синхрочастот передавача та приймача послідовного порту, який працює в асинхронному режимі, скористаємося інформацією, наведеною у лекції 7 першого кредитного модуля цієї дисципліни - «Проектування мікропроцесорних систем на базі мікроконтролерів сімейства MCS-51» [22].

У режимі 1 прийом/передача даних здійснюється у форматі восьмирозрядного УАПП. Через TxD передаються, а через RxD приймаються 10 біт: старт-біт (лог. 0), 8 біт даних і стоп-біт (лог. 1). Під час прийому стоп-біт заноситься у біт RB8 регістра SCON. Швидкість (частота) прийому/передачі визначається частотою переповнень таймера/лічильника1: $F_{OV\ T/C1}$. Ця частота всередині УАПП формується за схемою, наведеною на рисунку 2.14.

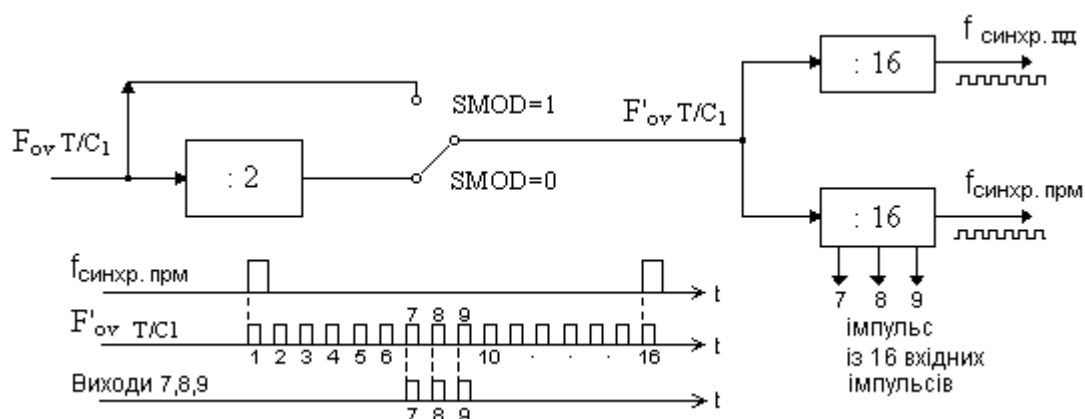


Рисунок 2.14 – Схема формування синхрочастот передачі та прийому всередині МК-51 для послідовного порту, що працює в режимах 1, 3

В залежності від значення біта SMOD регістра PCON частота, що надходить на вхід дільників на 16, $F'_{OV\ T/C1} = F_{OV\ T/C1}$, при SMOD = 1 і $F'_{OV\ T/C1} = \frac{F_{OV\ T/C1}}{2}$, при SMOD = 0. На виходах дільників на 16 формуються частоти синхронізації передавача $f_{\text{синхр.пд}}$ і приймача $f_{\text{синхр.прм.}}$. На виходах 7, 8, 9 дільника на 16, який формує $f_{\text{синхр.прм.}}$, виробляються 3 коротких імпульси, коли на вхід лічильника (дільника) надходять відповідно 7-, 8- і 9-й імпульси частоти $F'_{OV\ T/C1}$, починаючи з початку циклу лічби. Ці імпульси використовуються логічною схемою ідентифікації значення чергової прийнятої посилки (0/1) за мажоритарним принципом.

Передача (рисунок 2.15) ініціюється будь-якою командою, що використовує SBUF в якості регістра призначення, у який виконується запис. Внутрішній імпульс мікроконтролера ЗАПИС у SBUF, що виробляється при цьому, завантажує призначений до передачі байт у молодші 8 розрядів регістра зсуву передавача та ініціює початок роботи блоку керування передачею.

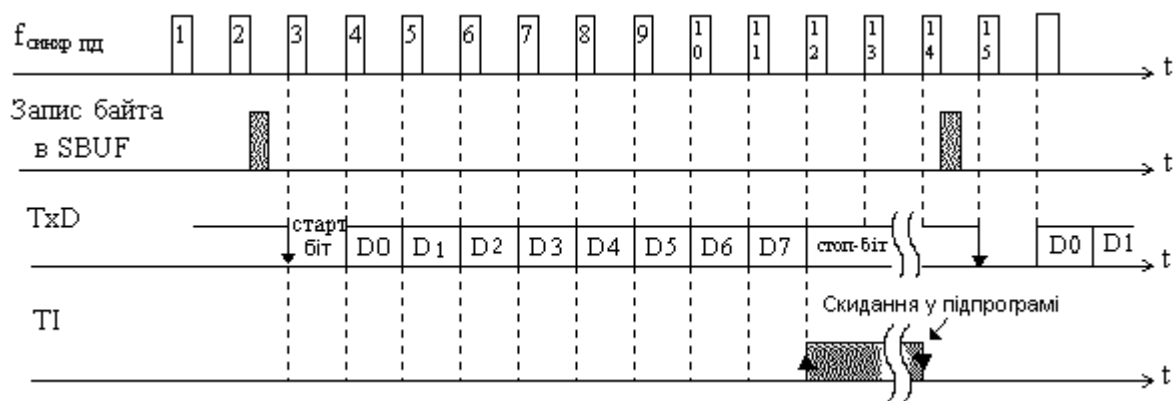


Рисунок 2.15 – Передача в режимі 1

В режимі 1 регістр зсуву передавача має 9 розрядів і в його 9-й розряд за імпульсом ЗАПИС у SBUF заноситься "1" (стоп-біт).

Реально передача починається у фазі S1 P1 машинного циклу, яка слідує за найближчим після ЗАПИС у SBUF переповненням дільника на 16 у ланцюзі формування сигналу $f_{\text{синхр. пд}}$ (рисунок 2.14). Таким чином, початок передачі синхронізований дільником на 16, а не імпульсом ЗАПИС у SBUF. Період сигналу $f_{\text{синхр. пд}}$ (синхронізація передачі) визначає час, протягом якого біт, який видається, присутній на виході TxD (час передачі біта).

Передача починається встановленням активного рівня внутрішнього сигналу мікроконтролера ПОСИЛКА, поява якого викликає видачу на вихід TxD рівня старт-біта (нуль). Після цього через час передачі одного біта стає активним внутрішній сигнал мікроконтролера ДАНІ, який дозволяє видачу вмісту регістра зсуву передавача на вихід TxD (вивід P3.0 мікроконтролера). При появі активного сигналу ДАНІ старт-біт на виході TxD замінюється бітом D0 регістра зсуву передавача. По закінченні часу передачі біта D0 формується перший внутрішній імпульс мікроконтролера ЗСУВ, за яким вміст регістра зсуву передавача зсувається на один розряд, і біт D0 на виході TxD замінюється бітом D1. Усього формується 9 імпульсів ЗСУВ, у результаті чого на вихід TxD видаються 8 біт даних і стоп-біт. По закінченні видачі всіх біт посилки блок керування передачею встановлює прапорець переривання передавача ТІ і знімає сигнали ПОСИЛКА і ДАНІ. За перериванням відбувається скидання прапорця ТІ, запис нового байта тощо.

Прийом (рисунок 2.16) починається при виявленні переходу сигналу на вході RxD з "1" в "0". Для того, щоб виявити такий перехід, вхід RxD апаратно опитується з частотою $F'_{\text{OV T/C1}}$ (рисунок 2.14), тобто 16 разів на одну посилку.

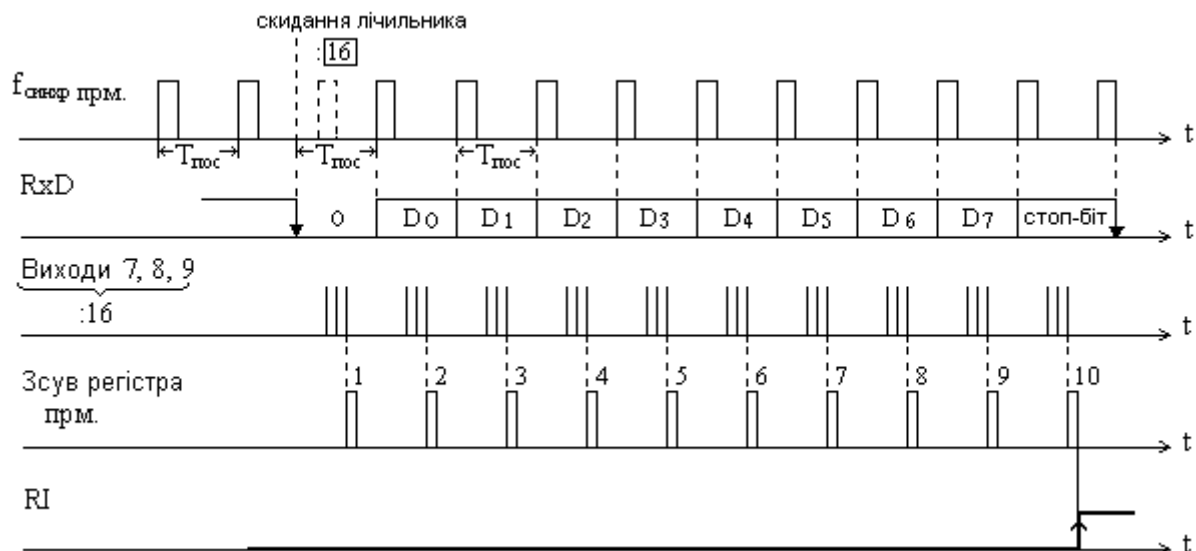


Рисунок 2.16 – Прийом у режимі 1

Коли перехід сигналу на вході RxD із "1" в "0" виявлений, негайно скидається лічильник (дільник на 16) у ланцюзі формування сигналу $f_{\text{синхр. прм}}$ (рисунок 2.14), у результаті чого відбувається поєднання моментів переповнення цього лічильника (дільника) (імпульси $f_{\text{синхр. прм}}$ на рисунку 16) із границями зміни бітів прийнятої посилки на вході RxD.

Шістнадцять станів лічильника (дільника) поділяють час, протягом якого кожний біт прийнятої посилки присутній на вході RxD, на 16 фаз, з 1-ї по 16-у для кожного біта. У фазах 7, 8 і 9 спеціальний пристрій мікроконтролера – біт-детектор, зчитує із входу RxD три значення прийнятого біта, за мажоритарним принципом "2 або 3 із 3-х" вибирає одне з них і подає його на вхід регістра зсуву приймача. Блок керування прийомом при цьому формує внутрішній імпульс мікроконтролера ЗСУВ, у результаті чого вміст регістра зсуву приймача зсувається на один розряд і прийнятий біт заноситься у регістр зсуву приймача. Усього формується 10 імпульсів ЗСУВ, а регістр зсуву приймача у режимі 1 є 9-розрядним. Тому після 10-го імпульсу ЗСУВ у регістрі зсуву приймача знаходяться біти даних D0-D7 і стоп-біт. Після 10-го імпульсу ЗСУВ блок керування прийомом завантажує дані з регістра зсуву приймача у SBUF, завантажує

стоп-біт з регістра зсуву приймача у розряд RB8 регістра SCON і встановлює прапорець переривання приймача RI. Сигнал завантаження SBUF, RB8 і встановлення RI виробляється блоком керування прийомом тільки в тому випадку, якщо в момент генерації останнього імпульсу ЗСУВ виконуються наступні умови:

- $RI = 0$ і
- або $SM2 = 0$, або $(SM2 = 1)$ прийнятий стоп-біт дорівнює "1".

Якщо хоча б одна з цих умов не виконується, прийнята посилка безповоротно втрачається, а прапорець RI не встановлюється. Якщо обидві наведені умови виконані, стоп-біт надходить у RB8, вісім біт даних надходять у SBUF і встановлюється прапорець RI. У цей час, незалежно від виконання наведених вище умов, послідовний порт знову починає відслідковувати наявність переходу сигналу з "1" а "0" на вході RxD і прийом нового байта. До закінчення цього процесу попередній байт повинний бути прочитаний із буфера ПРМ, інакше буде накладення нового прийнятого байта на старий.

Якщо мажоритарний відбір при прийомі першого біта посилки (старт-біт) показує ненульове значення біта, всі пристрої блоку прийому скидаються, і починається відслідковування наступного переходу сигналу з "1" в "0" на вході RxD. Таким чином, забезпечується захист від помилкових старт-бітів.

2.5.2 Розрахунок швидкості передачі інформації та тривалості одного біта для модуля USART AVR-мікроконтролерів

Дуже часто зв'язок МК-ра з міжконтролерною мережею здійснюється за допомогою послідовного інтерфейсу УСАПП (USART), який може, наприклад, програмуватися в асинхронний звичайний (біт $U2X_n = 0$) режим [1, 2].

При цьому швидкість передачі даних (обміну) для асинхронного звичайного режиму:

$$f_{\text{ПД}} = \frac{f_{\text{CLK}}}{16 \cdot (\text{UBRR} + 1)},$$

де f_{CLK} – частота системного тактового генератора (при підключенні зовнішнього кварцу визначається частотою, на яку виготовляється кварц);

UBRR – вміст регістра контролера швидкості передачі, що змінюється програмно (UBRR = 0...4095).

Прийmemo $f_{\text{CLK}} = 16$ МГц, UBRR = 24, тоді:

$$V_{\text{ПД}} = \frac{16000000}{16 \cdot 25} = 40000 \text{ посилок / секунду} = 40000 \text{ біт / секунду} = 40 \text{ Кбіт / с.}$$

Тоді тривалість однієї посилки:

$$t_{\text{ПОС}} = \frac{1}{V_{\text{ПД}}} = \frac{1000000}{40000} = 25 \text{ мксек.}$$

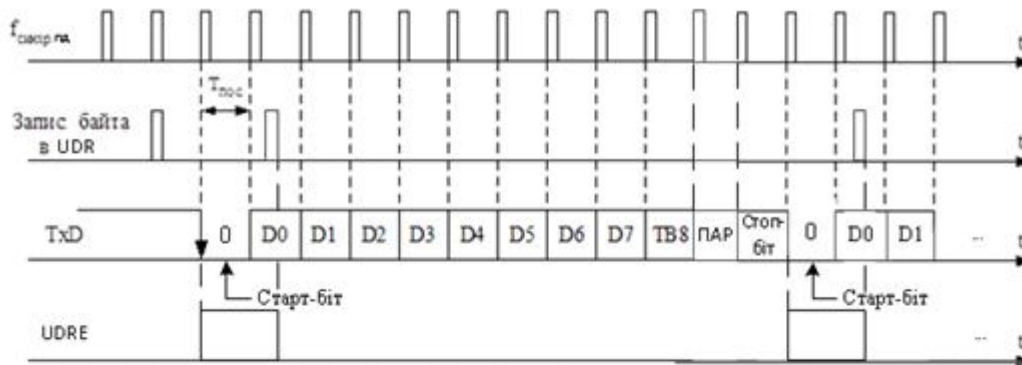
Часові діаграми роботи цього модуля для асинхронного режиму наведені на рисунку 2.17.

У лінію зв'язку передаються наступні біти: старт-біт, 8 інформаційних біт, 9-й службовий біт TB8, додатковий перевірючий біт на парність і один стоп-біт.

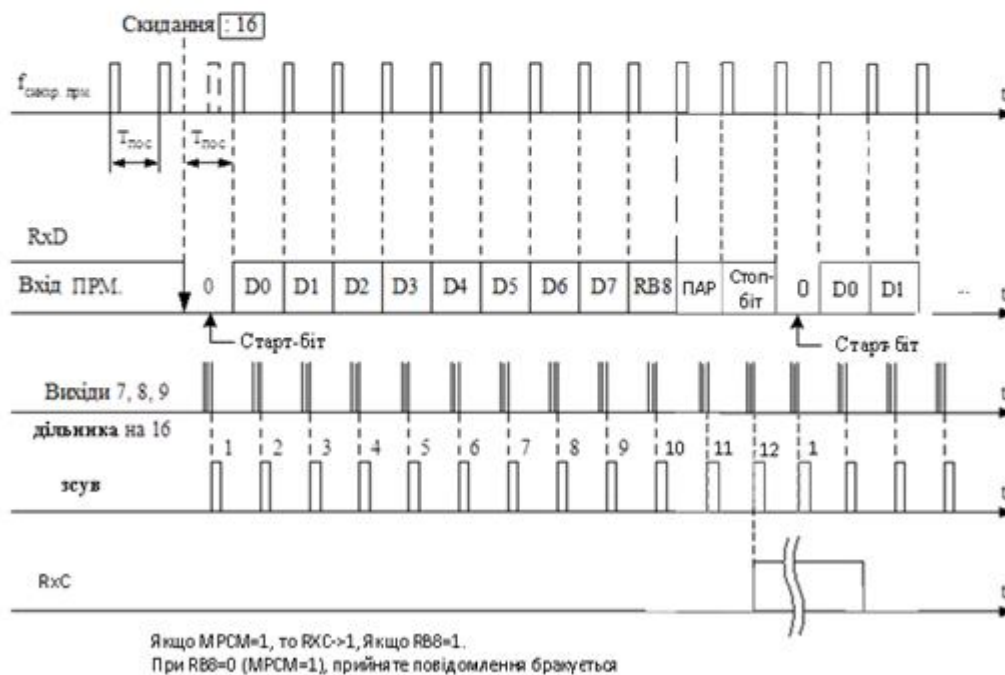
Тривалість цих 12 посилок однакова і дорівнює $t_{\text{ПОС}} = 25$ мксек, при $V_{\text{ПД}} = 40$ Кбіт/с.

Оскільки передавач містить буферний регістр даних UDR, в який програмно завантажується черговий байт для передачі, поки з регістра зсуву видаються посилки попереднього байта, то після передачі стоп-біта новий байт з регістра UDR пересилається в регістр зсуву та ініціюється новий цикл передачі.

Таким чином, час передачі одного байта дорівнює $12 \cdot t_{\text{посилок}}$, що при тривалості $t_{\text{пос}} = 25$ мксек становить $t_{\text{б}} = 25 \cdot 12 = 300$ мксек.



а



б

Рисунок 2.17 – Часові діаграми роботи інтерфейсу:

а – передача даних; б – прийом даних

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Що таке USART та UART? В чому полягає різниця між ними?
- 2) З яких трьох частин складається модуль USART/UART? Які елементи містить кожен з них?
- 3) Які регістри використовуються для керування модулями UART та USART?
- 4) Чим в асинхронному режимі задається швидкість прийому та передачі даних?
- 5) Поясніть структурну схему блока синхронізації модуля УСАПП для мікроконтролера Mega 128.
- 6) Які режими роботи синхронізації підтримує УСАПП?
- 7) За якими формулами визначається швидкість обміну?
- 8) Як визначається розмір слова даних у модулях USART?
- 9) Як працює схема контролю парності?
- 10) Як проходить передача даних в модулях USART?
- 11) Як проходить прийом даних в модулях USART?
- 12) Що таке режим мультипроцесорного обміну?
- 13) Опишіть послідовність дій для здійснення обміну даними в багатопроцесорному режимі.
- 14) Як визначити швидкість передачі даних для асинхронного звичайного режиму?

ЛІТЕРАТУРА [1...10]

3 ПОСЛІДОВНИЙ СИНХРОННИЙ ІНТЕРФЕЙС TWI (I²C)

3.1 Характеристика інтерфейсу TWI (I²C) на апаратному рівні

3.1.1 Загальна характеристика інтерфейсу I²C

Інтерфейс I²C (InterIC, або IC) є двопровідним послідовним синхронним інтерфейсом, розробленим фірмою “Philips Corporation”, і призначений для зв'язку між інтегральними мікросхемами або модулями. Існує ціла група I²C-сумісних пристроїв для різних додатків: цифро-аналогові та аналого-цифрові перетворювачі, мікросхеми пам'яті, мікроконтролери, що містять модуль I²C і т.ін [1...7].

Шина інтерфейсу I²C 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) I²C.

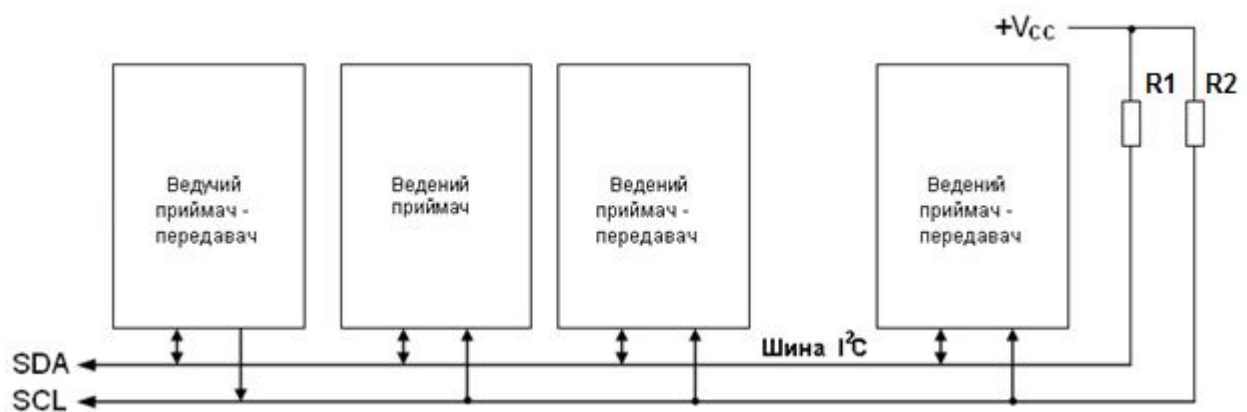


Рисунок 3.1 – Структурна схема мережі з інтерфейсом I²C

Лінії SDA і SCL шини з'єднані з додатним полюсом напруги живлення (+V_{cc}) через підтягуючі резистори, R1 та R2.

Передавач генерує і передає повідомлення, а приймач його приймає.

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

Кожний пристрій, який використовує для обміну інтерфейс I²C, має свою адресу. Коли ведучий пристрій бажає ініціювати обмін даними, він передає на лінію SDA адресу пристрою, з яким буде виконуватися обмін (передача/прийом). Всі ведені пристрої слідкують за адресою, яка виставляється на шину, і порівнюють її із власною адресою. Після адреси ведучий передає біт напрямку $R/\overline{W}$, який визначає, чи буде ведучий читати дані від веденого ($R/\overline{W} = 1$), чи буде передавати дані веденому ($R/\overline{W} = 0$). Ведений приймач після одержання адреси або даних видає на шину SDA біт підтвердження (логічний нуль). Інтерфейс I²C може використовувати два формати адреси:

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

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

На рисунку 3.2 наведений формат 7-ми бітної адреси.

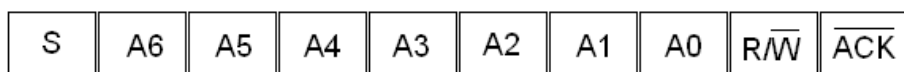
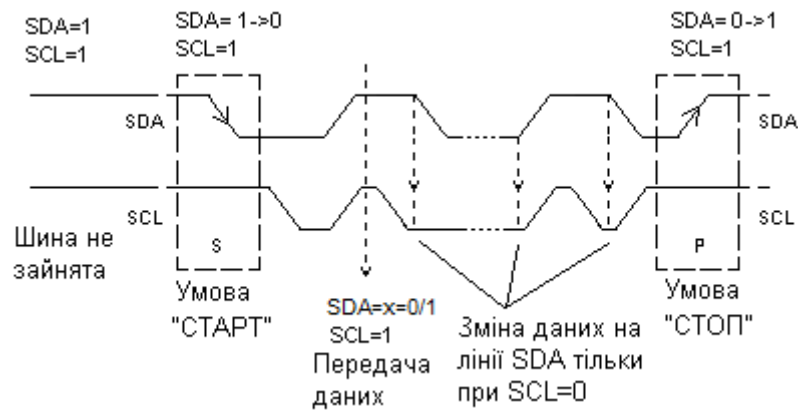


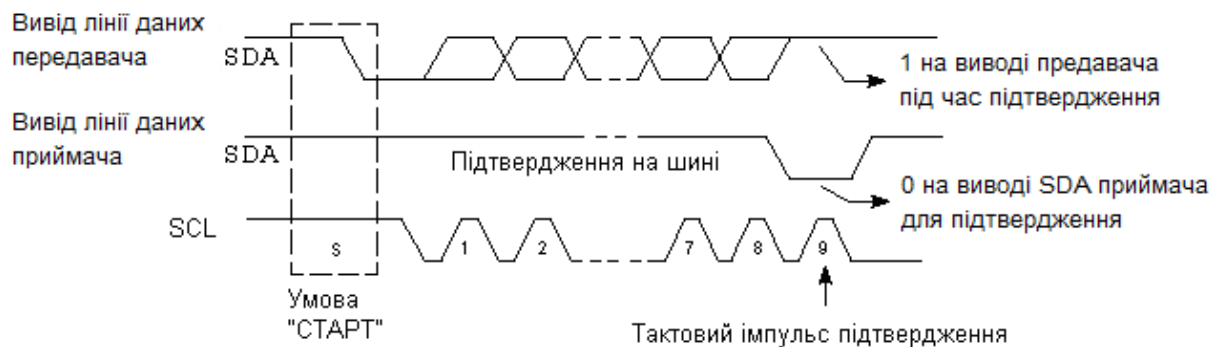
Рисунок 3.2 – Формат 7-бітної адреси

(S - старт, $R/\overline{W}$ - біт «читання / 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 I²C.

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

3) Ведений приймач: ведучий передавач видає на лінію SDA адресу веденого приймача й біт напрямку $R/\overline{W}=0$, що говорить про те, що буде виконуватися запис (передача). На лінії SCL ведучий передавач видає синхроімпульси. Після одержання адреси ведений приймач передає сигнал підтвердження $\overline{ACK}$, після чого ведучий передавач послідовно передає на лінію SDA дані. Ведений приймач після прийому чергового байта даних передає сигнал підтвердження $\overline{ACK}$, що надходить до ведучого передавача лінією SDA. Умови СТАРТ і СТОП формуються ведучим передавачем.

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

підтвердження $\overline{ACK}$, який передає ведучий приймач. Умови СТАРТ і СТОП формує ведучий приймач.

У підпорядкованому (веденому) режимі апаратні засоби інтерфейсу I²C здійснюють пошук своєї власної підпорядкованої адреси або адреси загального виклику. Якщо детектується одна із цих адрес, відбувається запит на переривання і виконуються відповідні дії. Якщо модуль I²C 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 I²C можуть спробувати стати ведучими і одночасно згенерувати умову «СТАРТ». У цьому випадку здійснюється арбітраж шини в моменти, коли шина SCL перебуває у високому стані. Якщо один ведучий передає на лінію даних низький рівень, а інший – високий, то останній відключається від лінії, тому що стан шини SDA (низький) не відповідає високому стану внутрішньої шини даних пристрою, який бажає стати ведучим.

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

На рисунку 3.4 наведені часові діаграми, які відображають обмін даними шиною I²C.

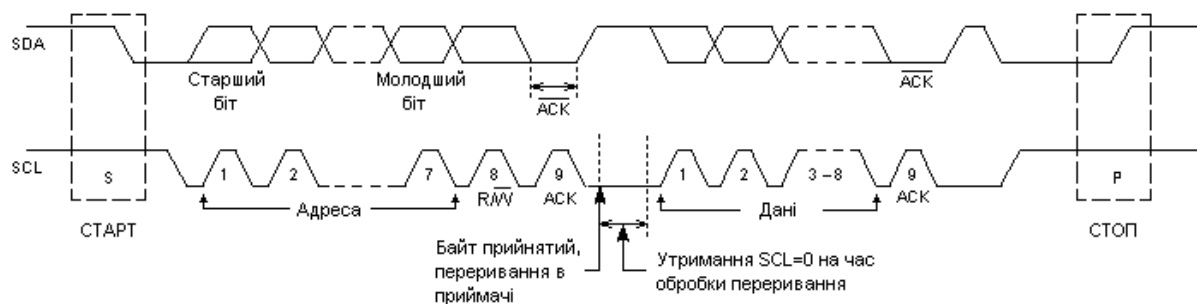


Рисунок 3.4 – Приклад обміну даними шиною I²C

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 ATmega8515х, 162х. В моделях 165х, 325х, 3250х, 645х, 6450х функцію TWI може виконувати модуль USI, який розглядається в розділі CPC до лекції 12 [2].

Інтерфейс TWI являється повним аналогом базової версії інтерфейсу I²C 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 кожен ліній (рисунок 3.5, резистори R1, R2).

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 ліній сигнал лог. 0, а ВИСОКИЙ рівень на лінії встановлюється 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.

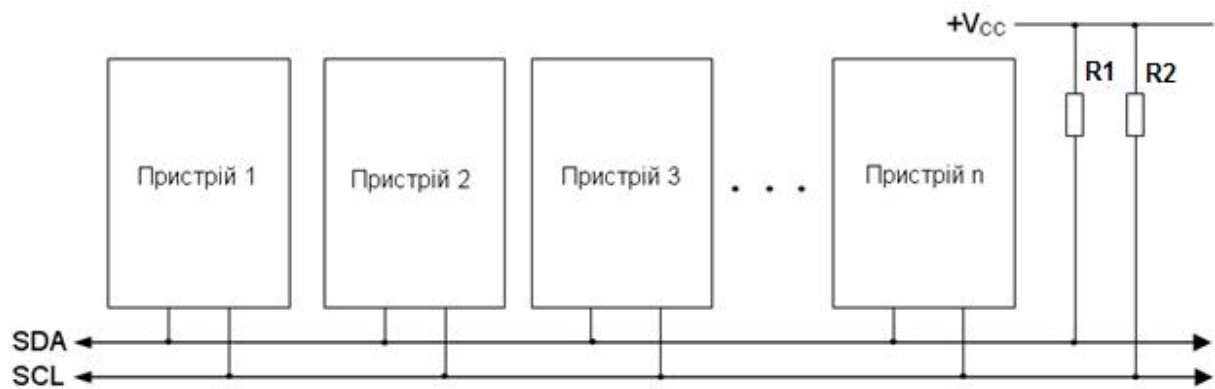


Рисунок 3.5 – З'єднання пристроїв за допомогою шини TWI

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.

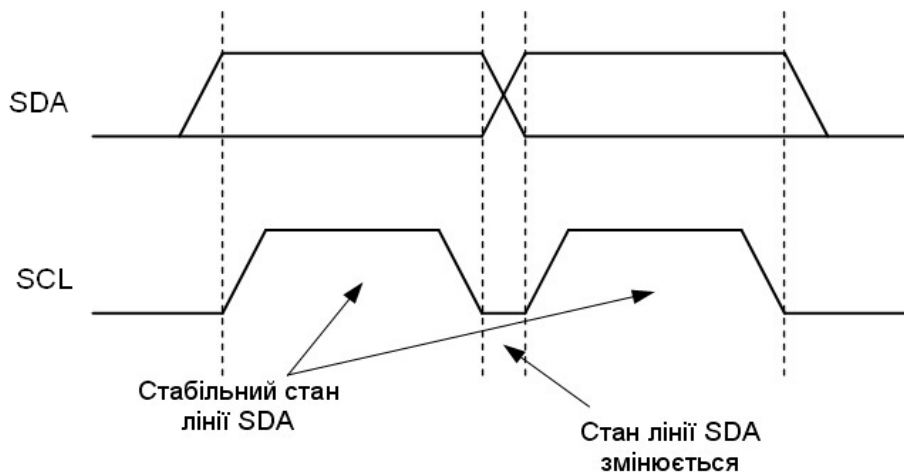


Рисунок 3.6 – Коректна видача даних на шину SDA

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.

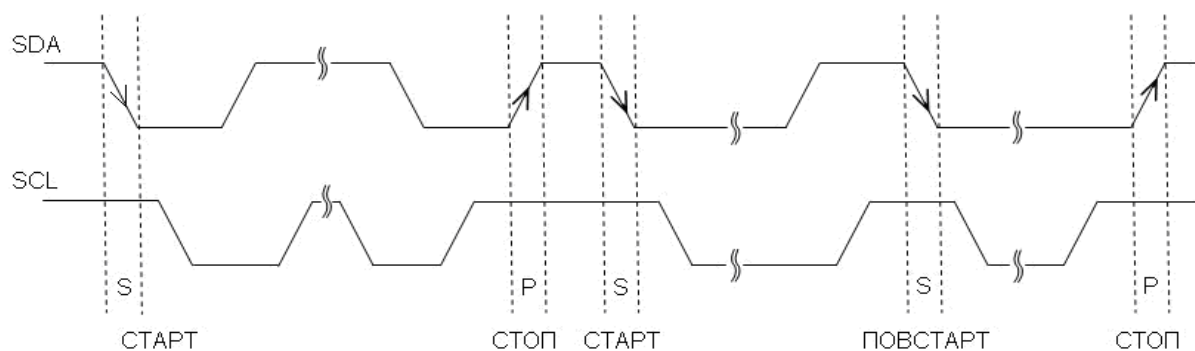


Рисунок 3.7 – Стани СТАРТ, СТОП і ПОВСТАРТ

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).

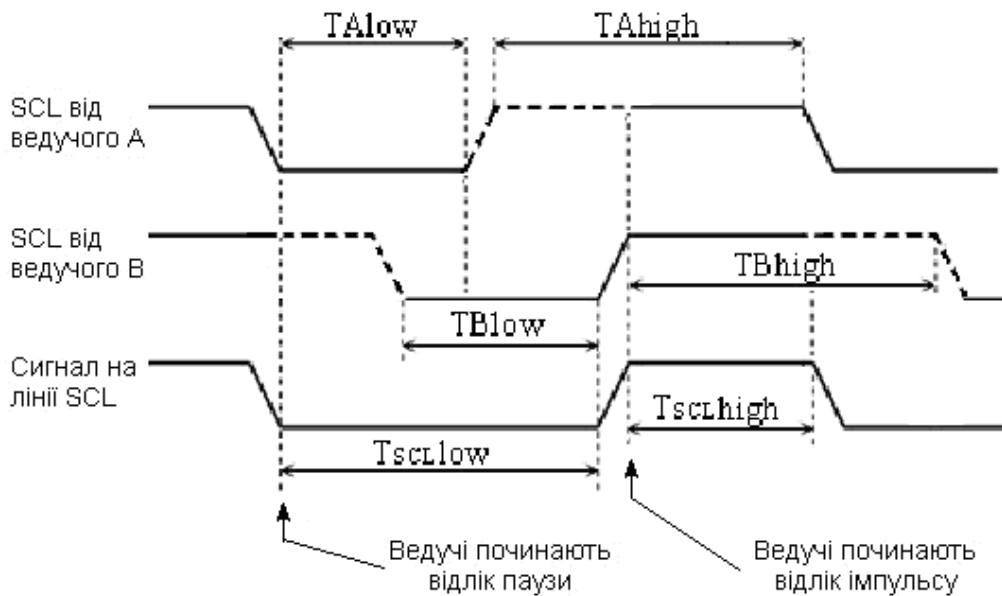


Рисунок 3.8 – Синхронізація сигналу SCL декількох ведучих

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.

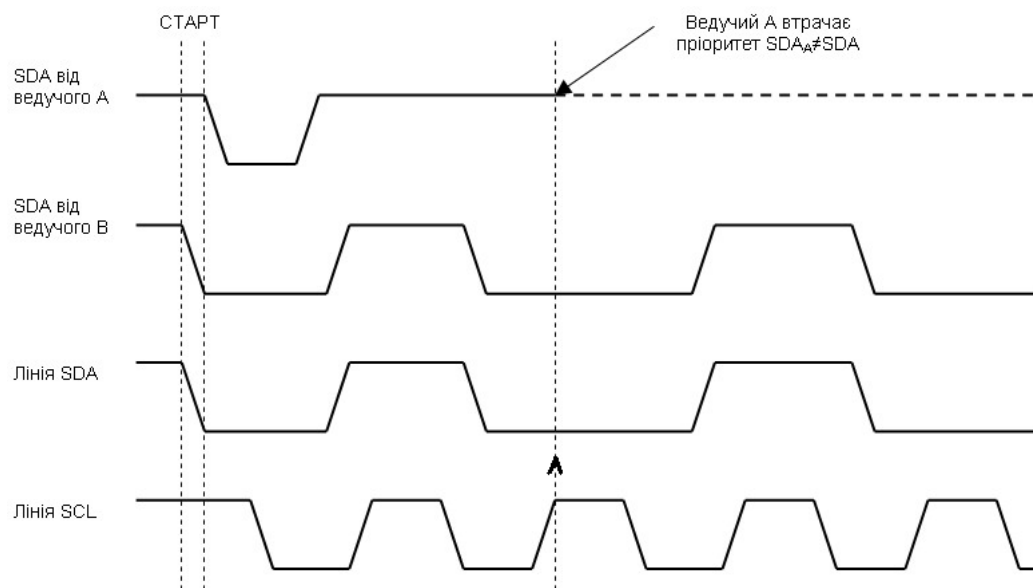


Рисунок 3.9 – Розподіл пріоритетів між двома ведучими

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 впливає, 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).

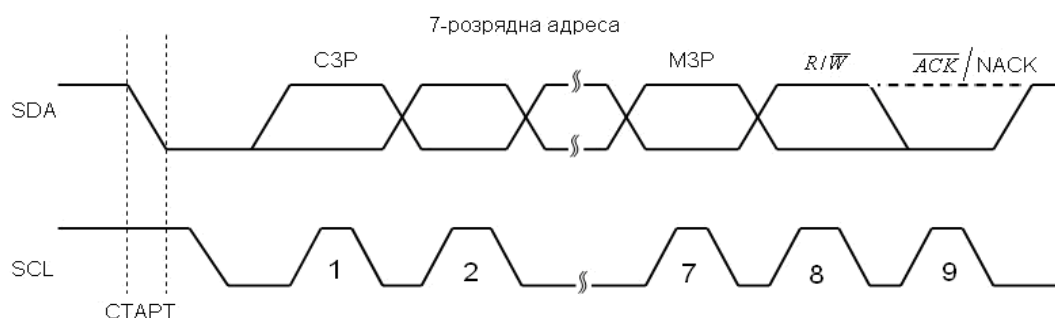


Рисунок 3.10 – Формат адресного пакета

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).

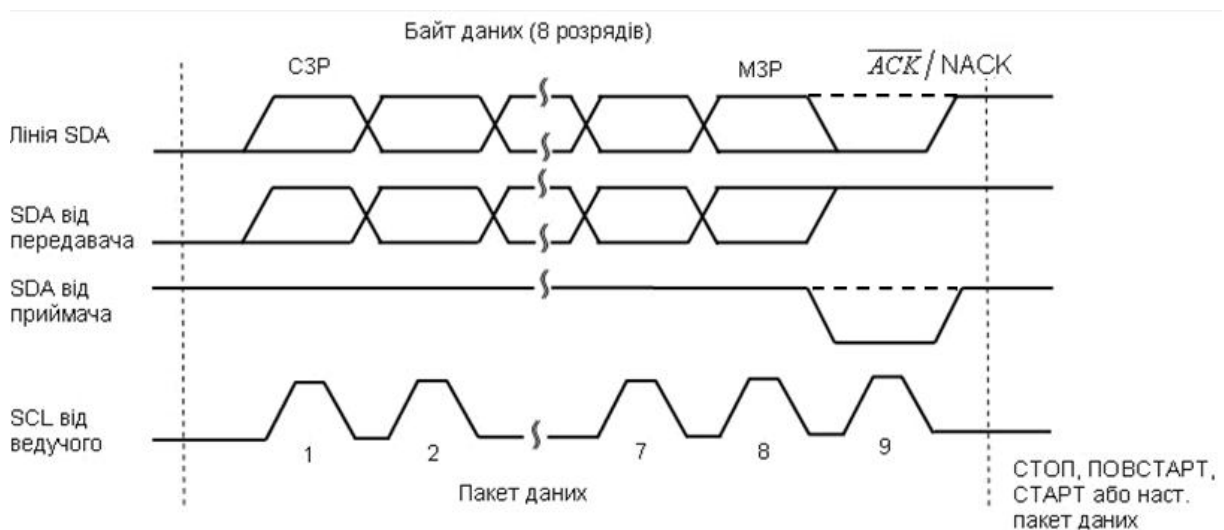


Рисунок 3.11 – Формат пакета даних

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.

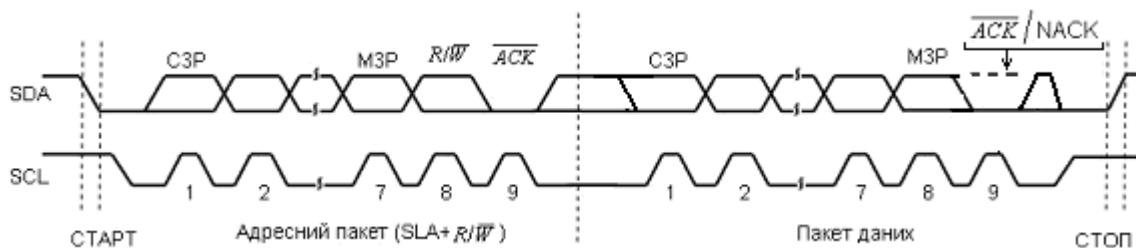


Рисунок 3.12 – Типовий цикл обміну

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.

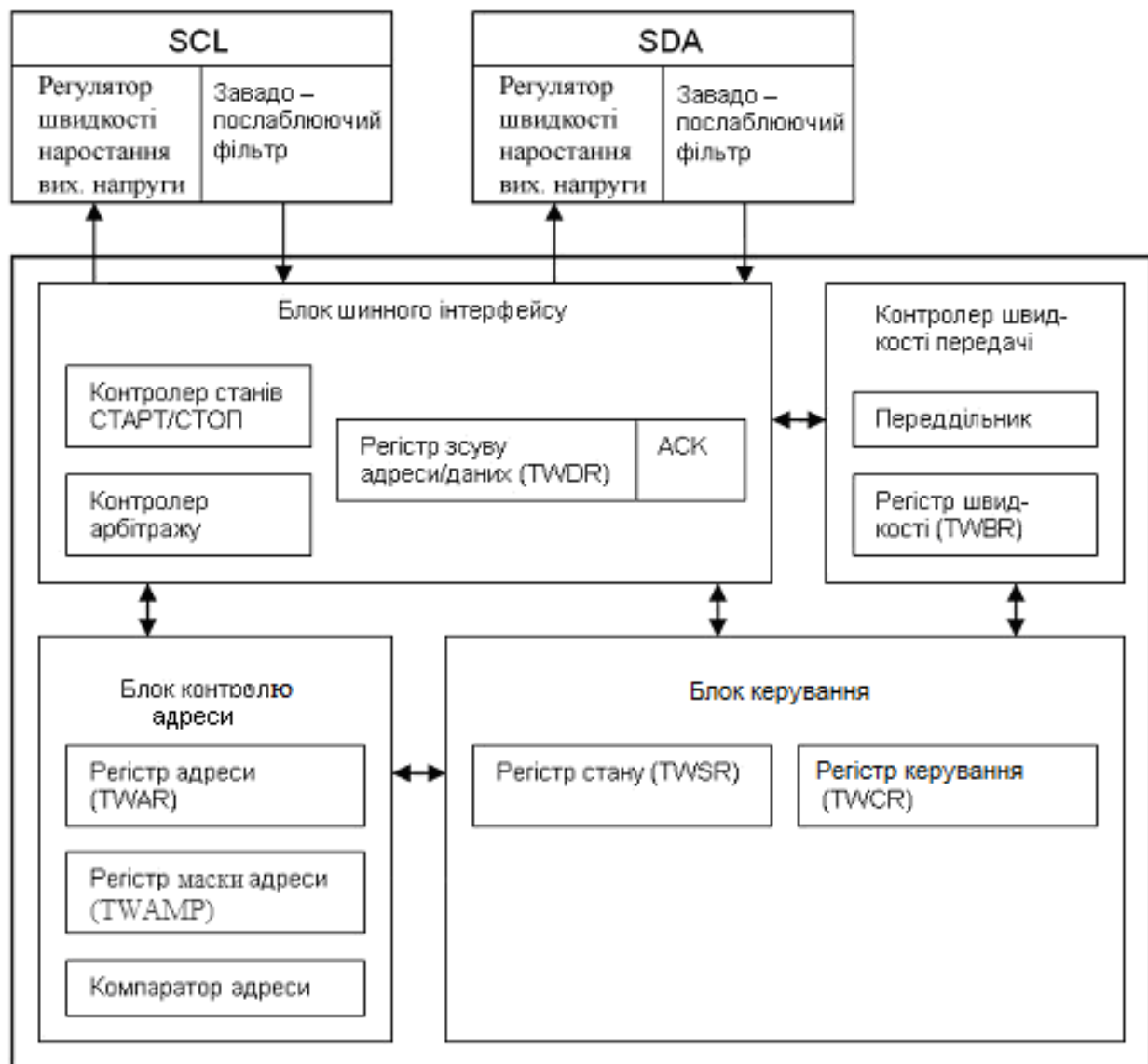


Рисунок 3.13 – Структурна схема модуля TWI

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]:

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

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:TWPS0 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 TWA_R. 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 TWA_R 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($\bar{W}$)	$R/\bar{W}$	$R/\bar{W}$	$R/\bar{W}$	$R/\bar{W}$	$R/\bar{W}$	$R/\bar{W}$	$R/\bar{W}$	$R/\bar{W}$
Початкове значення	1	1	1	1	1	1	1	0

Рисунок 3.14 – Формат регістра TWA_R

Таблиця 3.4 – Розряди регістра адреси TWA_R

Розряд	Назва	Опис
7...1	TWA6...TWA0	<u>Адреса пристрою</u> 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	<u>Дозвіл розпізнавання загальних викликів</u> 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

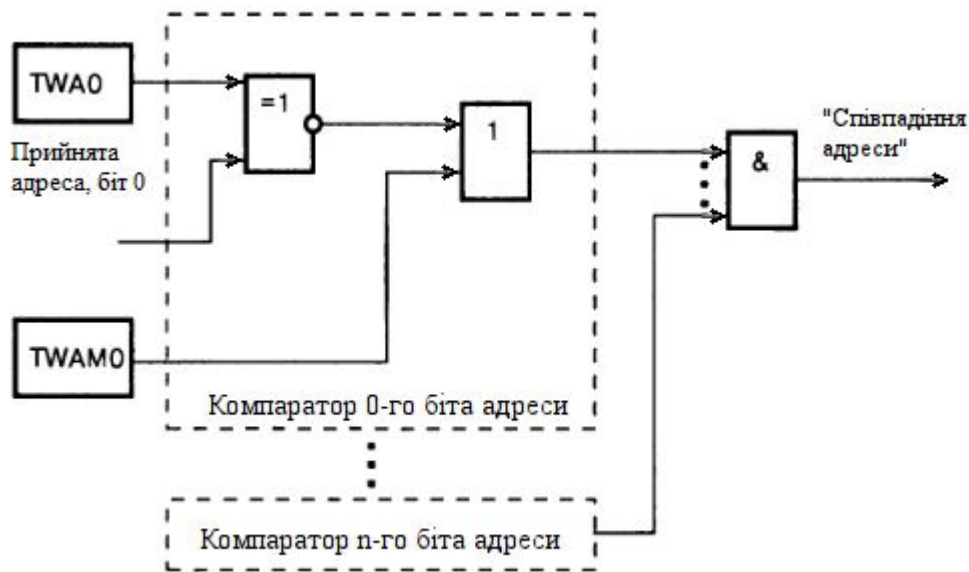


Рисунок 3.16 – Структурна схема блоку контролю адреси

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 ($SLA + R/\bar{W}$);
- 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($\bar{W}$)	$R/\bar{W}$	$R/\bar{W}$	$R/\bar{W}$	$R/\bar{W}$	R	$R/\bar{W}$	R	$R/\bar{W}$
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	<u>Прапорець переривання від модуля TWI</u> 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	<u>Дозвіл біта підтвердження</u> 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	<u>Прапорець стану СТАРТ</u> 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	<u>Прапорець стану СТОП</u> 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 повертається в режим неадресованого веденого, а виводи SCL і SDA для передачі переводяться у третій стан. Стан СТОП не формується
3	TWWC	<u>Прапорець конфлікту запису</u> 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	<u>Дозвіл роботи модуля TWI</u> 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	<u>Дозвіл переривання від модуля TWI</u> 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.

	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	—	—	—	ATmega163x ATmega323x
Читання(R)/Запис($\overline{W}$)	R	R	R	R	R	R	R	R	
Початкове значення	1	1	1	1	1	0	0	0	

	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	—	TWPS1	TWPS0	Інші моделі
Читання(R)/Запис($\overline{W}$)	R	R	R	R	R	R	R	R	
Початкове значення	1	1	1	1	1	0	0	0	

Рисунок 3.18 – Формат регістра TWSR

Таблиця 3.6 – Розряди регістра стану TWSR

Розряд	Назва	Опис
7...3	TWS7...TWS3	<u>Стан модуля TWI</u> 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	<u>Коефіцієнт ділення попереднього дільника контролера швидкості передачі</u> 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.

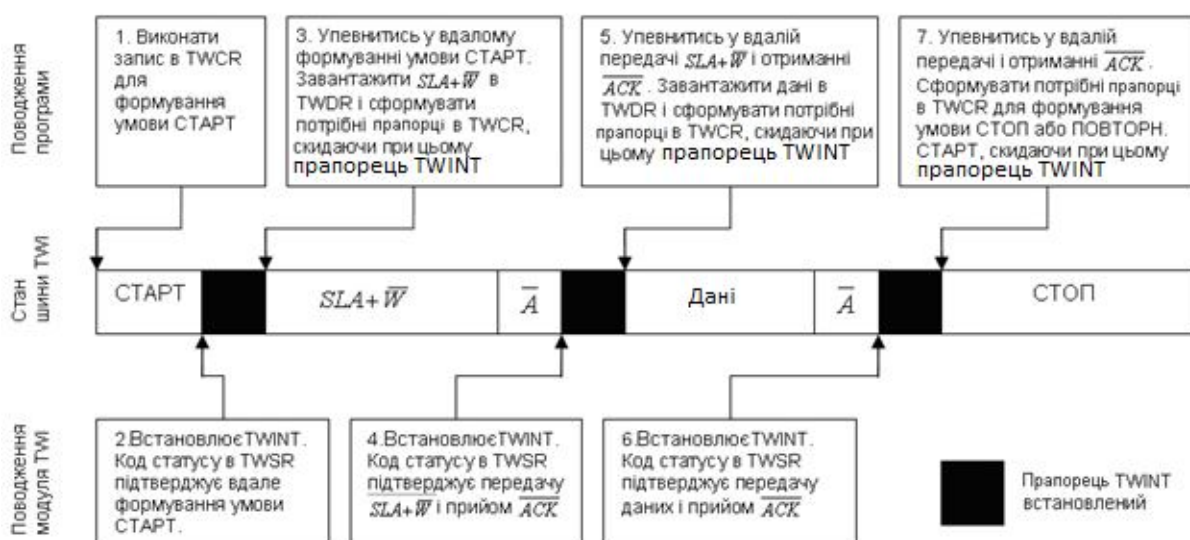


Рисунок 3.19 – Приклад взаємодії програми з модулем TWI

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 підтримують інтерфейс TWI (таблиця 3.9).

Таблиця 3.9 – Пристрої, які підтримують інтерфейс TWI

Пристрій	Опис пристрою
CP2120	Виробник: Silabs <i>Інтегральний контролер моста SPI в TWI</i> 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 <i>Постійний запам'ятовуючий пристрій, який програмується та використовує електричне стирання інформації</i> 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 <i>Синтезатор частоти</i> Єдиний задаючий генератор (4 МГц) для діапазонів АМ та FM Налаштовувана мережа частот 1, 10, 25 КГц Програмний контроль переключення діапазонів
TDA8424 (TDA8421, TDA8425, TDA8426)	Виробник: PhilipsSemiconductors <i>Стереофонічний аудіопроцесор</i> Регулювання тембру по високим і низьким частотам
PCF8574	Виробник: NXP <i>8-ми бітовий перетворювач даних від шини TWI</i> 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 <i>Годинник/Календар з 240*8 біт RAM</i> 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 <i>Цифровий термометр і термостат</i> Час перетворення сигналу: 1 с. Діапазон температур: -55...+125 (крок 0,5 C) або -67...+257 (крок 0,9 F)
PCF8591	Виробник: NXP Semiconductors <i>8-ми бітовий АЦП/ЦАП</i> 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 <i>Мостовий підсилювач потужності (ПНЧ) із вбудованими засобами діагностики стану</i> 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 перемикавання 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».

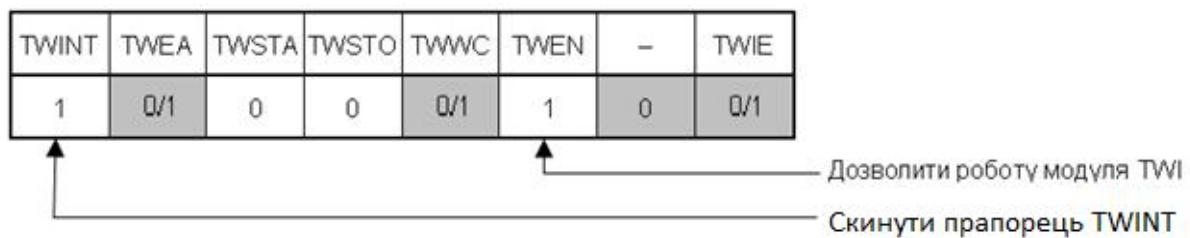
Формування стану СТАРТ почнеться після запису в регістр TWCR наступного значення:

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

Дозволити роботу модуля TWI
Сформувати стан СТАРТ
Скинути прапорець TWINT

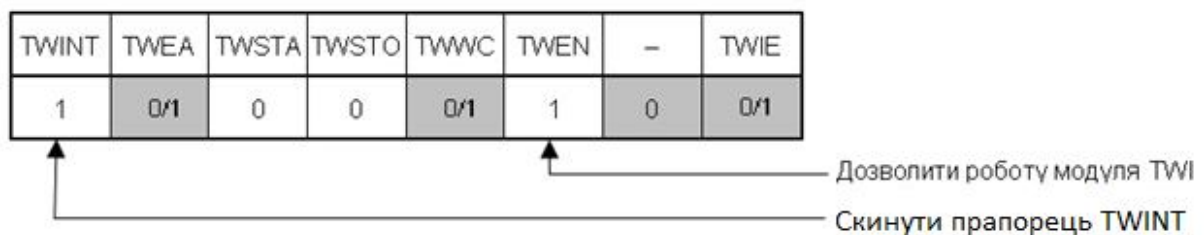
У результаті запису зазначеного значення модуль TWI почне контролювати стан шини і сформує стан СТАРТ відразу ж, як тільки вона стане вільною.

По закінченні формування стану СТАРТ встановлюється прапорець TWINT; код статусу повинен при цьому мати значення \$08 (таблиця 3.10). Для перемикавання 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:

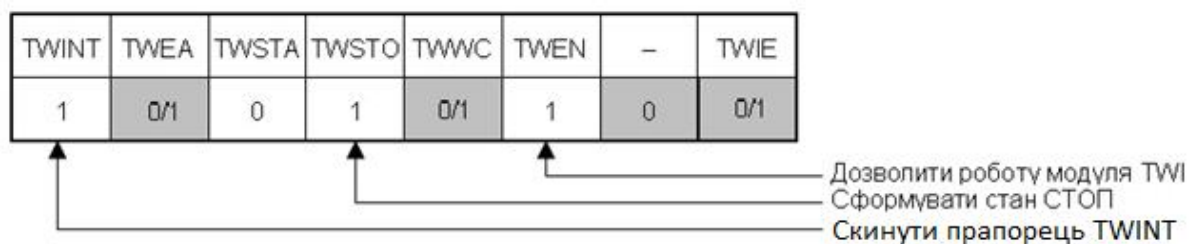


Після передачі адресного пакета і прийому біта підтвердження прапорець TWINT знову встановлюється в «1». Код статусу на цьому етапі може мати одне з наступних значень: \$18, \$20 або \$38. Які дії необхідно виконати при виявленні того або іншого коду, докладно розглянуто в таблиці 3.10.

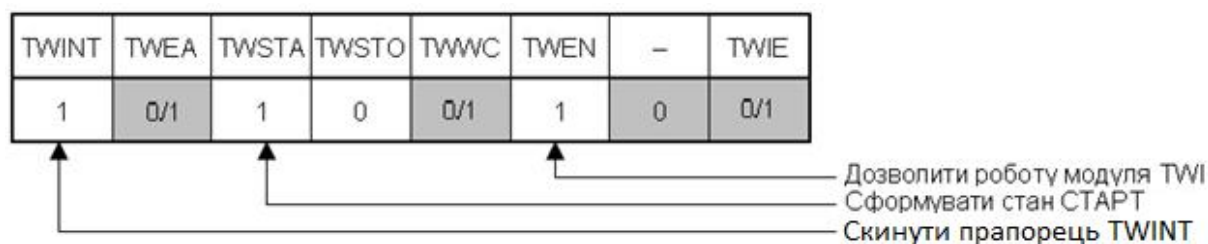
Після передачі адресного пакета повинні бути передані пакети даних. Значення байта даних завантажується в регістр TWDR. Передача пакета даних починається після запису в регістр TWCR наступного значення:



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».

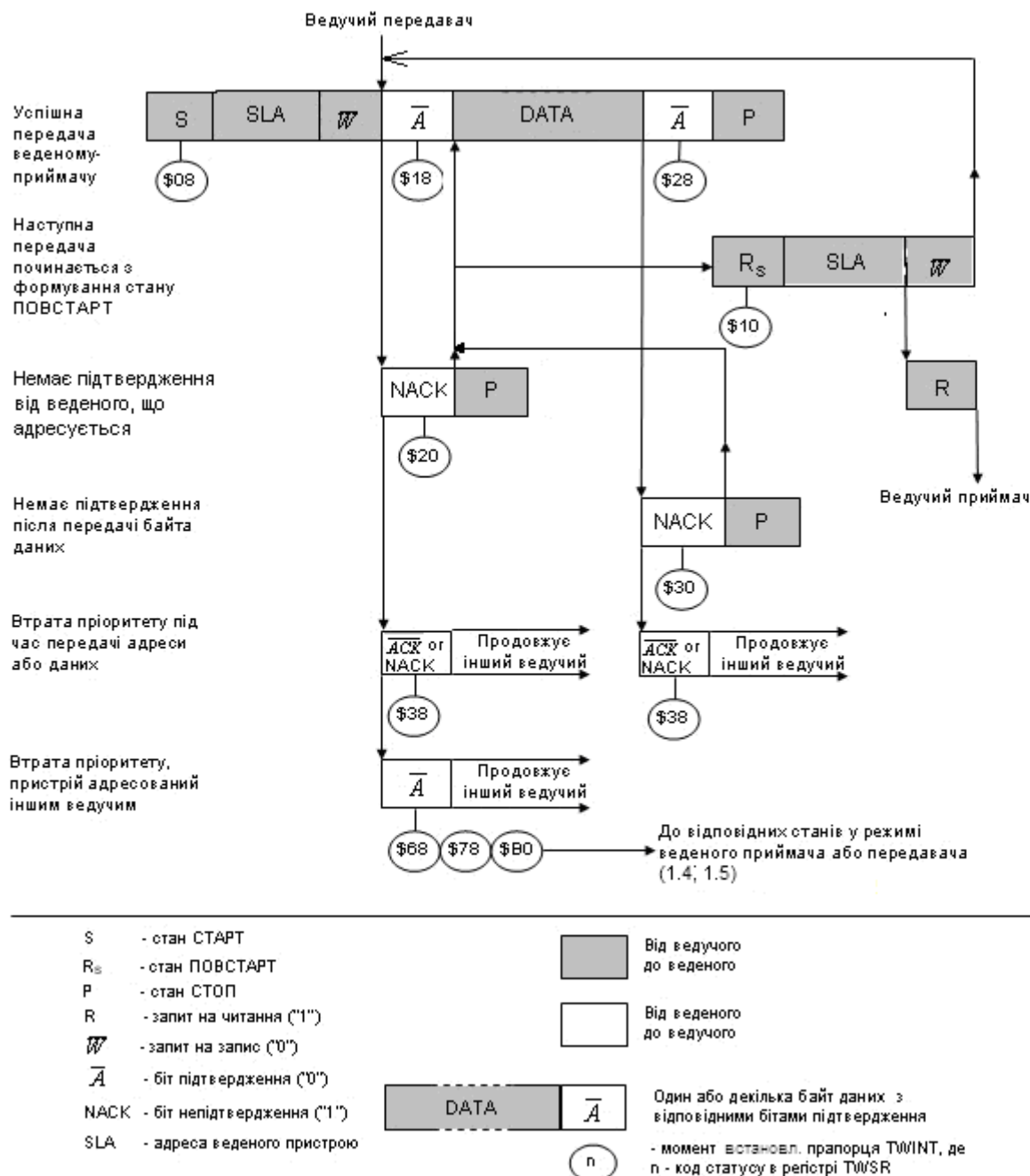


Рисунок 3.20 – Стани модуля TWI у режимі “Ведучий передавач”

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:

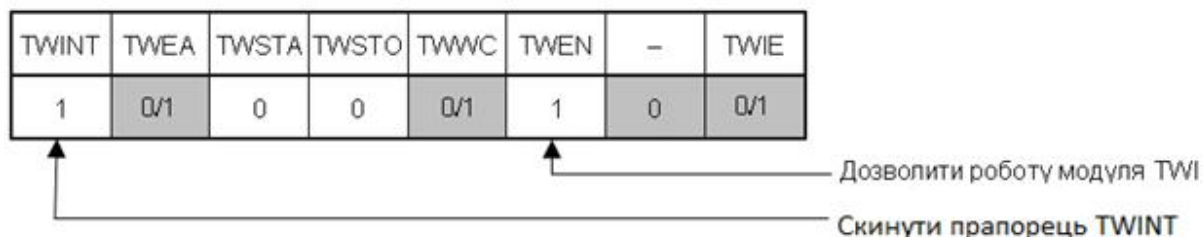
TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
1	0/1	1	0	0/1	1	0	0/1



Дозволити роботу модуля TWI
Сформувати стан СТАРТ
Скинути прапорець TWINT

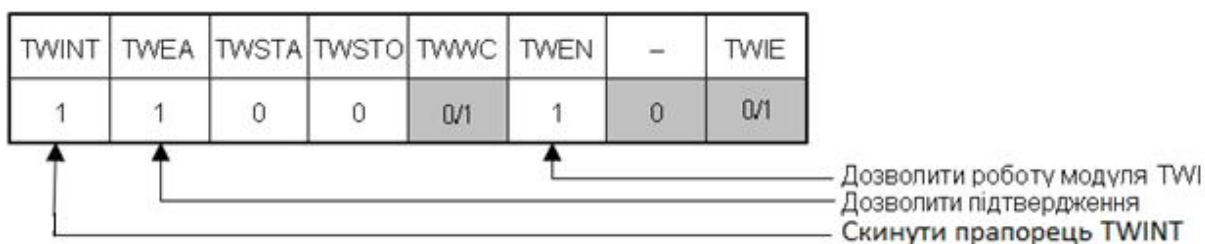
У результаті запису зазначеного значення модуль TWI почне контролювати стан шини і сформує стан СТАРТ одразу ж, як тільки вона стане вільною.

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:



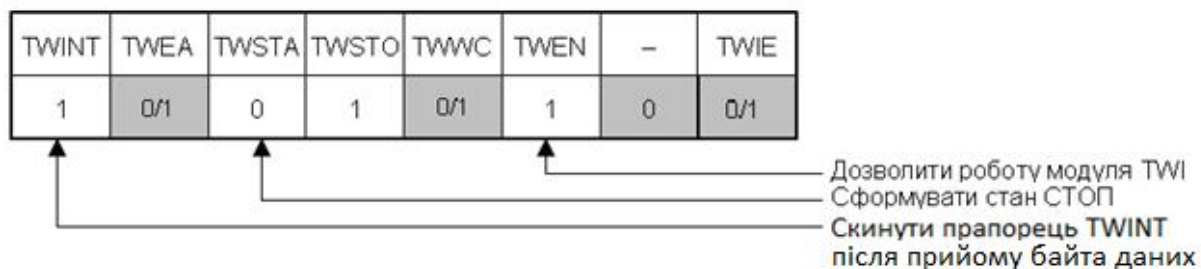
Після передачі адресного пакета і прийому біта підтвердження прапорець TWINT знову встановлюється в «1». Код статусу на цьому етапі може мати одне з наступних значень: \$38, \$40 або \$48. Які дії необхідно почати при виявленні того або іншого коду, докладно розглянуто в таблиці 3.11.

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:

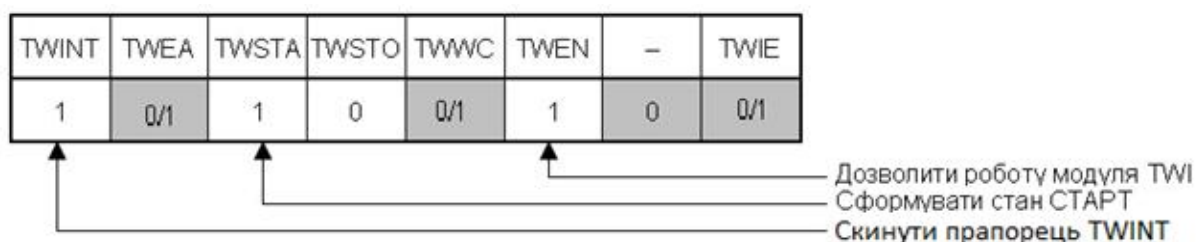


Після прийому байта код статусу має значення: \$50 або \$58. Дії модуля TWI у відповідь на це наведено в таблиці 3.11.

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 у 3.2.3 – програмна реалізація цього алгоритму.

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

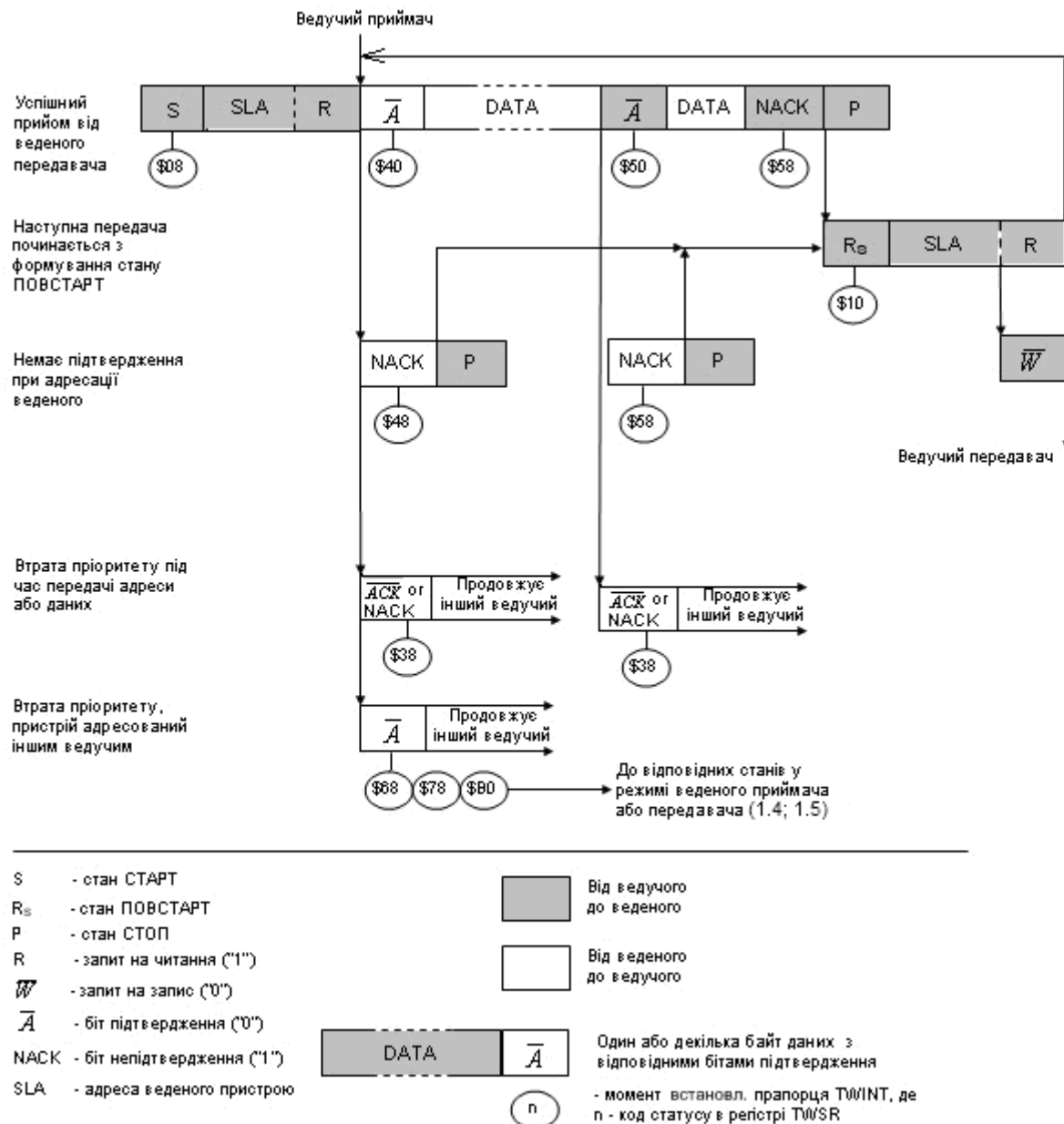
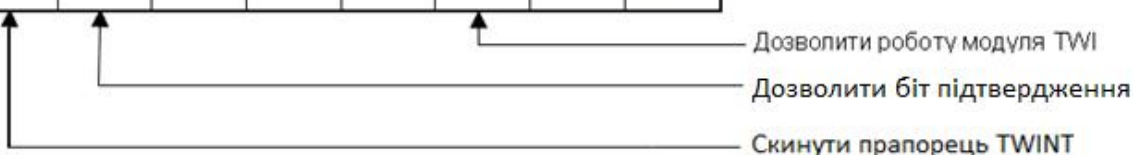


Рисунок 3.21 – Стани модуля TWI у режимі «Ведучий приймач»

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 напряду 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 $TWGCE = \langle 1 \rangle$
		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 $TWGCE = \langle 1 \rangle$; 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 (<i>ACK</i>)	Зчитати дані	0	0	1	0	1	Буде прийнято байт даних і передано <i>NACK</i>
		Зчитати дані	0	0	1	1	1	Буде прийнято байт даних і передано <i>ACK</i>
\$98	Пристрій уже адресовано (загальний виклик): було прийнято байт даних і послано непідтвердження (<i>NACK</i>)	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”.

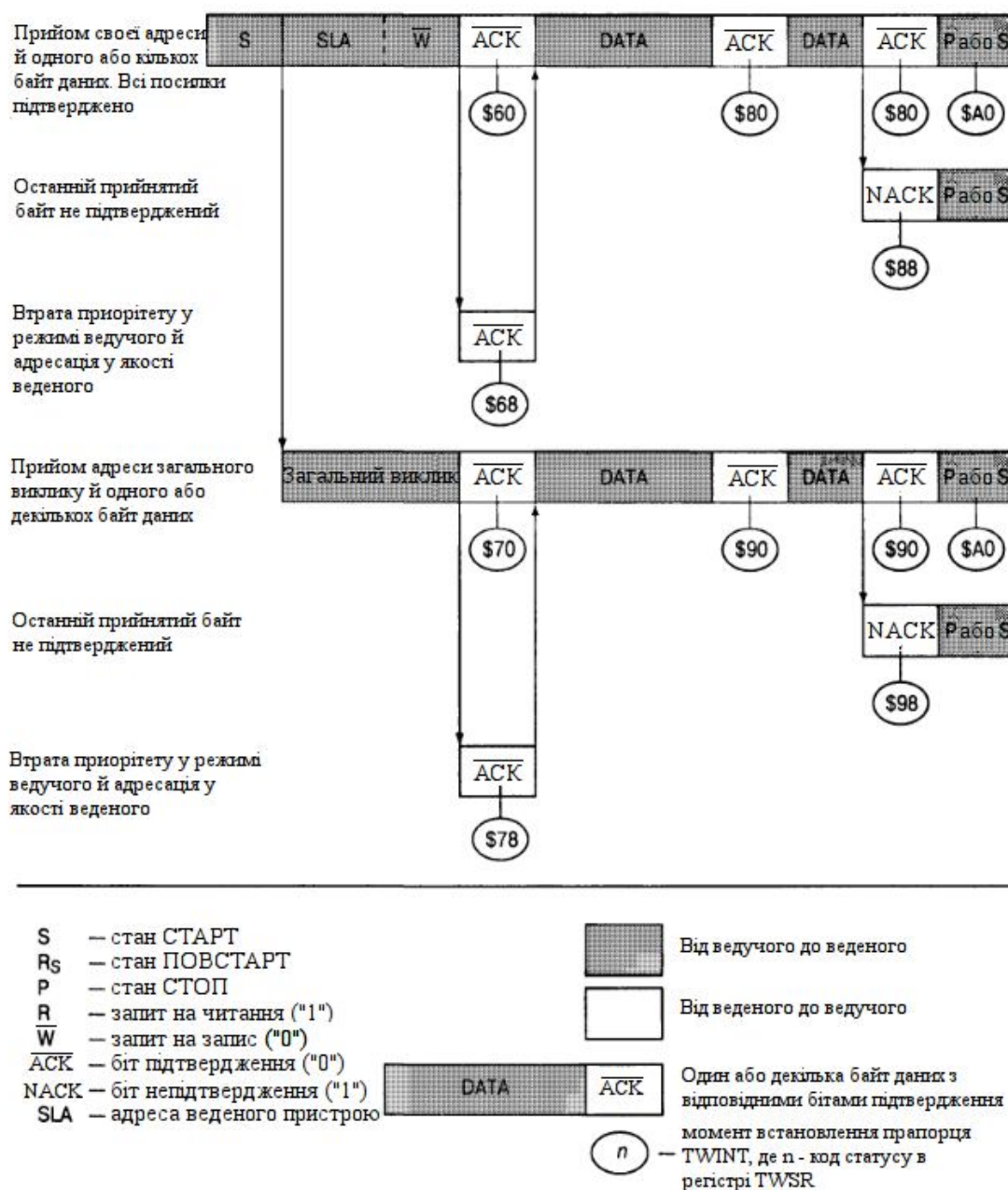
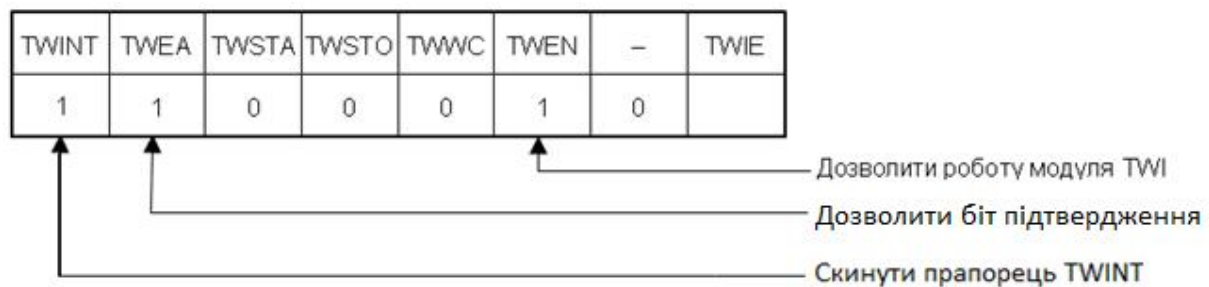


Рисунок 3.22 – Стани модуля TWI у режимі “Ведений приймач”

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 напряду 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 непідтвердження (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
SC8	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».

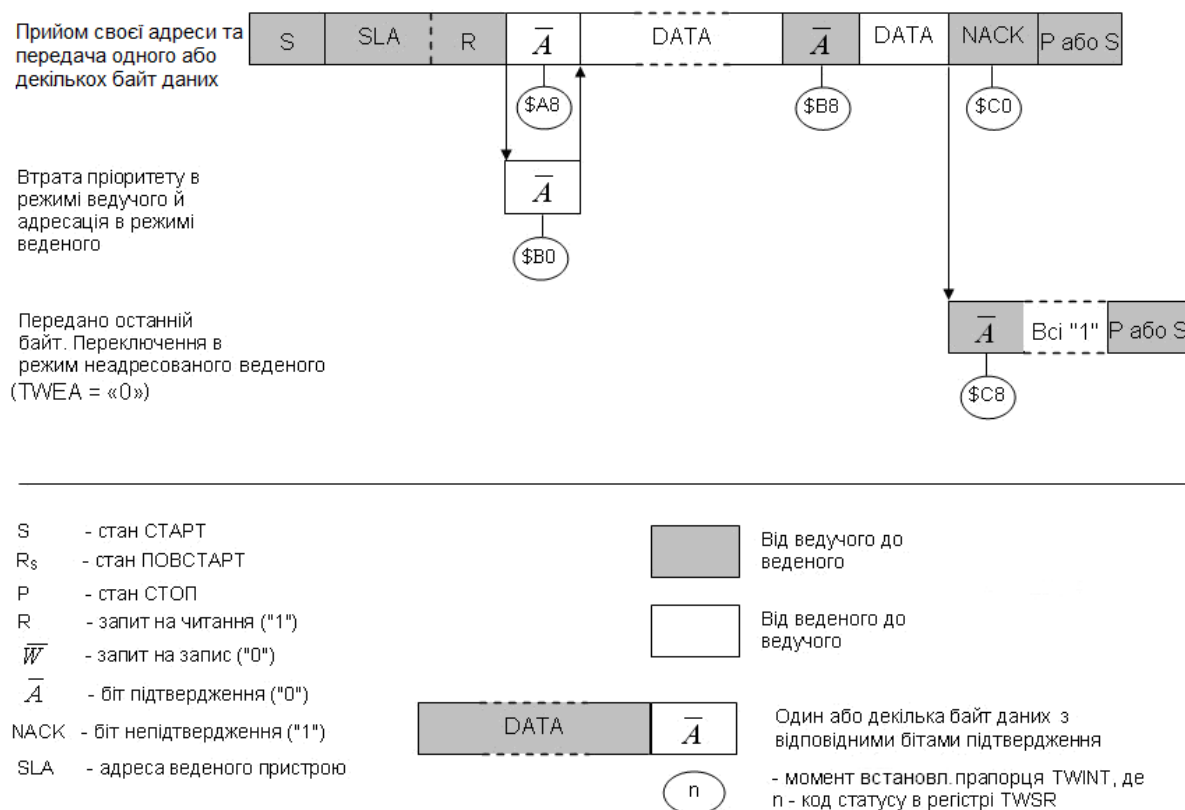


Рисунок 3.23 – Стани модуля TWI у режимі «Ведений передавач»

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 напряду. 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 напряду. 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]

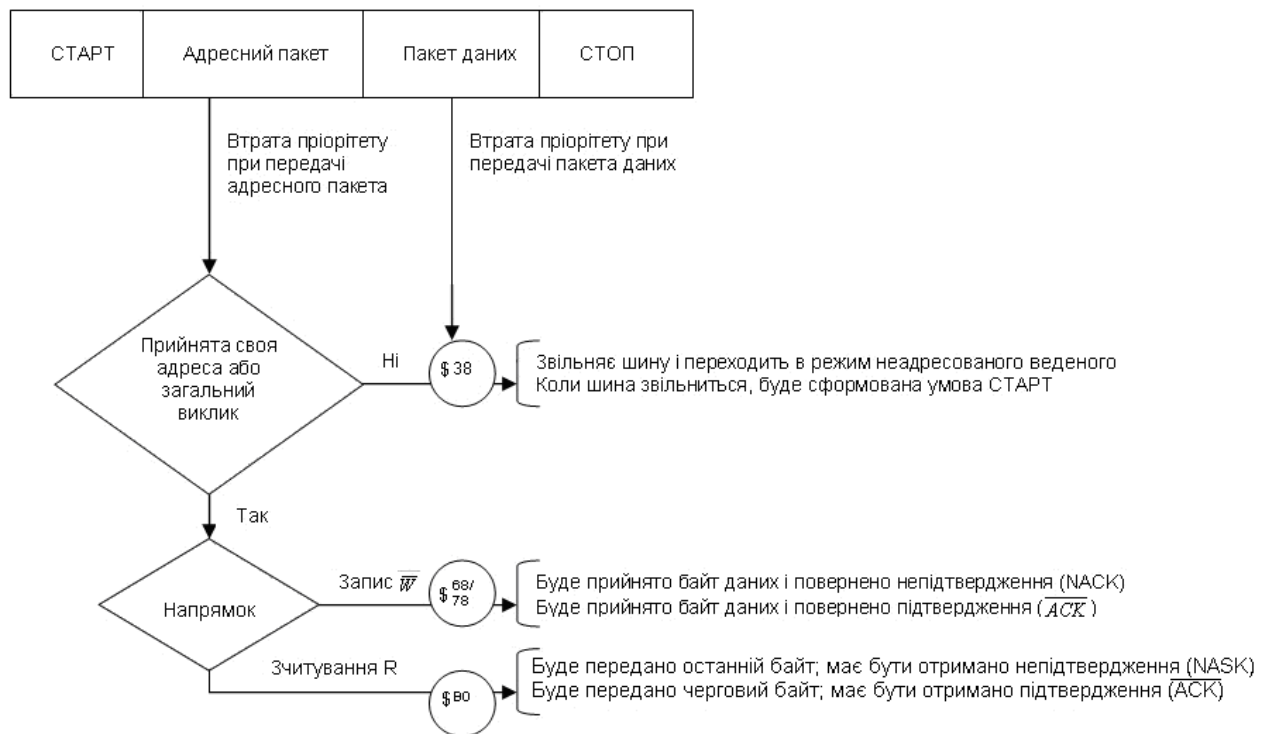


Рисунок 3.25 – Стани модуля TWI під час арбітражу

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_ACK : означає, що ведучим було передано пакет $SLA + \overline{W}$ і отримано підтвердження (ACK);
- MT_DATA_ACK : означає, що ведучим було передано пакет даних та отримано підтвердження (ACK);
- MR_SLA_ACK : означає, що ведучим було передано пакет $SLA + R$ і отримано підтвердження (ACK);
- MR_DATA_ACK : означає, що ведучим було прийнято байт даних і передано підтвердження (ACK);

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

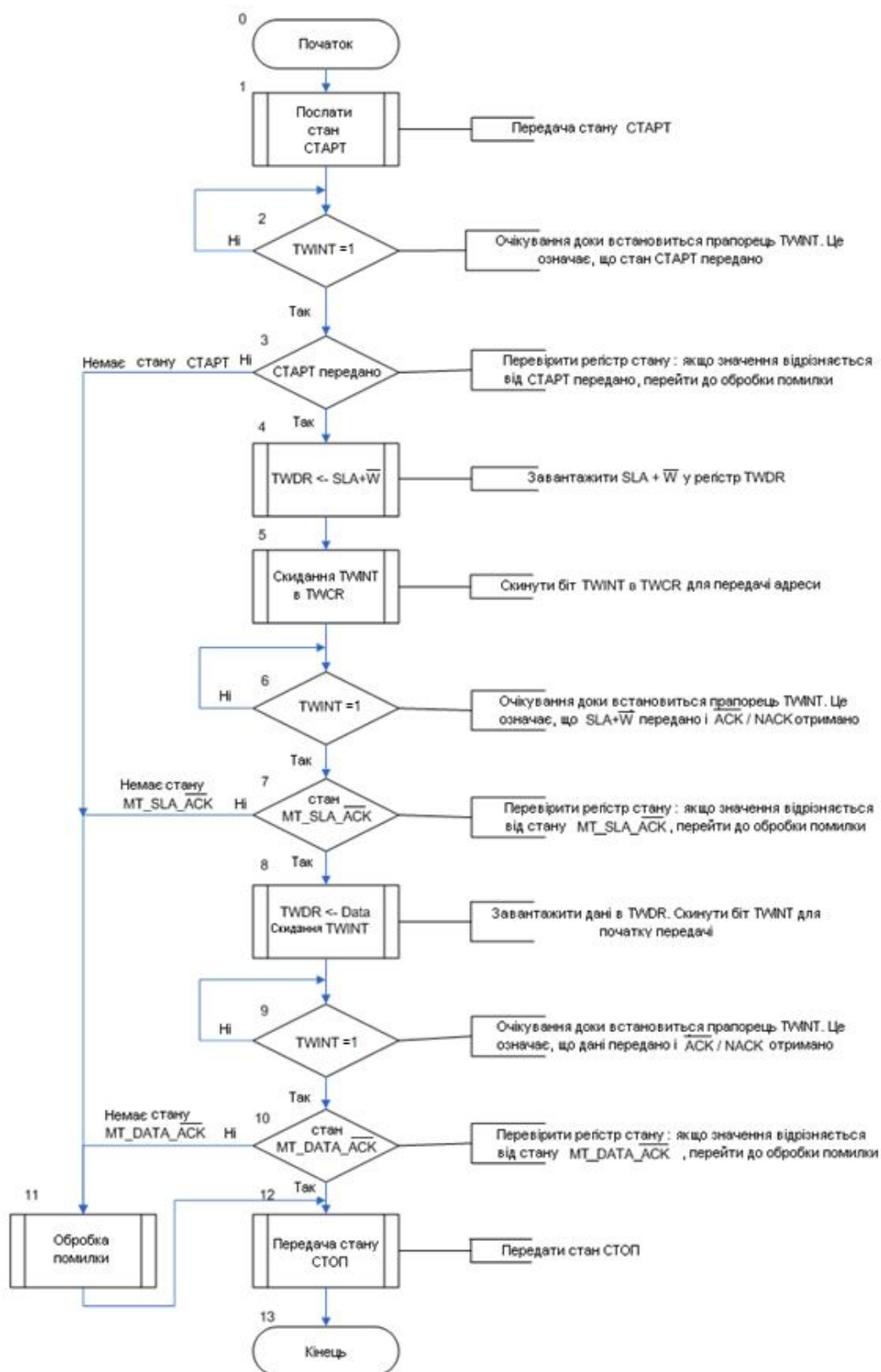


Рисунок 3.26 – Схема алгоритму роботи модуля в режимі «Ведучий передавач»

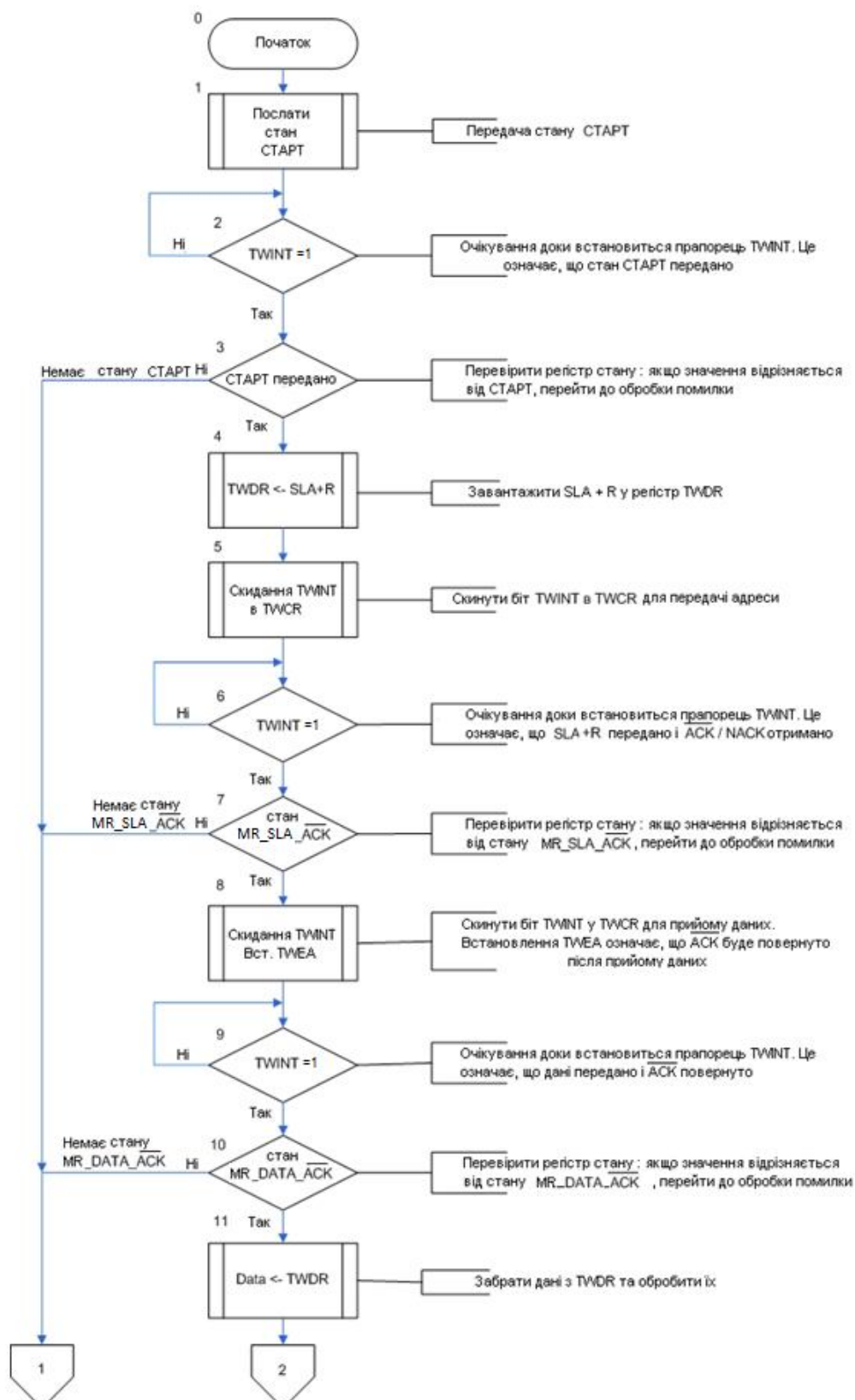


Рисунок 3.27 – Схема алгоритму роботи модуля у режимі «Ведучий приймач».

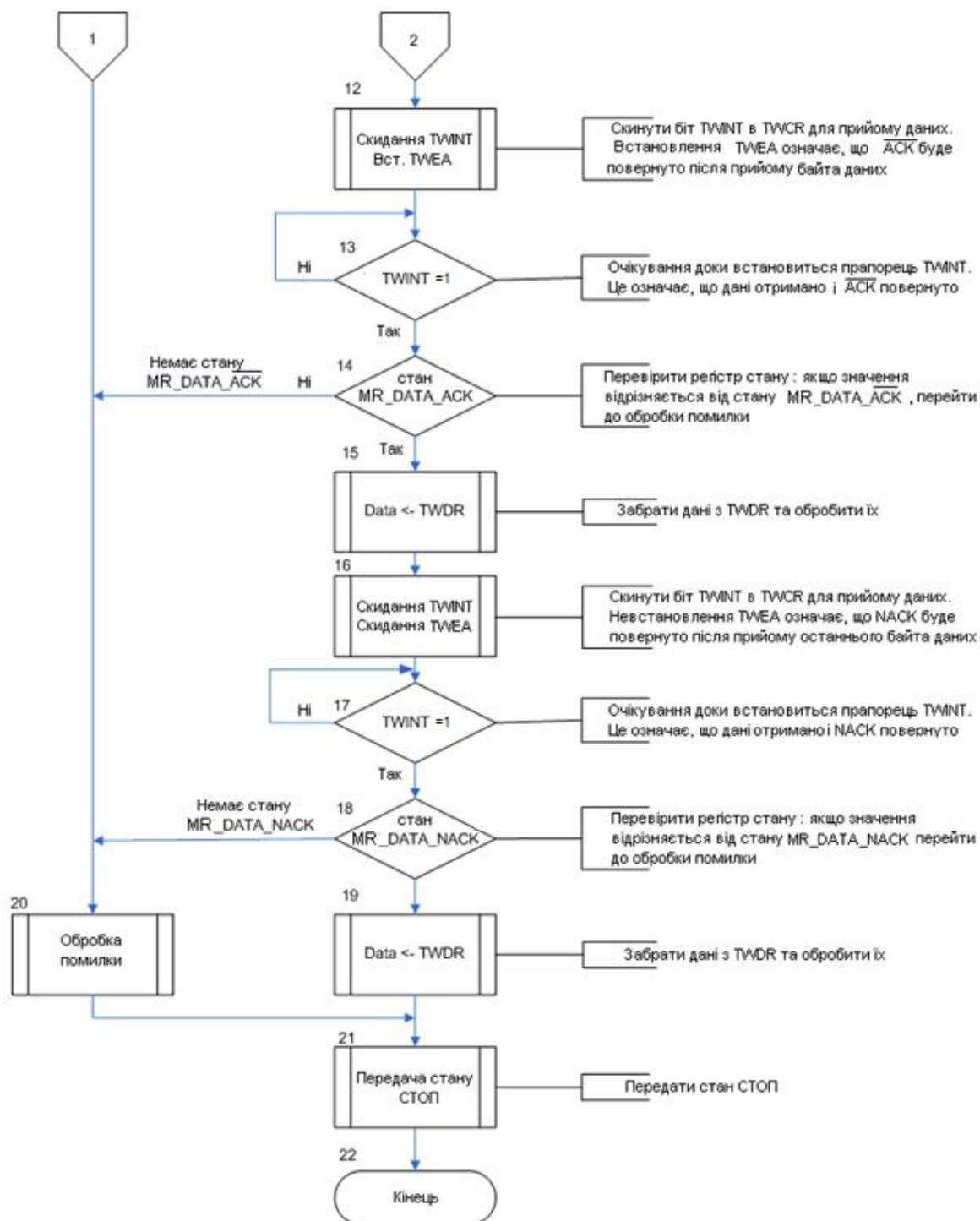


Рисунок 3.27 – Схема алгоритму роботи модуля у режимі «Ведучий приймач» (продовження)

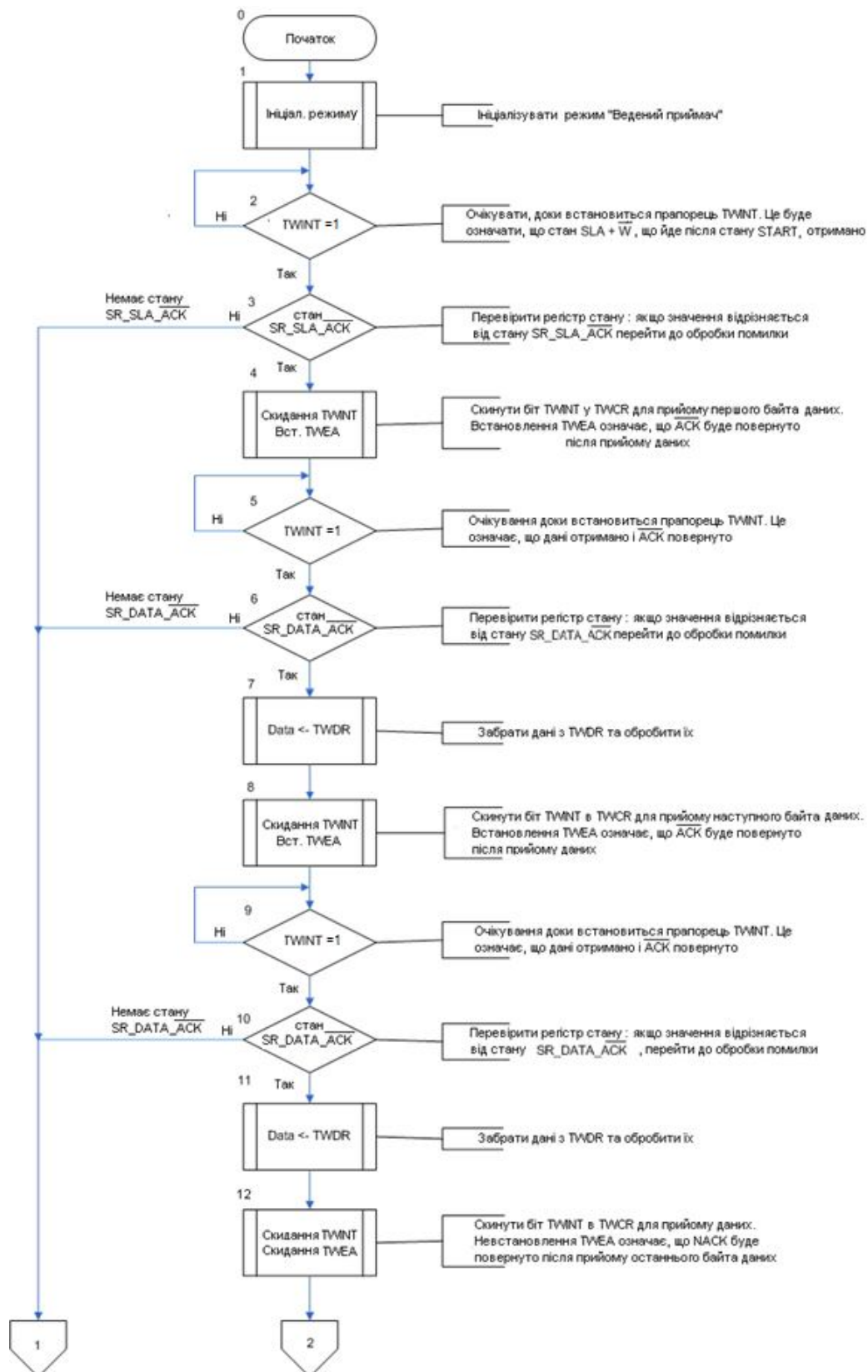


Рисунок 3.28 – Схема алгоритму роботи модуля у режимі «Ведений приймач».

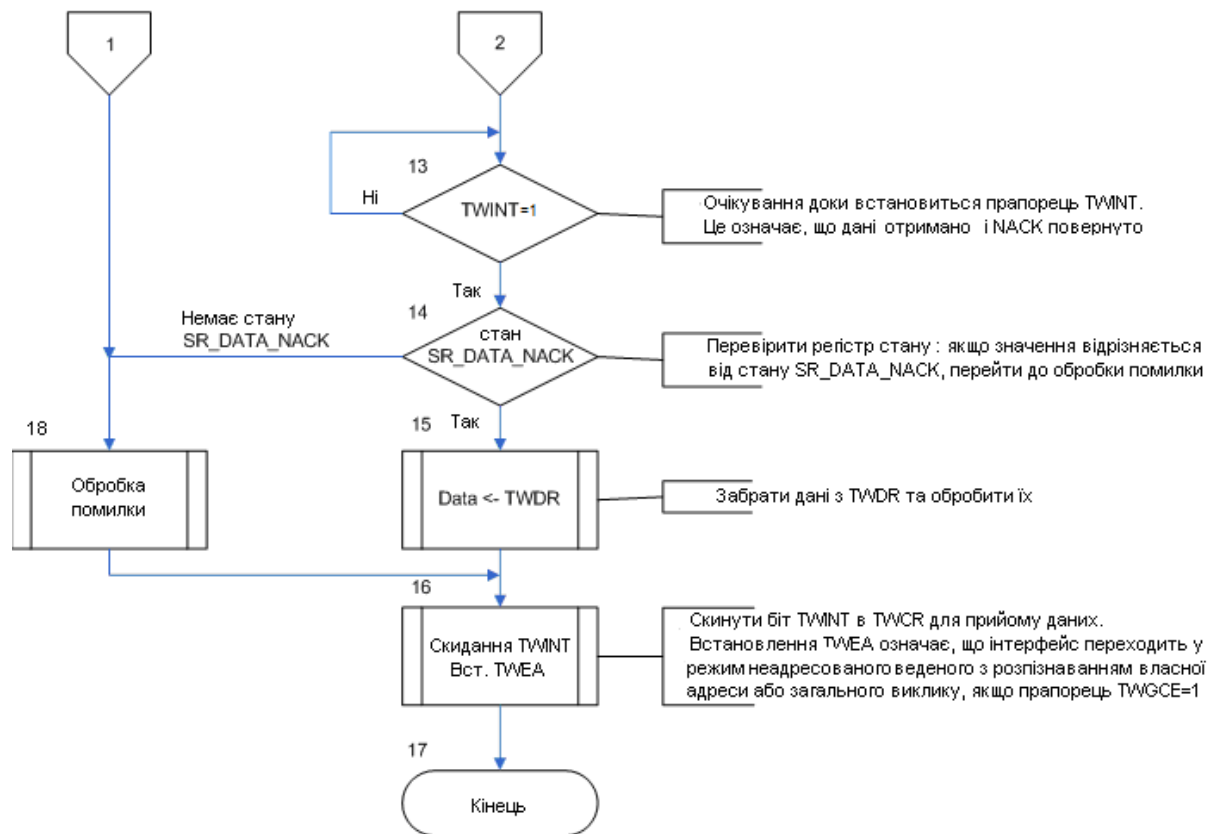


Рисунок 3.28 – Схема алгоритму роботи модуля у режимі «Ведений приймач» (продовження)

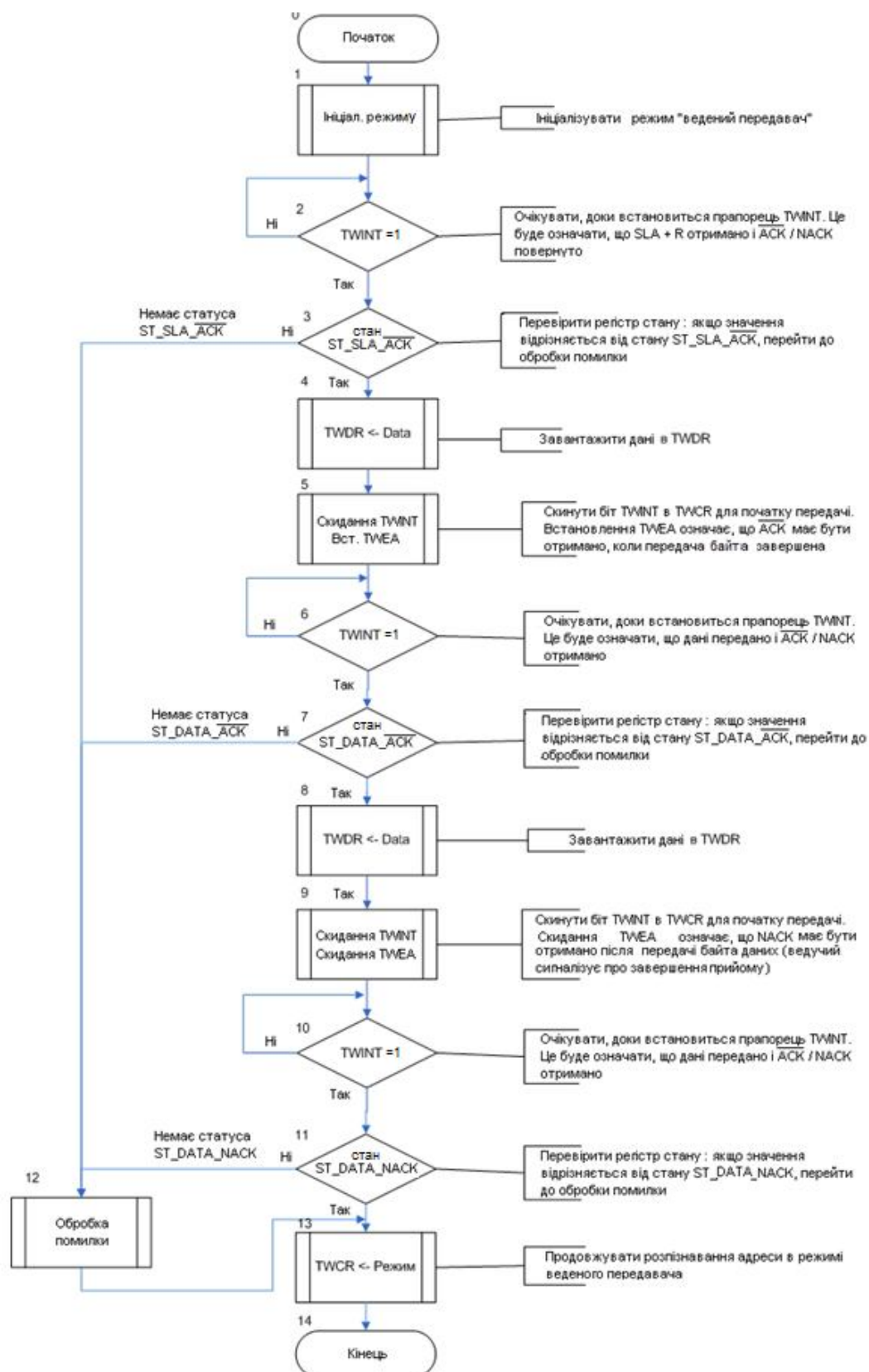


Рисунок 3.29 – Схема алгоритму роботи модуля в режимі «Ведений передавач».

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)
wait1:  out TWCR, r16                ; послати стан START (блок 1)
        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  $SLA + \overline{W}$  у регістр
        out TWDR, r16                ; TWDR (блок 4)
        ldi r16,
            (1<<TWINT) | (1<<TWEN)   ; скинути біт TWINT в TWCR
        out TWCR, r16                ; для передачі адреси (блок 5)
wait2:  in r16, TWCR                ; очікувати доки встановиться
        sbrs r16, TWINT             ; прапорець TWINT.
        rjmp wait2                  ; Це буде означати, що
                                    ;  $SLA + \overline{W}$  передано і  $\overline{ACK}/NACK$ 
                                    ; 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,
            (1<<TWINT) | (1<<TWEN)   ; скинути біт TWINT для
                                    ; 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
                                   ;  $\overline{ACK}/NACK$  отримано (блок 9)
                                   ; перевірити регістр стану.
        in r16, TWSR              ; Якщо значення відрізняється
        cpi r16, $28              ; від  $MT\_DATA\_ACK$ , перейти
        brne ERROR                ; 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
                                   ;  $\overline{ACK}/NACK$  отримано (блок 6)

```


<pre> in r16, TWSR cpi r16, \$40 brne ERROR ldi r16, (1<<TWINT) (1<<TWEA) (1<<TWEN) out TWCR, r16 wait7: in r16, TWCR sbrs r16, TWINT rjmp wait7 in r16, TWSR cpi r16, \$50 brne ERROR in r16, TWDR ;: ldi r16, (1<<TWINT) (1<<TWEA) (1<<TWEN) out TWCR, r16 </pre>	<pre> ; перевірити регістр стану. ; Якщо значення відрізняється ; від MR_SLA_ACK, перейти до ; обробки помилки (блок 7) ; скинути біт TWINT в TWCR ; для прийому даних. ; Встановлення TWEA означає, ; що ACK буде повернуто після ; отримання даних (блок 8) ; очікувати встановлення ; прапорця TWINT. Це означає, ; що дані отримано і ; ACK повернуто (блок 9) ; перевірити регістр стану. ; Якщо значення відрізняється ; від MR_DATA_ACK, перейти ; до обробки помилки (блок 10) ; забрати дані з TWDR (блок 11) ; обробити дані ; скинути біт TWINT в TWCR ; для прийому даних. (блок 12) </pre>
<pre> ; отримання інших байтів ... ; отримання передостаннього байта wait8: in r16, TWCR sbrs r16, TWINT rjmp wait8 in r16, TWSR cpi r16, \$50 brne ERROR in r16, TWDR ;... ldi r16, ~(1<<TWINT) (1<<TWEN) out TWCR, r16 </pre>	<pre> ; очікувати встановлення ; прапорця TWINT. Це означає, ; що дані отримано та ACK ; повернуто (блок 13) ; перевірити регістр стану. ; Якщо значення відрізняється ; від MR_DATA_ACK, перейти до ; обробки помилки (блок 14) ; забрати дані з TWDR (блок 15) ; обробити дані ; скинути біт TWINT в TWCR для ; прийому даних. Не </pre>

```

; 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  $SLA + \overline{W}$ ,
                           ; що йде за станом START,
                           ; отримано. (блок 2)
      in r16, TWSR          ; перевірити регістр стану.
      cpi r16, $60          ; Якщо значення відрізняється
      brne ERROR           ; від стану  $SR\_SLA\_ACK$ ,

```

		; 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 $\overline{ACK}$ 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 $\overline{ACK}$ 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 $\overline{ACK}$ 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 $\overline{ACK}$
		; 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

```

wait13: in r16, TWCR          ; (блок 12)
        sbrs r16, TWINT      ; очікувати встановлення прапорця
        rjmp wait13         ; TWINT. Це означає, що дані
                              ; 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)

```

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

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) З яких двох ліній складається шина інтерфейсу I²C (TWI)?
- 2) Наведіть та поясніть структурну схему типової мережі, яка використовує для обміну даними шину I²C (TWI).
- 3) Опишіть роботу пристроїв, які використовують для обміну інтерфейс I²C (TWI).
- 4) Опишіть чотири можливі режими обміну даними інтерфейсом I²C (TWI).
- 5) Поясніть призначення станів СТАРТ і СТОП шини TWI. Яким чином вони формуються?
- 6) У чому полягає задача розподілу пріоритетів при підключенні до шини TWI декількох ведучих пристроїв? Як вона вирішується?
- 7) Опишіть формат адресного пакету та пакету даних.
- 8) Наведіть та поясніть структурну схему типового модуля TWI.
- 9) Що задає контролер швидкості передачі?
- 10) Які два вузли входять до складу блоку шинного інтерфейсу? Які їх функції?
- 11) Що перевіряє блок контролю адреси?
- 12) При виникненні яких подій блок керування здійснює формування запиту на переривання?
- 13) З яких трьох частин складається будь-який етап взаємодії прикладної програми з модулем TWI? Поясніть цю взаємодію.
- 14) Назвіть деякі пристрої, які підтримують інтерфейс TWI.
- 15) У яких режимах може працювати модуль TWI, реалізований у мікроконтролерах сімейства Mega?
- 16) Опишіть роботу модуля TWI у режимі «Ведучий передавач».
- 17) Опишіть роботу модуля TWI у режимі «Ведучий приймач».
- 18) Опишіть роботу модуля TWI у режимі «Ведений приймач».

- 19) Опишіть роботу модуля TWI у режимі «Ведений передавач».
- 20) Які можливості надає використання стану ПОВСТАРТ?
- 21) Що трапиться, якщо адресація пристрою відбудеться при перебуванні мікроконтролера у «сплячому» режимі?
- 22) За якими сценаріями може розвиватися процес арбітражу?

ЛІТЕРАТУРА [1...10]

4 ПОСЛІДОВНИЙ ПЕРИФЕРІЙНИЙ ІНТЕРФЕЙС SPI

4.1 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 X Mega, 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.

4.2 Використання модуля SPI

4.2.1 Загальна характеристика архітектури модуля SPI

Структурна схема модуля SPI наведена на рисунку 4.1. 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 4.1 наведені виводи, які використовуються модулем SPI деяких МК-в сімейства Mega.

При включеному модулі SPI режим роботи зазначених виводів (напрямок передачі даних) перевизначається згідно з таблицею 4.2. Напрямок передачі даних визначається станом відповідного розряду регістра DDRB.

Як видно з таблиці 4.2, 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 4.2, а короткий опис функцій розрядів регістра наведено в таблиці 4.3. Докладне використання різних розрядів регістра буде описано далі [2].

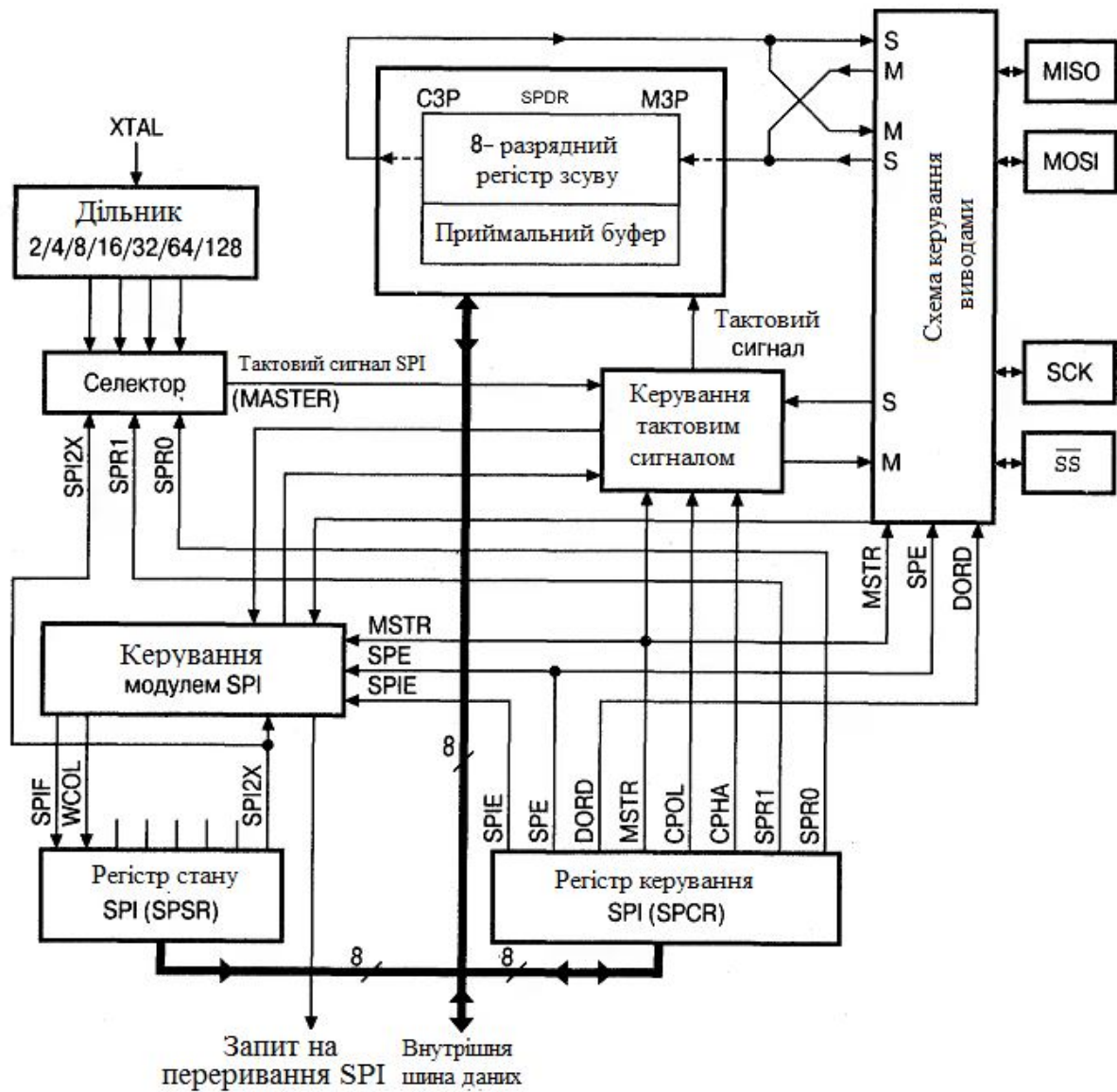


Рисунок 4.1 – Структурна схема модуля SPI

Таблиця 4.1 – 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 4.2 – 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 4.2 – Формат регістра SPCR

Таблиця 4.3 – 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 4.3, а призначення його розрядів описано в таблиці 4.4.

	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 4.3 – 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 4.4. Вивід SCK ведучого мікроконтролера є виходом тактового сигналу, а веденого мікроконтролера – входом.

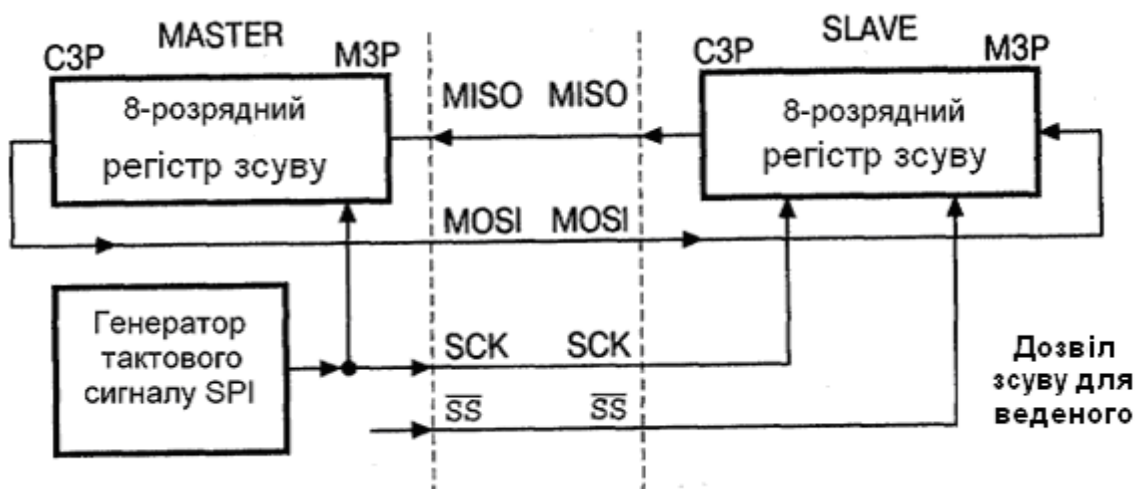


Рисунок 4.4 – З'єднання мікроконтролерів інтерфейсом SPI

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 4.5, 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.

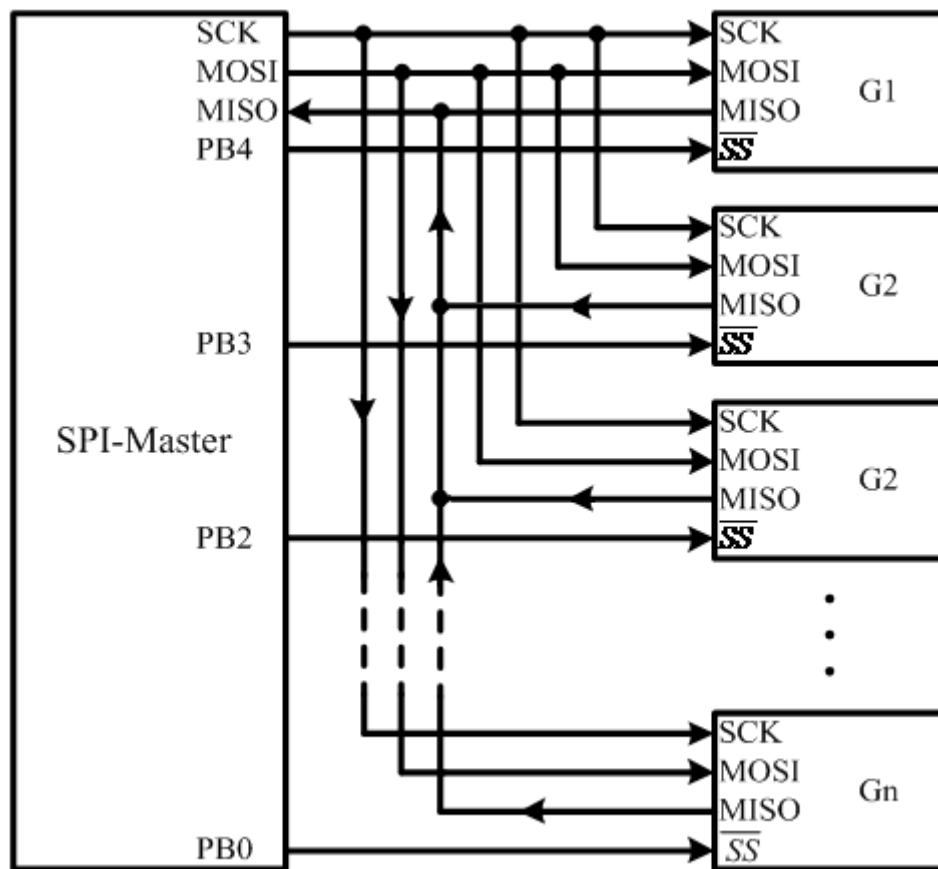


Рисунок 4.5 – Підключення декількох пристроїв до інтерфейсу ведучого SPI

Блокам G1, G2 ... Gn на рисунку 4.5 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).

4.2.2 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 регістра SPCR (таблиця 4.5).

Таблиця 4.5 – 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 4.6,4.7)
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 4.6); «1» – обробка даних виконується за заднім фронтом імпульсів сигналу SCK (для CPOL = «0» – за спадаючим фронтом, а для CPOL = «1» – за наростаючим фронтом) (рисунок 4.7)

Формати обміну даними через SPI, які відповідають цим режимам наведені на рисунку 4.6 і рисунку 4.7 (передача ведеться від старшого розряду до молодшого).

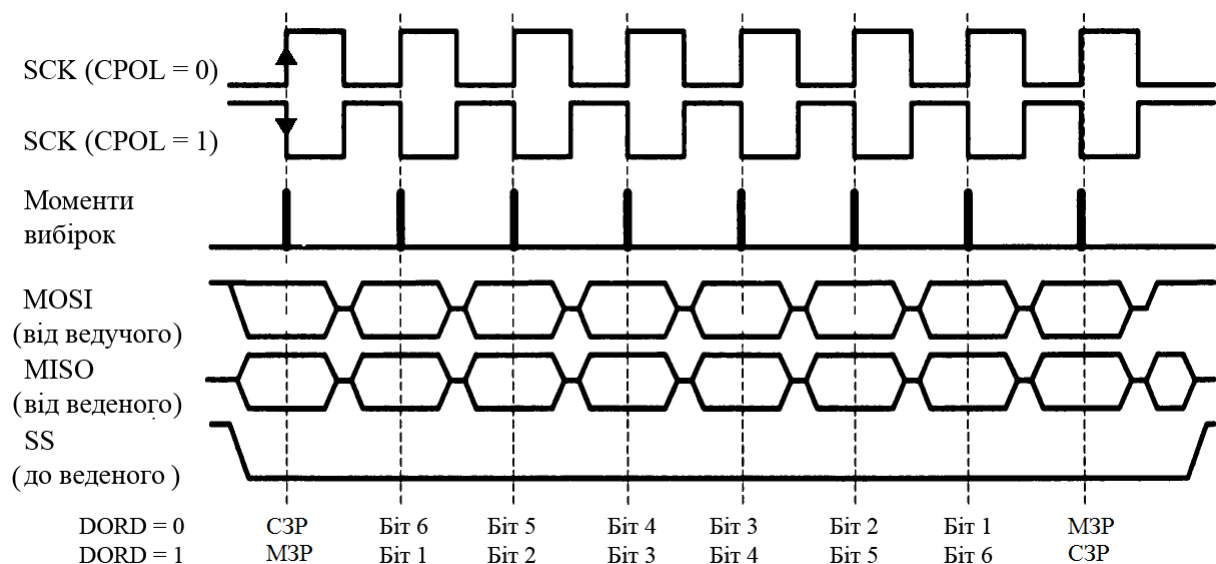


Рисунок 4.6 – Передача даних при $CPHA = \text{«0»}$ (режими 0, 1)

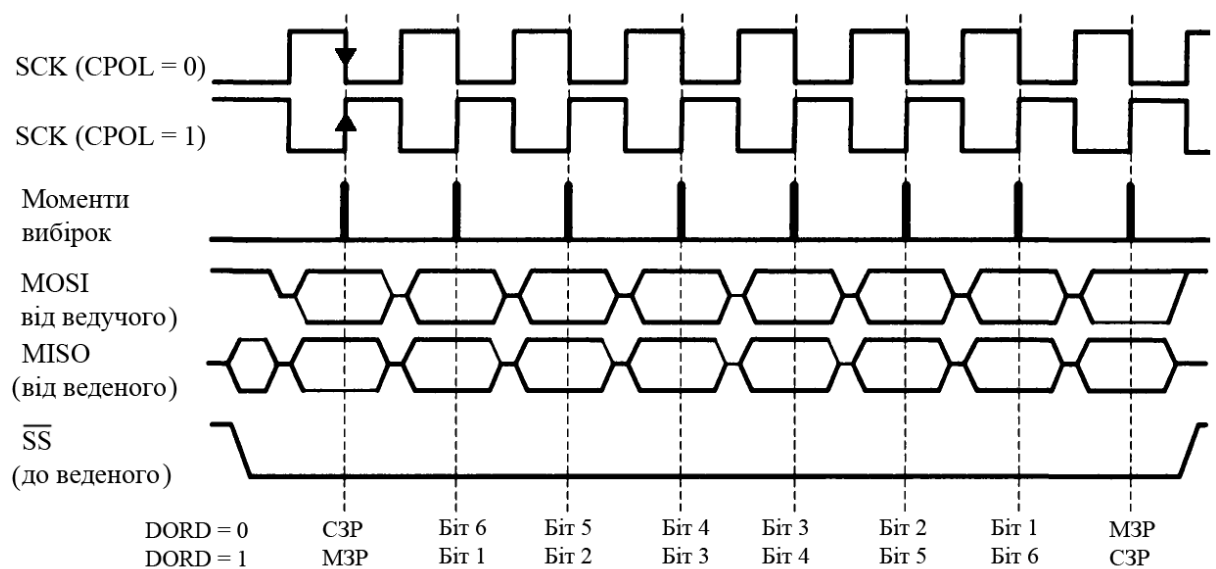


Рисунок 4.7 – Передача даних при $CPHA = \text{«1»}$ (режими 2, 3)

У режимах 0, 1 ($CPHA = 0$) 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 4.6). 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 4.4) 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 4.6 відмічений на лінії «Моменти вибірок».

Другим перепадом імпульсів на лінії SCK відбувається зсув інформації вліво відповідно до рисунку 4.4. 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 4.8, а), що буде визначати початок обміну наступним байтом.

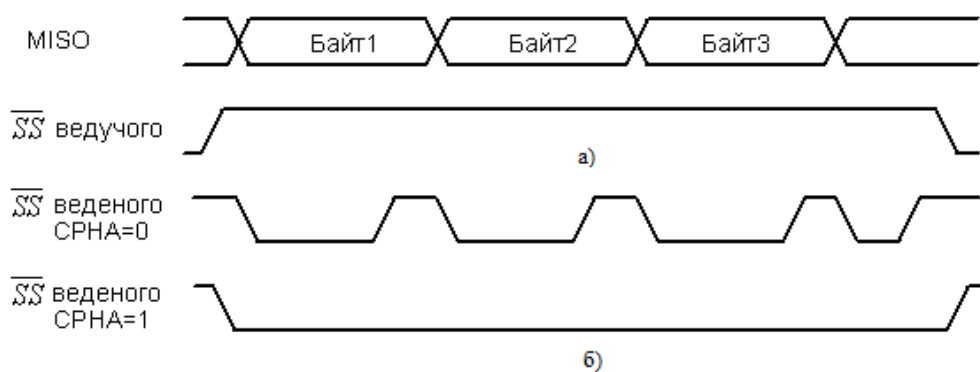


Рисунок 4.8 – Вплив сигналу $\overline{SS}$ на початок обміну даними

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 4.7).

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 посилки 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 4.8,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 4.7). 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 4.6). 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 4.6 – 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.

4.2.3 Використання виводу $\overline{SS}$

Цей вивід призначений для вибору активного веденого пристрою і у режимі «Slave» завжди є входом (рисунки 4.5). При подачі на нього напруги НИЗЬКОГО рівня модуль SPI активується і вивід MISO переключається в режим виведення даних (якщо це задано користувачем, таблиця 4.2). 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 4.2).

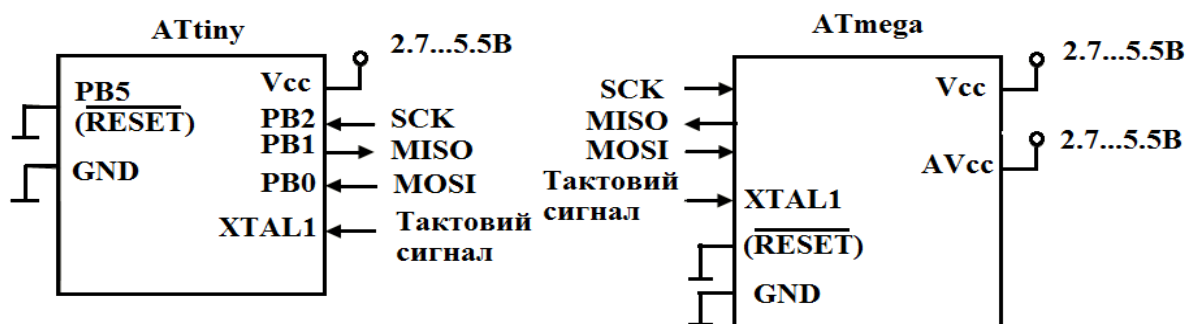
2) Встановлюється прапорець SPIF регістра SPSR, генеруючи запит на переривання від SPI. Якщо переривання від SPI дозволені і прапорець I регістра SREG встановлений в «1», відбувається запуск підпрограми обробки переривання.

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».

4.3 Режим програмування пам'яті послідовним каналом

Режим програмування пам'яті послідовним каналом підтримується всіма мікроконтролерами сімейства Mega, XMeta, 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 4.9.



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

Рисунок 4.9 – Включення мікроконтролерів у режимі програмування послідовним каналом

Як видно з рисунка 4.9, 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 4.7.

Таблиця 4.7 – 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	ATmega1281xATmega2561x	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 4.7).

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.

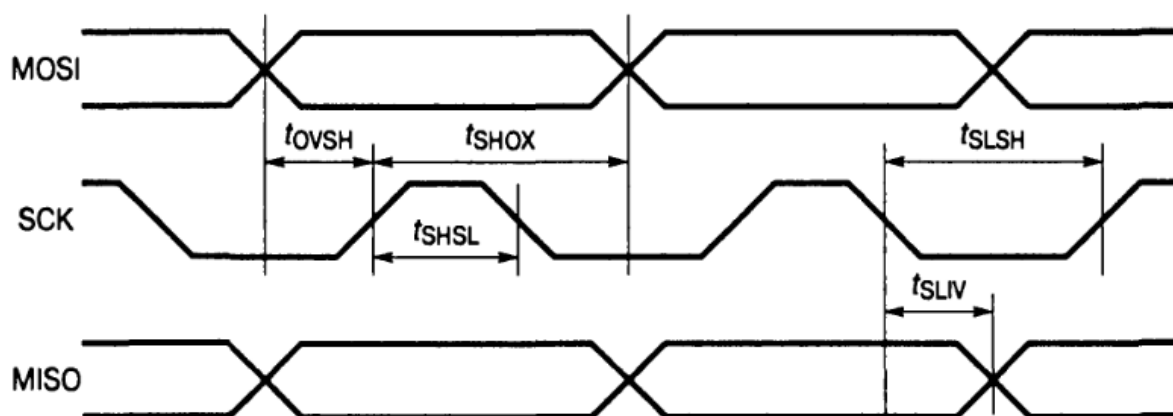


Рисунок 4.10 – Часові діаграми сигналів при програмуванні послідовним каналом

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 4.11).

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

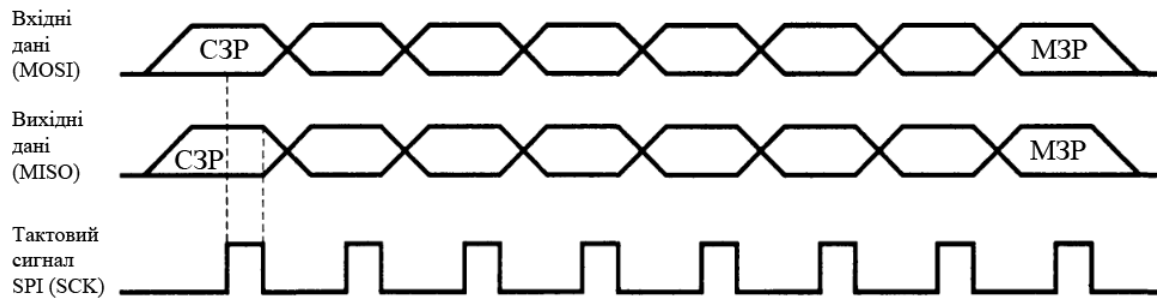


Рисунок 4.11 – Передача даних при програмуванні послідовним каналом

4.4 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 4.8, в якій наведені деякі мікросхеми, що містять інтерфейс SPI.

Таблиця 4.8 – 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

Продовження таблиці 4.8

Найменування	Короткий опис
DS1267	Виробник: Dallas Semiconductors <i>Двоканальний цифровий монолітний потенціометр</i> Стандартні опори: 10 Ом, 50 кОм і 100 кОм. Діапазони температур: – комерційний: 0...70°C; – індустріальний: -40...+85°C Напруга живлення: 5 В
MAX5481	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
ADS7816	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)
TLC1518	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
DAC7512	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

Продовження таблиці 4.8

Найменування	Короткий опис
AD5308	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
DS1307	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
AT45DB161B	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
W25Q32	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
DS1626	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
MAX6627	Виробник: Maxim <i>Вимірювач віддаленої температури в корпусі SOT</i> Температурний діапазон: -55...+125°C Діапазон напруги живлення: 3...5.5 В

Продовження таблиці 4.8

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

4.5 Універсальний послідовний інтерфейс USI

4.5.1 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 4.12.

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.

4.5.2 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 4.9.

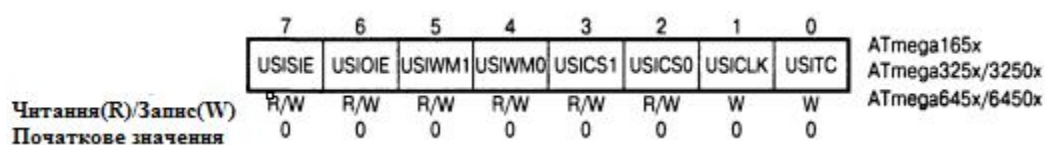


Рисунок 4.13 – Формат регістра USICR

Таблиця 4.9 – 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 4.9

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 4.11
2	USICS0	
1	USICLK	<u>Строб тактового сигналу</u> Призначення цього біта залежить від встановлень бітів USICS1:0 (див. таблицю 4.11). 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 записується 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 4.10 – 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 4.10

USIWM1	USIWM0	Опис
0	1	<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)
1	0	<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
1	1	<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

Таблиця 4.11 – 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 4.14, а опис його бітів наведено в таблиці 4.12.

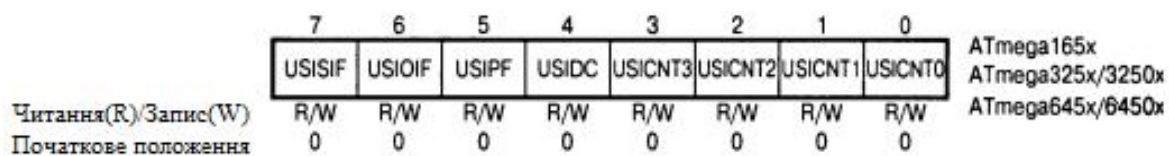


Рисунок 4.14 - Формат регістра USISR

Таблиця 4.12 – 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. 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). Джерело тактового сигналу визначається станом бітів USICS1:0 та USICLK регістру керування USICR (таблиця 4.11)
2	USICNT2	
1	USICNT1	
0	USICNT0	

4.5.3 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.

4.5.3.1 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 частини 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 4.15.

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.

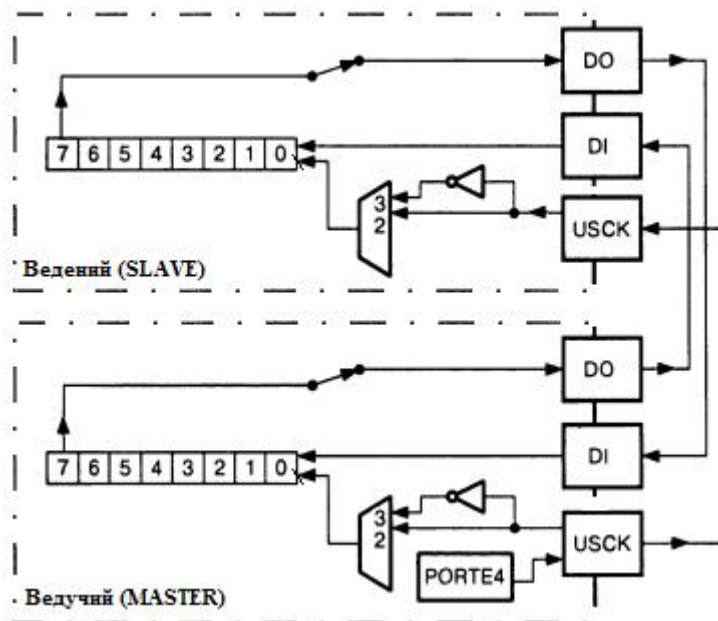


Рисунок 4.15 - Модуль USI в трьохпровідному режимі

Часові діаграми обміну в трьохпровідному режимі представлені на рисунку 4.16.

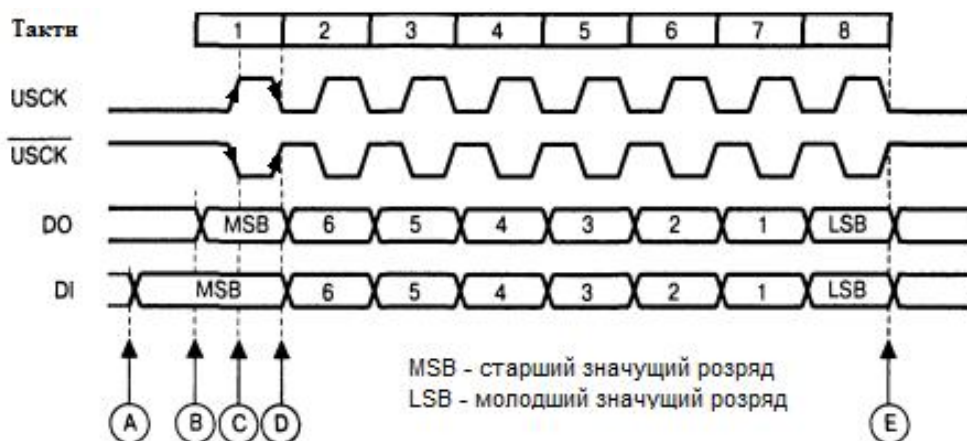


Рисунок 4.16 – Часові діаграми обміну в трьохпровідному режимі

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 ($USICS0 = 1$). 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 A та B 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 A та B 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 C та D на рисунку 4.16). Значення біта даних на входах (DI) ведучого та веденого пристрою зчитується за першим фронтом сигналу USCK (точка C), 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.

4.5.3.2 Двопровідний режим

У двопровідному режимі модуль USI може використовуватися для обміну шиною TWI (I^2C), 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 4.17.

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.

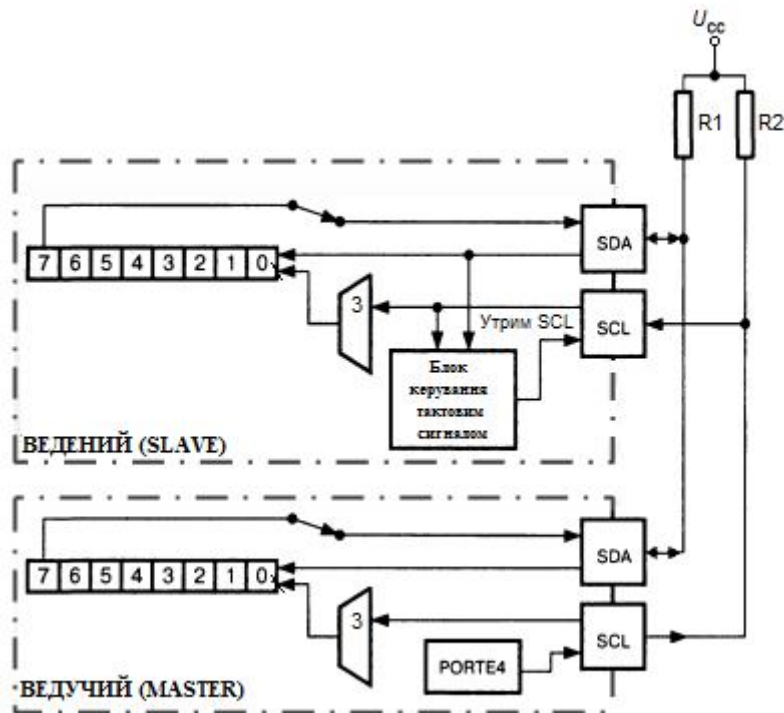


Рисунок 4.17 – Модулі USI в двопровідному режимі

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 4.18.

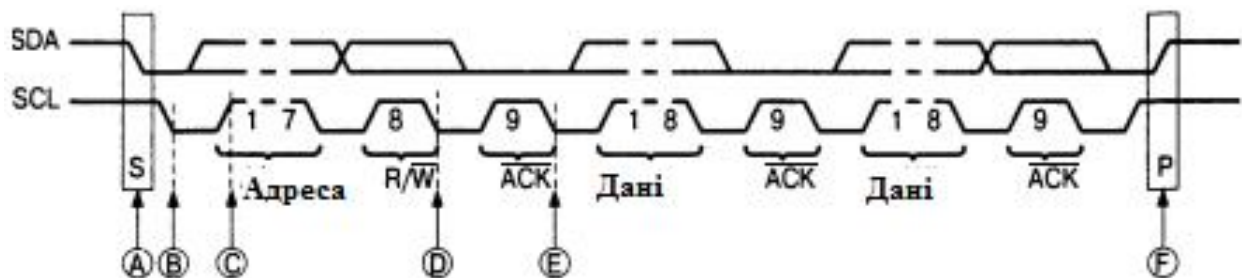


Рисунок 4.18 – Часові діаграми обміну в двопровідному режимі

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 4.18). 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 4.19), 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 4.19 наведена функціональна схема вузла виявлення стану СТАРТ модуля USI.

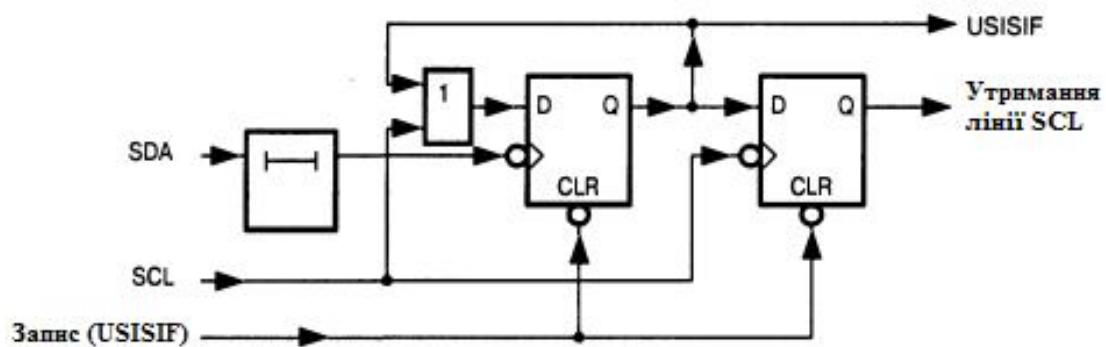


Рисунок 4.19 – Схема виявлення стану СТАРТ

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.

4.5.3.3 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.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) На яку відстань і з якою швидкістю можна здійснювати обмін даними через інтерфейс SPI?
- 2) У яких режимах може працювати мікроконтролер при обміні даними інтерфейсом SPI?
- 3) Скільки та які виводи мікроконтролера використовує модуль SPI?
- 4) Який регістр призначений для керування модулем SPI?
- 5) Який регістр використовується для контролю стану модуля, а також для додаткового керування швидкістю обміну?
- 6) Як задати режим роботи мікроконтролера в режимі «MASTER»?
- 7) Як визначити кінець передачі байта?
- 8) За якої умови, при передачі даних від ведучого до веденого, можлива передача у зворотному напрямі?
- 9) Що означає те, що у модулі реалізована одинарна буферизація при передачі та подвійна при прийомі?
- 10) Чи можна до інтерфейсу підключати декілька периферійних пристроїв?
- 11) Назвіть кількість режимів передачі даних та принцип їх роботи.
- 12) Що є джерелом тактового сигналу при роботі модуля?
- 13) При яких частотах гарантується функціонування мікроконтролера в режимі «Slave»?
- 14) Яких суттєвих покращень можна досягти при використанні модуля універсального послідовного інтерфейсу (USI)?

- 15) Які три лінії введення/виведення мікроконтролера задіює модуль USI?
- 16) В яких основних режимах може працювати модуль USI? Для обміну якими шинами він може використовуватись у кожному з цих режимів?
- 17) З яких етапів складається обмін в режимі SPI?
- 18) Поясніть структурну схему вузла виявлення стану СТАРТ модуля USI.
- 19) Перелічіть відомі вам варіанти альтернативного використання вузлів модуля USI.

ЛІТЕРАТУРА [1...10]

5 ТАЙМЕРИ

5.1 Восьмирозрядні таймери. Модулятор. Вартовий таймер

5.1.1 ОСОБЛИВОСТІ АРХІТЕКТУРИ ТАЙМЕРІВ AVR-МІКРОКОНТРОЛЕРІВ

5.1.1.1 Загальні відомості

Всі сучасні мікроконтролери мають у своєму складі модуль таймерів/лічильників (T/CNT). Розглянемо архітектуру модуля таймерів AVR-мікроконтролерів на прикладі сімейства Mega. Ці мікроконтролери, залежно від моделі, мають у своєму складі до шести таймерів/лічильників загального призначення (таблиця 5.1) [1,2].

Таблиця 5.1 – Таймери/лічильники загального призначення

Таймер/лічильник	ATmega8515x	ATmega8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/3250x/645x/6450x	ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x
Таймер/лічильник T0 (8-розрядний)	•	•	•	•	•	•	•	•	•	•

Продовження таблиці 5.1

Таймер/лічильник	ATmega8515x	ATmega8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/3250x/645x/645	ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x
Таймер/лічильник T1 (16-розрядний)	•	•	•	•	•	•	•	•	•	•
Таймер/лічильник T2 (8-розрядний)		• ¹⁾	• ¹⁾	•	• ¹⁾	• ¹⁾	• ¹⁾	• ¹⁾	• ¹⁾	• ¹⁾
Таймер/лічильник T3 (16-розрядний)				•		•				•
Таймер/лічильник T4 (16-розрядний)										•
Таймер/лічильник T5 (16-розрядний)										•
¹⁾ Асинхронний таймер/лічильник.										

Як видно з таблиці, у всіх моделях мікроконтролерів сімейства присутні, як мінімум, два таймери/лічильники - T0 і T1. Таймер/лічильник T0 має мінімальний набір функцій, який залежить від моделі мікроконтролера. В деяких моделях він може використовуватися тільки для відліку й вимірювання часових інтервалів, або як лічильник зовнішніх подій. В інших моделях до цих функцій додається можливість генерації сигналів із широтно-імпульсною модуляцією (ШІМ) фіксованої розрядності, а також можливість працювати в асинхронному режимі в якості годинника реального часу. Таймер/лічильник T1 також може використовуватися для відліку часових інтервалів і як лічильник зовнішніх подій. Крім того, він може виконувати запам'ятовування свого стану за зовнішнім сигналом. Як і таймер/лічильник T0, він може працювати як широтно-імпульсний модулятор, але вже змінної розрядності й до того ж

багатоканальний (кількість каналів залежить від моделі). Таймер/лічильник T2 повністю аналогічний таймеру/лічильнику T0. При наявності в мікроконтролері обох таймерів/лічильників, один з них може працювати в асинхронному режимі, а інший - як лічильник зовнішніх подій. Таймери/лічильники T3...T5 за функціональними можливостями ідентичні таймеру/лічильнику T1. У складі всіх мікроконтролерів сімейства є також вартовий таймер, що є неодмінним атрибутом всіх сучасних мікроконтролерів. Цей таймер дозволяє уникнути несанкціонованого зациклення програми, яке може виникати з тих чи інших причин.

5.1.1.2 Призначення виводів таймерів/лічильників

Кожен таймер/лічильник використовує один або більше виводів мікроконтролера. Як правило, ці виводи - лінії портів введення/виведення загального призначення, а функції, реалізовані цими виводами при роботі разом з таймерами/лічильниками, є їхніми альтернативними функціями. Всі виводи AVR-мікроконтролерів сімейства Mega, що використовуються таймерами/лічильниками загального призначення, зведені в таблиці 5.2.

У таблиці 5.2 також зазначено функції цих виводів. Не забувайте про те, що при використанні альтернативних функцій ліній портів введення/виведення необхідно, як правило, самостійно сконфігурувати ці виводи відповідно до їхнього функціонального призначення.

5.1.1.3 Переривання від таймерів/лічильників

Для дозволу/заборони переривань від таймерів/лічильників T0, T1 і T2 у частині моделей призначений регістр TIMSK (Timer/Counter Interrupt MaSK Register — регістр маски переривань від таймерів/лічильників).

У нових моделях кількість таких регістрів (3...6) дорівнює числу таймерів/лічильників у конкретній моделі. Так само відбувається і з регістрами, що містять прапорці переривань. Назва і адреси цих регістрів наведено у таблиці 5.3.

Таблиця 5.2 – Виводи, що використовуються таймерами/лічильниками загального призначення

Назва	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x/325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x/1280x/2560x	Опис
T0	PB0	PB0	PD4	PB0	-	PD4	PB0	PB0	PG4	PG4	PD7	PD7	Вхід зовнішнього сигналу таймера T0
OC0	PB0	PB3	-	PB3	PB4	-	PB0	-	-	-	-	-	Вихід схеми порівняння таймера T0
OC0A	-	-	-	-	-	PD6	-	PB3	PB4	PB4	PB7	PB7	
OC0B	-	-	-	-	-	PD5	-	PB4	-	-	PG5	PG5	
T1	PB1	PB1	PD5	PB1	PD6	PD5	PB1	PB1	PG3	PG3	PD6	PD6	Вхід зовнішнього сигналу таймера T1
ICP	PE0	-	-	-	-	-	-	-	-	-	-	-	Вхід захоплення таймера T1
ICP1	-	PD6	PB0	PD6	PD4	PB0	PE0	PD6	PD0	PD0	PD4	PD4	
OC1A	PD5	PD5	PB1	PD5	PB5	PB1	PD5	PD5	PB5	PB5	PB5	PB5	Вихід схеми порівняння таймера T1
OC1B	PE2	PD4	PB2	PD4	PB6	PB2	PE2	PD4	PB6	PB6	PB6	PB6	
OC1C	-	-	-	-	PB7	-	-	-	-	-	PB7	PB7	
T2	-	-	-	-	PD7	-	-	-	-	-	-	-	Вхід зовнішнього сигналу таймера T2
OC2	-	-	PB0	PD7	PB7	-	PB1	-	-	-	-	-	Вихід схеми порівняння таймера T2
OC2A	-	-	-	-	-	PB3	-	PD7	PB7	PB7	PB4	PB4	
OC2B	-	-	-	-	-	PD3	-	PD6	-	-	-	PH6	

Продовження таблиці 5.2

Назва	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x/325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x/1280x/2560x	Опис
T3	-	-	-	-	PE6	-	-	-	-	-	PE6	PE6	Вхід зовнішнього сигналу таймера T3
ICP3	-	-	-	-	PE7	-	PD3	-	-	-	PE7	PE7	Вхід захоплення таймера T3
OC3A	-	-	-	-	PE3	-	PD4	-	-	-	PE3	PE3	Вихід схеми порівняння таймера T3
OC3B	-	-	-	-	PE4	-	PB4	-	-	-	PE4	PE4	
OC3C	-	-	-	-	PE5	-	-	-	-	-	PE5	PE5	
T4	-	-	-	-	-	-	-	-	-	-	-	PH7	Вхід зовнішнього сигналу таймера T4
ICP4	-	-	-	-	-	-	-	-	-	-	-	PL0	Вхід захоплення таймера T4
OC4A	-	-	-	-	-	-	-	-	-	-	-	PH3	Вихід схеми порівняння таймера T4
OC4B	-	-	-	-	-	-	-	-	-	-	-	PH4	
OC4C	-	-	-	-	-	-	-	-	-	-	-	PH5	
T5	-	-	-	-	-	-	-	-	-	-	-	PL2	Вхід зовнішнього сигналу таймера T5
ICP5	-	-	-	-	-	-	-	-	-	-	-	PL1	Вхід захоплення таймера T5
OC5A	-	-	-	-	-	-	-	-	-	-	-	PL3	Вихід схеми порівняння таймера T5
OC5B	-	-	-	-	-	-	-	-	-	-	-	PL4	
OC5C	-	-	-	-	-	-	-	-	-	-	-	PL5	
TOSC1	-	PC6	PB6	PC6	PG4	PB6	PD4	PC6	- ¹⁾	- ¹⁾	PG4	PG4	Вхід для підключення резонатора
TOSC2	-	PC7	PB7	PC7	PG3	PB7	PD5	PC7	- ¹⁾	- ¹⁾	PG3	PG3	Вихід для підключення резонатора

¹⁾ Окремі контакти введення/виведення, суміщені із XTAL1/XTAL2.

Таблиця 5.3 – Регістри керування перериваннями від таймерів/лічильників

Модель	Таймер/ лічильник	Дозвіл переривань		Прапорці переривань	
		Регістр	Адреса	Регістр	Адреса
ATmega8515x	T0,T1	TIMSK	\$39 (\$59)	TIFR	\$38 (\$58)
ATmega8535x	T0, T1, T2	TIMSK	\$39 (\$59)	TIFR	\$38 (\$58)
ATmega8x/16x/32x	T0, T1, T2	TIMSK	\$39 (\$59)	TIFR	\$38 (\$58)
ATmega64x/128x	T0, T1, T2	TIMSK	\$37 (\$57)	TIFR	\$36 (\$56)
	T1, T3	ETIMSK	(\$7D)	ETIFR	(\$7C)
ATmega162x	T0, T1, T2	TIMSK	\$39 (\$59)	TIFR	\$38 (\$58)
	T1, T3	ETIMSK	(\$7D)	ETIFR	(\$7C)
ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega165x/325x/645x, ATmega3250x/6450x	T0	TIMSK0	(\$6E)	TIFR0	\$15 (\$35)
	T1	TIMSK1	(\$6F)	TIFR1	\$16 (\$36)
	T2	TIMSK2	(\$70)	TIFR2	\$17 (\$37)
ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x	T0	TIMSK0	(\$6E)	TIFR0	\$15 (\$35)
	T1	TIMSK1	(\$6F)	TIFR1	\$16 (\$36)
	T2	TIMSK2	(\$70)	TIFR2	\$17 (\$37)
	T3	TIMSK3	(\$71)	TIFR3	\$18 (\$38)
	T4	TIMSK4	(\$72)	TIFR4	\$19 (\$39)
	T5	TIMSK5	(\$73)	TIFR5	\$1A (\$3A)

Формат регістра TIMSK для деяких моделей показаний на рисунку 5.1, а опис його розрядів наведено у таблиці 5.4.

	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0	ATmega8x ATmega163x
Зчитування (R) / Запис (W)	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	TOIE1	OCIE1A	OCIE1B	–	TICIE1	–	TOIE0	OCIE0	ATmega8515x
Зчитування (R) / Запис (W)	R/W	R/W	R/W	R	R/W	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	ATmega16x ATmega323x ATmega32x ATmega64x ATmega128x
Зчитування (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	

	7	6	5	4	3	2	1	0	
	TOIE1	OCIE1A	OCIE1B	TOIE2	TICIE1	OCIE2	TOIE0	OCIE0	ATmega161x
Зчитування (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	

	7	6	5	4	3	2	1	0	
	TOIE1	OCIE1A	OCIE1B	OCIE2	TICIE1	TOIE2	TOIE0	OCIE0	ATmega162x
Зчитування (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 5.1 - Формат регістра TIMSK

Для дозволу/заборони переривань від таймерів/лічильників T1 і T3 у моделях ATmega64x й ATmega128x є ще один регістр — ETIMSK (Extended Timer/Counter Interrupt MaSK Register — розширений регістр маски переривань від таймерів/лічильників). Формат цього регістра наведений на рисунку 5.2, а опис розрядів — у таблиці 5.5.

	7	6	5	4	3	2	1	0	
	–	–	TICIE3	OCIE3A	OCIE3B	TOIE3	OCIE3C	OCIE1C	ATmega64x ATmega128x
Зчитування (R) / Запис (W)	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

Рисунок 5.2 - Формат регістра ETIMSK

Таблиця 5.4 – Розряди регістра TIMSK

Назва	Опис
OSIE2	Прапорець дозволу переривання за подією «Збіг» таймера/лічильника T2
TOIE2	Прапорець дозволу переривання за переповненням таймера/лічильника T2
TICIE1	Прапорець дозволу переривання за подією «Захоплення» таймера/лічильника T1
OSIE1A	Прапорець дозволу переривання за подією «Збіг А» таймера/лічильника T1
OSIE1B	Прапорець дозволу переривання за подією «Збіг В» таймера/лічильника T1
TOIE1	Прапорець дозволу переривання за переповненням таймера/лічильника T1
OSIE0	Прапорець дозволу переривання за подією «Збіг» таймера/лічильника T0
TOIE0	Прапорець дозволу переривання за переповненням таймера/лічильника T0

Для дозволу якого-небудь переривання від таймера/лічильника необхідно встановити в «1» відповідний розряд регістра TIMSK(ETIMSK) і, зрозуміло, прапорець I регістра SREG. Для індикації надходження переривань від таймерів T0, T1 і T2 призначений регістр TIFR (Timer/Counter Interrupt Flag Register — регістр прапорців переривань від таймерів/лічильників). Формат цього регістра для ряду моделей показаний на рисунку 5.3, а опис його розрядів наведено в таблиці 5.6.

Таблиця 5.5 – Розряди регістра ETIMSK

Розряд	Назва	Опис
7,6	-	Не використовується, вважається за «0»
5	TICIE3	Прапорець дозволу переривання за подією «Захоплення» таймера/лічильника T3
4	OCIE3A	Прапорець дозволу переривання за подією «Збіг А» таймера/лічильника T3
3	OCIE3B	Прапорець дозволу переривання за подією «Збіг В» таймера/лічильника T3
2	TOIE3	Прапорець дозволу переривання за переповненням таймера/лічильника T3
1	OCIE3C	Прапорець дозволу переривання за подією «Збіг С» таймера/лічильника T3
0	OCIE1C	Прапорець дозволу переривання за подією «Збіг С» таймера/лічильника T1

Для індикації настання переривань від таймерів/лічильників T1 і T3 у моделях ATmega64x й ATmega128x призначено регістр ETIFR. Формат цього регістра наведено на рисунку 5.4, а опис його розрядів - у таблиці 5.7. Формати регістрів прапорців переривань та масок переривань для таймерів інших мікроконтролерів сімейства Mega наведені у [1,2].

Коли настає яка-небудь подія, відповідний прапорець регістра TIFR (ETIFR) встановлюється в «1». При запуску підпрограми обробки переривання він апаратно скидається в «0». Будь-який прапорець може бути скинуто в «0» також програмно шляхом запису в нього логічної «1».

	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	–	TOV0	ATmega8x ATmega163x
Зчитування (R) / Запис (W)	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	TOV1	OCF1A	OCF1B	–	ICF1	–	TOV0	OCF0	ATmega8515x
Зчитування (R) / Запис (W)	R/W	R/W	R/W	R	R/W	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	ATmega16x ATmega323x ATmega32x ATmega64x ATmega128x
Зчитування (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	

	7	6	5	4	3	2	1	0	
	TOV1	OCF1A	OCF1B	TOV2	ICF1	OCF2	TOV0	OCF0	ATmega161x
Зчитування (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	

	7	6	5	4	3	2	1	0	
	TOV1	OCF1A	OCF1B	OCF2	ICF1	TOV2	TOV0	OCF0	ATmega162x
Зчитування (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 5.3 – Формат регістра TIFR

	7	6	5	4	3	2	1	0	
	–	–	ICF3C	OCF3A	OCF3B	TOV3	OCF3C	OCF1C	ATmega64x ATmega128x
Зчитування (R) / Запис (W)	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

Рисунок 5.4 – Формат регістра ETIFR

Таблиця 5.6 – Розряди регістра TIFR

Назва	Опис
OCF2	Прапорець переривання за подією «Збіг» таймера/лічильника T2
TO2	Прапорець переривання за переповненням таймера/лічильника T2
ICF1	Прапорець переривання за подією «Захоплення» таймера/лічильника T1
OCF1A	Прапорець переривання за подією «Збіг А» таймера/лічильника T1
OCF1B	Прапорець переривання за подією «Збіг В» таймера/лічильника T1
TOV1	Прапорець переривання за переповненням таймера/лічильника T1
OCF0	Прапорець переривання за подією «Збіг» таймера/лічильника T0
TOV0	Прапорець переривання за переповненням таймера/лічильника T0

5.1.1.4 Попередні дільники таймерів/лічильників

Блоки попередніх дільників призначені для формування тактових сигналів для таймерів/лічильників f_{clkTn} , де n – номер таймера. Спрощена структурна схема блоку попереднього дільника таймерів/лічильників, що не мають асинхронного режиму роботи, наведена на рисунку 5.5. Структурна схема блоку попереднього дільника таймерів/лічильників, що мають можливість роботи в асинхронному режимі, наведена на рисунку 5.6.

Як показано на рисунках 5.5 та 5.6, до складу кожного блоку входять власне 10-розрядний попередній дільник і вихідний мультиплексор (МПЛ - селектор тактового сигналу), а для таймерів, що мають можливість роботи в асинхронному режимі, — ще й вхідний мультиплексор тактового сигналу. Попередній дільник таймера/лічильника T0 моделей ATmega64x/128x і таймера/лічильника T2 інших моделей виконаний за схемою, наведеною на рисунку 5.6.

Таблиця 5.7 – Розряди регістра ETIFR

Розряд	Назва	Опис
7, 6	-	Не використовується, вважається за «0»
5	ICF3	Прапорець переривання за подією «Захоплення» таймера/лічильника T3
4	OCF3A	Прапорець переривання за подією «Збіг А» таймера/лічильника T3
3	OCF3B	Прапорець переривання за подією «Збіг В» таймера/лічильника T3
2	TOV3	Прапорець переривання за переповненням таймера/лічильника T3
1	OCF3C	Прапорець переривання за подією «Збіг С» таймера/лічильника T3
0	OCF1C	Прапорець переривання за подією «Збіг С» таймера/лічильника T1

Варто мати на увазі, що всі таймери/лічильники кожної моделі сімейства, що не мають асинхронного режиму роботи, використовують той самий 10-розрядний попередній дільник. При цьому керування тактовим сигналом кожного таймера/лічильника здійснюється індивідуально і буде описано при їхньому огляді. Варто розуміти, що попередні дільники працюють незалежно від таймерів/лічильників. Наслідком цього є, зокрема, невизначений проміжок часу ($1 \dots N+1$ тактів вхідного сигналу, де N — коефіцієнт ділення попереднього дільника) між запуском таймера/лічильника та першим його відліком при роботі разом із попереднім дільником. Щоб уникнути цієї невизначеності, можна скористатися засобами, описаними в наступному підрозділі.

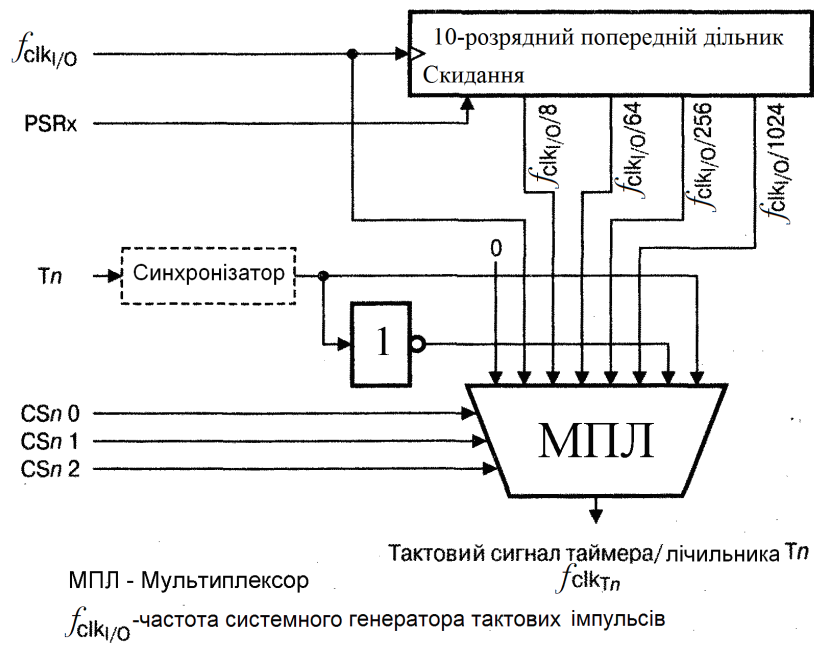


Рисунок 5.5 – Блок попереднього дільника таймера/лічильника без асинхронного режиму

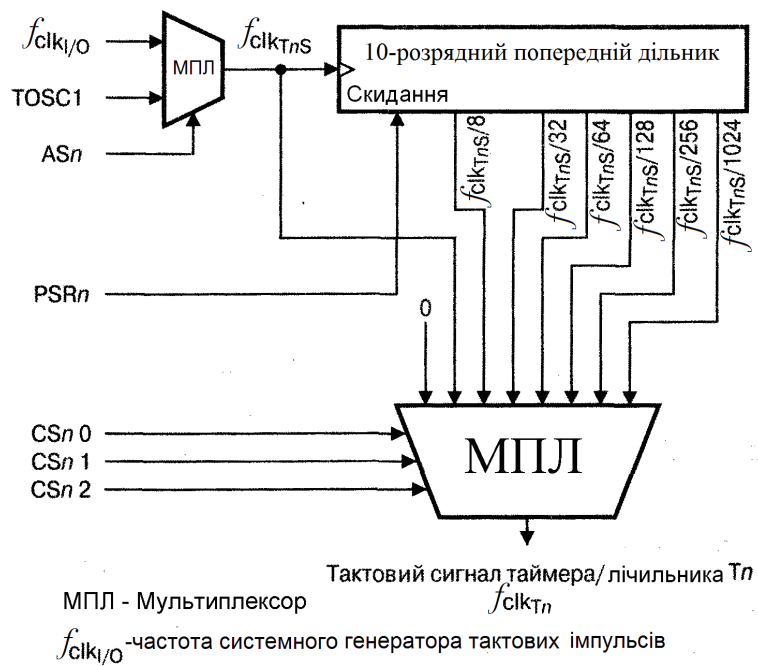


Рисунок 5.6 –Блок попереднього дільника таймера/лічильника з асинхронним режимом

5.1.1.4.1 Керування попередніми дільниками

Крім керування тактовим сигналом таймера/лічильника, всі мікроконтролери сімейства дозволяють здійснювати скидання попередніх дільників, а деякі моделі дозволяють також здійснювати їхню зупинку. Для цього використовується регістр спеціальних функцій SFIOR а для останніх моделей – регістр GTCCR [2]. Формат регістра SFIOR для деяких моделей мікроконтролерів наведений на рисунку 5.7 (розряди, що не використовуються для керування попередніми дільниками таймерів/лічильників, зазначені на рисунку як «X»).

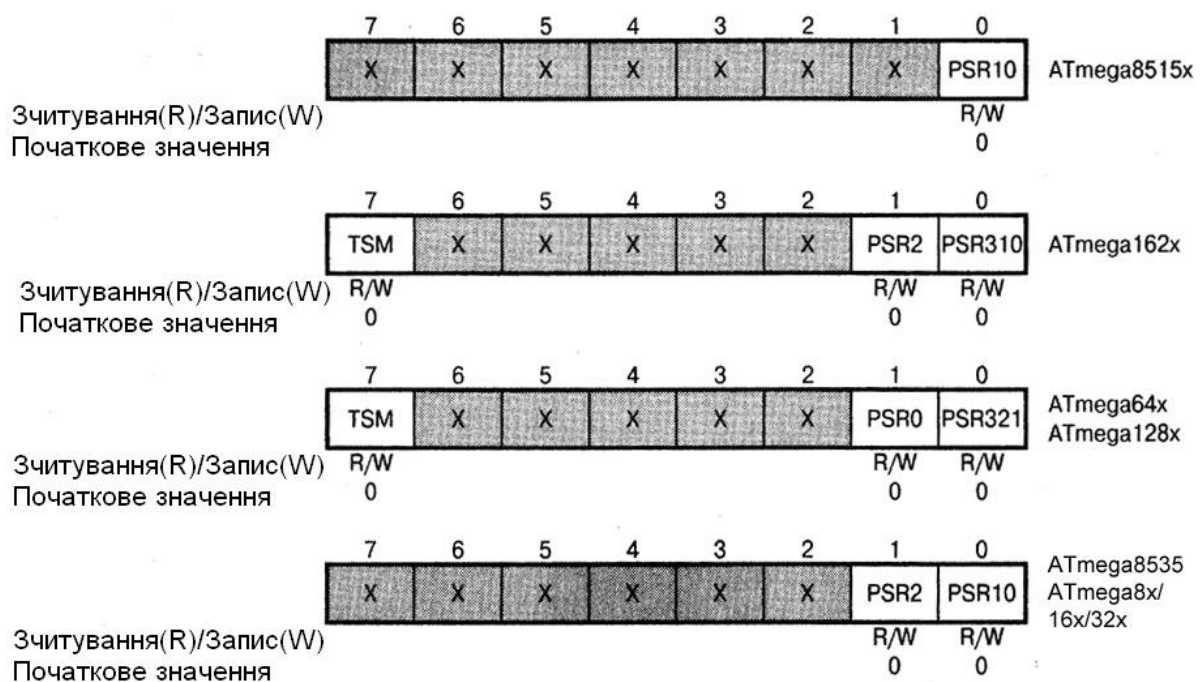


Рисунок 5.7 – Регістр SFIOR

Для скидання попередніх дільників таймерів/лічильників використовуються розряди PSRx регістра, де x – номер таймера. При записі в ці розряди логічної «1» попередні дільники відповідних таймерів/лічильників переводяться у початковий стан. Розряди скидаються

в «0» апаратно після виконання операції скидання. Ще раз нагадаємо, що один попередній дільник може використовуватися декількома таймерами/лічильниками й, відповідно, скидання попереднього дільника вплине на всі таймери/лічильники, що використовують його. Зупинка всіх попередніх дільників мікроконтролера здійснюється записом лог. 1 у розряд TSM регістра SFIOR або GTCCR. Наступний запуск попередніх дільників здійснюється записом у розряд TSM лог. 0. Зазначена функція може використовуватися, зокрема, для синхронізації таймерів/лічильників. Після встановлення розряду TSM і необхідних розрядів PSRx, відповідні таймери/лічильники зупиняються й можуть бути ініціалізовані необхідними значеннями. Після скидання розряду TSM всі таймери/лічильники почнуть працювати одночасно.

5.1.1.4.2 Використання зовнішнього тактового сигналу

Таймери/лічильники, у яких в якості тактового сигналу може використатися зовнішній сигнал, зазначені в таблиці 5.2. Відразу відзначимо, що це таймери/лічильники, що не мають асинхронного режиму роботи.

Зовнішній сигнал, що надходить на вхід Tn ($n = 0...5$) мікроконтролера, перш ніж надійти на вхід селектора тактового сигналу, проходить через спеціальний вузол, що має схему синхронізації й детектор фронтів [2]. Синхронізація зовнішнього сигналу здійснюється із частотою тактового генератора мікроконтролера (стан виводу Tn зчитується за наростаючим фронтом тактового сигналу $f_{clkI/0}$). Тому частота зовнішнього сигналу повинна бути в 2 рази меншою за частоту тактового сигналу мікроконтролера ($f_{ext} < f_{clkI/0/2}$). Однак, щоб гарантувати виявлення фронтів зовнішнього сигналу у всьому діапазоні можливих змін частоти й шпаруватості тактового сигналу мікроконтролера (через розкид параметрів

елементів тактового генератора), частота зовнішнього сигналу повинна бути ще менше — $f_{clkI/O}/2.5$. Також варто розуміти, що вхідний каскад вносить затримку між зміною стану виводу й оновленням лічильного регістра таймера/лічильника. Величина затримки становить 2.5...3.5 такти [2].

5.2 ВОСЬМИРОЗРЯДНІ ТАЙМЕРИ/ЛІЧИЛЬНИКИ

Восьмирозрядний таймер/лічильник T0 присутній у всіх моделях мікроконтролерів сімейства Mega, а таймер/лічильник T2 – в усіх, крім ATmega8515x. Усього в мікроконтролерах сімейства реалізовано три виконання восьмирозрядних таймерів/лічильників, що відрізняються набором функцій, які використовуються. Приклади структурних схем всіх трьох варіантів наведено на рисунку 5.8, а функції, які вони можуть виконувати, перераховано в таблицях 5.8 і 5.9. До складу таймерів/лічильників першого виконання (рисунок 5.8, а) входять два регістри введення/виведення: лічильний регістр TCNTn і регістр керування TCCRn. До складу таймерів/лічильників другого виконання (рисунок 5.8, б) входять вже 3 регістри введення/виведення: лічильний регістр TCNTn, регістр керування TCCRn і регістр порівняння OCRn. У третьому виконанні (рисунок 5.8, в) додається регістр ASSR, що служить для керування модулем таймера/лічильника в асинхронному режимі. Слід зазначити, що в деяких нових моделях МК-рів сімейства Mega, які виконані у другому та третьому виконаннях (рисунок 5.8, б, в), міститься додатковий канал порівняння із відповідними регістрами керування [2]. Адреси всіх перерахованих регістрів зазначені в таблиці 5.10.

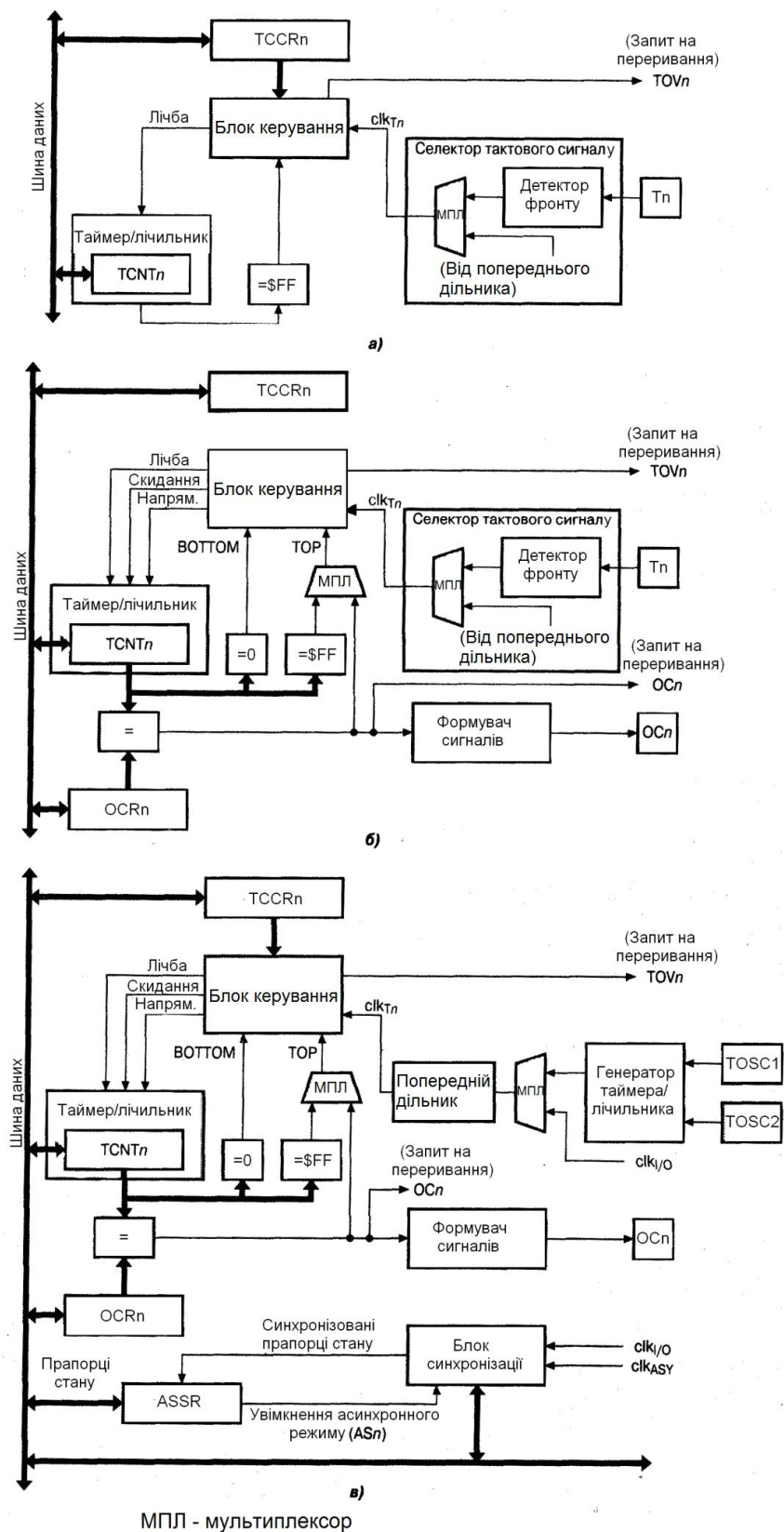


Рисунок 5.8 – Структурні схеми 8-розрядних таймерів/лічильників (T0 і T2): а — виконання 1; б — виконання 2; в — виконання 3

Таблиця 5.8 – Функції таймера/лічильника T0

Модель	ATmega8x	ATmega8515x /8535x	ATmega16x /32x	ATmega165x	ATmega162x	ATmega164x /324x/644x	ATmega48x	ATmega325x /3250x/645x/6450x	ATmega64x/128x	ATmega128x/640x/ 1280x/1281x/2560x /2561x
Варіант виконання	1	2	2	2	2	1	2	2	3	3
Восьмирозрядний лічильник	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦
Лічильник зовнішніх подій	♦	♦	♦	♦	♦	♦	♦	♦	-	♦
Широтно-імпульсний модулятор (8-розрядний)	-	♦	♦	♦	♦	-	♦	♦	♦	♦
Формувач сигналів на виводі OC0	-	♦	♦	♦	♦	-	♦	♦	♦	♦
Годинник реального часу	-	-	-	-	-	-	-	-	♦	-

Таблиця 5.9 – Функції таймера/лічильника T2

Модель	ATmega8x	ATmega8515x /8535x	ATmega16x /32x	ATmega165x	ATmega162x	ATmega164x /324x/644x	ATmega48x	ATmega325x /3250x/645x/6450x	ATmega64x/128x	ATmega128x/640x/ 1280x/1281x/2560x /2561x
Варіант виконання	3	3	3	3	3	3	3	3	2	3
Восьмирозрядний лічильник	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦
Лічильник зовнішніх подій	-	-	-	-	-	-	-	-	♦	-
Широтно-імпульсний модулятор (8-розрядний)	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦
Формувач сигналів на виводі OC2	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦
Годинник реального часу	♦	♦	♦	♦	♦	♦	♦	♦	-	♦

Таблиця 5.10 – Регістри 8-розрядових таймерів/лічильників

Регістр	Адрес	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/3250x, ATmega645x/6450x	ATmega640x/1280x/1281x/2560x/2561x	Призначення
TCCR0	\$33 (\$53)	•	•	•	•	•	•	•					Регістр керування
TCCR0A	\$24 (\$44)						•		•	•	•	•	
TCCR0B	\$25 (\$45)						•		•			•	
TCNT0	\$32 (\$52)	•	•	•	•	•		•					Лічильний регістр
	\$26 (\$46)						•		•	•	•	•	
OCR0	\$31 (\$51)	•				•		•					Регістр порівняння
	\$3C (\$5C)		•		•								
OCR0A	\$27 (\$47)						•		•	•	•	•	
OCR0B	\$28 (\$48)						•		•			•	
TCCR2	\$25 (\$45)		•	•	•	•							Регістр керування
	\$27 (\$47)							•					
TCCR2A	(\$B0)						•		•	•	•	•	
TCCR2B	(\$B1)						•		•			•	
TCNT2	\$24 (\$44)		•	•	•	•							Лічильний регістр
	\$23 (\$43)							•					
	(\$B2)						•		•	•	•	•	
OCR2	\$23 (\$43)		•	•	•	•							Регістр порівняння
OCR2A	(\$B3)						•		•	•	•	•	
OCR2B	(\$B4)						•		•			•	

Продовження таблиці 5.10

Регістр	Адреса	ATmega851x	ATmega853x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/3250x, ATmega645x/6450x	ATmega640x/1280x/1281x /2560x/2561x	Призначення
ASSR	\$22 (\$42)		•	•	•								Регістр стану асинхронного режиму
	\$26 (\$46)							•					
	\$30 (\$50)					•							
	(\$B6)						•		•	•	•	•	

Кількість переривань, які можуть генерувати таймери/лічильники T0 і T2, також залежить від виконання. Таймер/лічильник T0 першого виконання може генерувати переривання тільки при переповненні лічильного регістра. У таймерах/лічильниках другого й третього виконання переривання може генеруватися також при рівності лічильного регістра й регістра порівняння. Прапорці обох переривань перебувають у регістрі TIFR, а дозвіл/заборона цих переривань здійснюється встановленням/скиданням відповідних прапорців регістра TIMSK. Лічильний регістр таймера/лічильника TCNT0 (TCNT2) входить до складу основного блоку модуля — блоку реверсивного лічильника. Залежно від режиму роботи модуля вміст лічильного регістра скидається, інкрементується або декрементується з кожним імпульсом тактового сигналу таймера/лічильника f_{clkT0} (f_{clkT2}). Незалежно від того, є тактовий сигнал чи ні, регістр доступний у будь-який момент часу, як для зчитування, так і для запису. Однак варто пам'ятати, що будь-яка операція запису в лічильний регістр блокує роботу блоку порівняння на час одного періоду тактового сигналу таймера/лічильника. Після подачі напруги

живлення в регістрі TCNT0 (TCNT2) перебуває нульове значення. При деяких змінах стану таймера/лічильника, обумовлених режимом його роботи, встановлюється прапорець TOV0 (TOV2) регістра TIFR. Дозвіл переривання здійснюється встановленням в «1» розряду TOIE0 (TOIE2) регістра TIMSK (зрозуміло, прапорець I регістра SREG також повинен бути встановлений в «1»). Регістр порівняння OCR0 (OCR2) входить до складу блоку порівняння модуля таймера. Під час роботи таймера/лічильника виконується безперервне (у кожному машинному циклі) порівняння цього регістра з регістром TCNT0 (TCNT2). У випадку рівності вмісту цих регістрів у наступному машинному циклі встановлюється прапорець OCF0 (OCF2, OCFnA, OCFnB) регістра TIFR і генерується переривання (якщо воно дозволено). Крім того, при появі цієї події може змінюватися стан виводу OC0 (OC2, OCnA, OCnB) мікроконтролера. Щоб таймер/лічильник міг керувати станом цих виводів, вони повинні бути сконфігуровані як вихідні (відповідний розряд регістра DDRx повинен бути встановлений в «1»).

Варто звернути увагу на те, що кожна операція запису у лічильний регістр блокує формування сигналу про збіг, якщо воно відбудеться в наступному такті.

Регістри TCCRn (TCCRnA/TCCRnB) призначені для керування модулем таймера/лічильника. Формат регістрів TCCR0 та TCCR2 наведено на рисунку 5.9, а опис їхніх розрядів – у таблиці 5.11. Опис регістрів TCCRnA та TCCRnB, які не суттєво відрізняються від регістрів TCCRn, наведено у [2].

	7	6	5	4	3	2	1	0	
	–	–	–	–	–	CS02	CS01	CS00	ATmega8x ATmega163x
Зчитування(R)/Запис(W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	ATmega8515x ATmega16x ATmega162x ATmega32x ATmega64x/128x
Зчитування(R)/Запис(W)	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	

	7	6	5	4	3	2	1	0	
	FOC0	PWM0	COM01	COM00	CTC0	CS02	CS01	CS00	ATmega161x ATmega323x
Зчитування(R)/Запис(W)	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	

а)

	7	6	5	4	3	2	1	0	
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	ATmega8x ATmega16x ATmega162x ATmega32x ATmega64x/128x
Зчитування(R)/Запис(W)	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	

	7	6	5	4	3	2	1	0	
	FOC2	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	ATmega161x ATmega163x ATmega323x
Зчитування(R)/Запис(W)	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	

б)

Рисунок 5.9 – Формат регістра TCCR0(а) і TCCR2(б)

Формування тактового сигналу таймера/лічильника f_{clkT0} (f_{clkT2}) здійснюється блоком попереднього дільника, розглянутого в 5.1.1.4. В якості тактового сигналу f_{clkT0} (f_{clkT2}) таймера/лічильника T0, T2 виконань 1, 2 може використовуватися (рисунок 5.5):

- системний тактовий сигнал ($f_{\text{clkT0(T2)}} = f_{\text{clkI/O}}$);
- масштабований системний тактовий сигнал ($f_{\text{clkT0(T2)}} = f_{\text{clkI/O}}/N$, де N – коефіцієнт ділення попереднього дільника);
- зовнішній сигнал, що надходить на вхід T0 (T2) мікроконтролера ($f_{\text{clkT0(T2)}} = f_{\text{clkEXT}}$).

Таблиця 5.11 – Розряди регістра TCCR0 (TCCR2)

Розряд	Назва	Опис			
7	FOCn	Примусова зміна стану виводу OCn (режими Normal і CTC). При записі лог. 1 у цей розряд стан виводу OCn змінюється відповідно до значень розрядів COMn1:COMn0. Переривання при цьому не генерується й скидання таймера (у режимі CTC) не відбувається. У режимах Fast PWM й Phase Correct PWM цей розряд повинен бути скинутий в «0». При зчитуванні розряду завжди повертається «0».			
6,3	WGMn1: WGMn0 (CTCn, PWMn в ATmega161x, 323x)	Режим роботи таймера/лічильника. Ці розряди визначають режим роботи таймера/лічильника в такий спосіб:			
		Номер режиму	WGMn1 (CTCn)	WGMn0 (PWMn)	Режим роботи таймера/лічильника Tn
		0	0	0	Normal
		1	0	1	Phase correct PWM
		2	1	0	CTC (скидання за збігом)
		3	1	1	Fast PWM
5,4	COMn1: COMn0	Режим роботи блоку порівняння. Ці розряди визначають поведінку виводу OCn при появі події «Збіг». Вплив вмісту цих розрядів на стан виводу залежить від режиму роботи таймера/лічильника			
2...0	CSn2... CSn0	Керування тактовим сигналом. Ці розряди визначають джерело тактового сигналу для мікроконтролера. Дія цих розрядів залежить від виконання таймера/лічильника й буде описана нижче			

Примітки. 1) $n = 0$ або 2. 2) У регістрі TCCR0 моделей ATmega8x й ATmega163x задіяні тільки розряди CS02...CS00. Інші розряди регістра

зарезервовані й вважаються за «0». Таймер/лічильник T0 цих моделей працює тільки в режимі Normal.

Тактовий сигнал таймерів/лічильників T0 і T2 виконання 3 може формуватися або із системного тактового сигналу $f_{clkI/O}$ ($f_{clkT0(T2)} = f_{clkI/O}/N$), або в асинхронному режимі із сигналу від додаткового кварцового резонатора ($f_{clkT0(T2)} = f_{clkTOSC}/N$). Перехід між синхронним й асинхронним режимами роботи здійснюється за допомогою розряду AS0 (AS2) регістра ASSR (рисунок 5.6). Вибір джерела тактового сигналу, а також запуск і зупинка таймерів/лічильників здійснюються за допомогою розрядів CS02...CS00 (CS22...CS20) регістра керування таймером TCCRn0 (TCCRnA/TCCRnB) відповідно до таблиці 5.12.

Таблиця 5.12 – Вибір джерела тактового сигналу таймерів/лічильників T0 і T2

CSn2	CSn1	CSn0	Джерело тактового сигналу	
			Виконання 1, 2 (звичайний таймер/лічильник)	Виконання 3
				(асинхронний таймер/лічильник)
				ASn = «0» ASn = «1»
0	0	0	Таймер/лічильник зупинений	Таймер/лічильник зупинений

Продовження таблиці 5.12

CSn2	CSn1	CSn0	Джерело тактового сигналу		
			Виконання 1, 2 (звичайний таймер/лічильник)	Виконання 3 (асинхронний таймер/лічильник)	
				ASn = «0»	ASn = «1»
0	0	1	$f_{clkI/0}$	$f_{clkI/0}$	$f_{clkTOSCI}$
0	1	0	$f_{clkI/0}/8$	$f_{clkI/0}/8$	$f_{clkTOSCI}/8$
0	1	1	$f_{clkI/0}/64$	$f_{clkI/0}/32$	$f_{clkTOSCI}/32$
1	0	0	$f_{clkI/0}/256$	$f_{clkI/0}/64$	$f_{clkTOSCI}/64$
1	0	1	$f_{clkI/0}/1024$	$f_{clkI/0}/128$	$f_{clkTOSCI}/128$
1	1	0	Вивід Tn, лічба здійснюється за спадаючим фронтом імпульсів	$f_{clkI/0}/256$	$f_{clkTOSCI}/256$
1	1	1	Вивід Tn, лічба здійснюється за наростаючим фронтом імпульсів	$f_{clkI/0}/1024$	$f_{clkTOSCI}/1024$

Примітка. $n = 0$ або 2.

5.2.1 Режими роботи

Режим роботи таймера/лічильника T0 (T2) визначається станом розрядів WGMn1:WGMn0 (CTCn, PWMn в ATmega161x/163x) регістра TCCRn (TCCRnA/TCCRnB). Залежність режиму роботи таймерів/лічильників від стану цих розрядів показана в таблиці 5.13.

Таблиця 5.13 – Режими роботи таймерів/лічильників T0 і T2

Номер режиму	WG Mn2 ¹⁾	WG Mn1	WG Mn0	Режим роботи таймера/лічильника Tn	Модуль лічби (TOP)	Оновлення регістрів OCRnx	Момент встановлення прапорця TOVn
0	0	0	0	Normal	\$FF	Одразу	\$FF
1	0	0	1	Phase correct PWM	\$FF	При TOP	\$00
2	0	1	0	CTC (скидання за збігом)	OCRn (OCRnA)	Одразу	\$FF
3	0	1	1	Fast PWM	\$FF	При TOP	\$FF
4 ¹⁾	1	0	0	Зарезервовано	-	-	-
5 ¹⁾	1	0	1	Phase correct PWM	OCRn (OCRA)	При TOP	\$00
6 ¹⁾	1	1	0	Зарезервовано	-	-	-
7 ¹⁾	1	1	1	Fast PWM	OCRn (OCRnA)	При TOP	TOP
¹⁾ У моделях ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega640x/1280x/1281x/2560x/2561x.							

Примітка. n = 0 або 2.

5.2.2.1 Режим Normal

Це найбільш простий режим роботи таймерів/лічильників. А в таймері/лічильнику T0 моделей ATmega8x й ATmega163x це взагалі єдиний режим. У цьому режимі лічильний регістр функціонує як звичайний лічильник, що підсумовує. З кожним імпульсом тактового сигналу $fclk_{T0}$ ($fclk_{T2}$) здійснюється інкремент лічильного регістра. При переході через значення \$FF виникає переповнення, і лічба продовжується

зі значення \$00. У тому ж такті сигналу f_{clkT0} (f_{clkT2}), у якому обнуляється регістр TCNT0 (TCNT2), встановлюється в «1» прапорець переповнення TOV0 (TOV2).

У таймерах/лічильниках виконань 2 й 3 при рівності лічильного регістра й регістра порівняння встановлюється прапорець переривання OCFn (OCFnA/OCFnB) і, якщо розряд OCIE_n (OCIE_nA/OCIE_nB) регістра маскування встановлений в «1», генерується переривання. Поряд з встановленням прапорця при рівності лічильного регістра й регістра порівняння може змінюватися стан виводу OC_n (OC_nA/OC_nB) мікроконтролера. Яким чином він буде змінюватися, визначається розрядами COM_{n1}:COM_{n0} (COM_{n1}:COM_{n0}) регістра TCCR_n (TCCR_nA/TCCR_nB) відповідно до таблиці 5.14.

При необхідності стан виводу OC_n (OC_nA/OC_nB) може бути змінений примусово, записом лог. 1 у розряд FOC_n (FOC_nA/FOC_nB) регістра TCCR_n (TCCR_nA/TCCR_nB). При цьому стани виводів змінюються відповідно до таблиці 5.15. Переривання при цьому не генерується. Після збігу вмісту регістрів TCNT_n і OCR_n в режимі Normal лічба продовжується до переповнення лічильника.

Таблиця 5.14 – Керування виводом OC_n (OC_nA/OC_nB) у режимі Normal

COM _{n1} (COM _{n1})	COM _{n0} (COM _{n0})	Опис
0	0	Таймер/лічильник T _n відключений від виводу OC _n (OC _n)
0	1	Стан виводу змінюється на протилежний
1	0	Вивід скидається в «0»
1	1	Вивід встановлюється в «1»

Примітка. $n = 0$ або 2, $x = A$ або B .

5.2.2.2 Режим CTC (скидання за збігом / CHOP ON TIMER COINCIDENCE)

У цьому режимі лічильний регістр теж функціонує як звичайний лічильник, що підсумовує, інкремент якого здійснюється з кожним імпульсом тактового сигналу f_{clkT0} (f_{clkT2}). Однак максимально можливе значення лічильного регістра й, отже, роздільна здатність лічильника визначається регістром порівняння OCRn (OCRnA/OCRnB). Після досягнення значення, що записано в регістрі порівняння, наступним тактовим імпульсом лічильник скидається, і лічба продовжується зі значення “\$00”, якщо в регістрі порівняння записано “\$FF”. У тому ж такті сигналу f_{clkT0} (f_{clkT2}), у якому обнуляється лічильний регістр, встановлюється прапорець переривання за переповненням TOV0 (TOV2) регістра TIFR. Часові діаграми для цього режиму роботи таймера/лічильника наведені на рисунку 5.10. При досягненні лічильником максимального значення встановлюється прапорець OCFn (OCFnA/OCFnB), та якщо розряд OCIE_n (OCIE_nA/OCIE_nB) відповідного регістра встановлений у 1, генерується переривання. Одночасно з встановленням прапорця може змінюватися стан виводу OC_n (OC_nA/OC_nB) мікроконтролера.

Поведінка цих виводів визначається розрядами COM_{n1}:COM_{n0} (COM_{nx1}:COM_{nx0}) регістра TCCR_n (TCCR_nA/TCCR_nB), згідно таблиці 5.15.

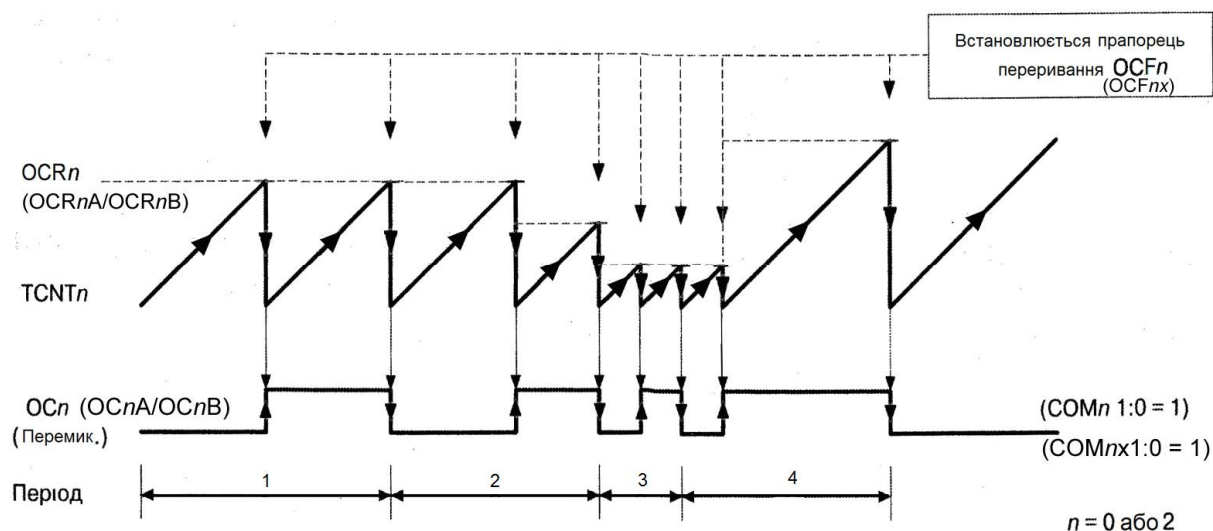


Рисунок 5.10 - Часові діаграми для режиму CTC

Таблиця 5.15 – Керування виводом OCn ($OCnA/OCnB$) у режимі CTC

$COMn1$ ($COMnx1$)	$COMn0$ ($COMnx0$)	Опис
0	0	Таймер/лічильник Tn відключений від виводу OCn
0	1	Стан виводу змінюється на протилежний
1	0	Вивід скидається в «0»
1	1	Вивід встановлюється в «1»

Примітка. $n = 0$ або 2 , $x = A$ або B .

Для генерації сигналу заданої частоти необхідно записати в розряди $COMn1:COMn0$ ($COMnx1:COMnx0$) значення «01» (зміна стану виводу на протилежний). В цьому випадку частота сигналу, що генерується, тривалість імпульсу і період сигналу, які генерується, визначаються виразами:

$$f_{OCn} = f_{clkI/O} / 2N * (1 + OCRnx),$$

$$t_{имп} = N * (1 + OCRnx) * T_{clkI/O},$$

$$T_{OCn} = t_{имп} + t_{II} = 2 * t_{имп} = 2 * N * (1 + OCRnx) * T_{clkI/O},$$

де N — коефіцієнт ділення попереднього дільника, OCR_{nx} — вміст регістра порівняння, $f_{clkI/O}$, $T_{clkI/O}$ — відповідно частота та період генератора тактових імпульсів підсистеми введення/виведення.

При необхідності стан виводу OC_n (OC_nA/OC_nB) може бути змінено примусово, записом лог. 1 у розряд FOC_n (FOC_nA/FOC_nB) регістра $TCCR_n$ ($TCCR_nA/TCCR_nB$). При цьому стан виводів змінюється відповідно до таблиці 5.15. Переривання при цьому не генерується й скидання лічильного регістра не відбувається.

5.2.2.3 Режим «Швидкодіючий ШІМ» (Fast PWM)

Режим «Швидкодіючий ШІМ» (Fast PWM) дозволяє генерувати високочастотний сигнал із широтно-імпульсною модуляцією. У зв'язку з високою частотою сигналу, що генерується, даний режим з успіхом може використовуватися в таких застосуваннях, як регулювання потужності, випрямлення, цифро-аналогове перетворення і т. ін. Лічильний регістр у цьому режимі функціонує як лічильник, що підсумовує, інкремент якого здійснюється з кожним імпульсом тактового сигналу f_{clkT0} (f_{clkT2}). Стан лічильника змінюється від $\$00$ до $\$FF$, після чого лічильний регістр скидається й цикл повторюється. При досягненні лічильником максимального значення встановлюється прапорець переривання TOV_n регістра $TIFR$. При рівності вмісту лічильного регістра й регістра порівняння OCR_n (OCR_nA/OCR_nB) встановлюється прапорець OCF_n (OCF_nA/OCF_nB) регістра $TIFR$ і змінюється стан виводу OC_n (OC_nA/OC_nB) (рисунок 5.11).

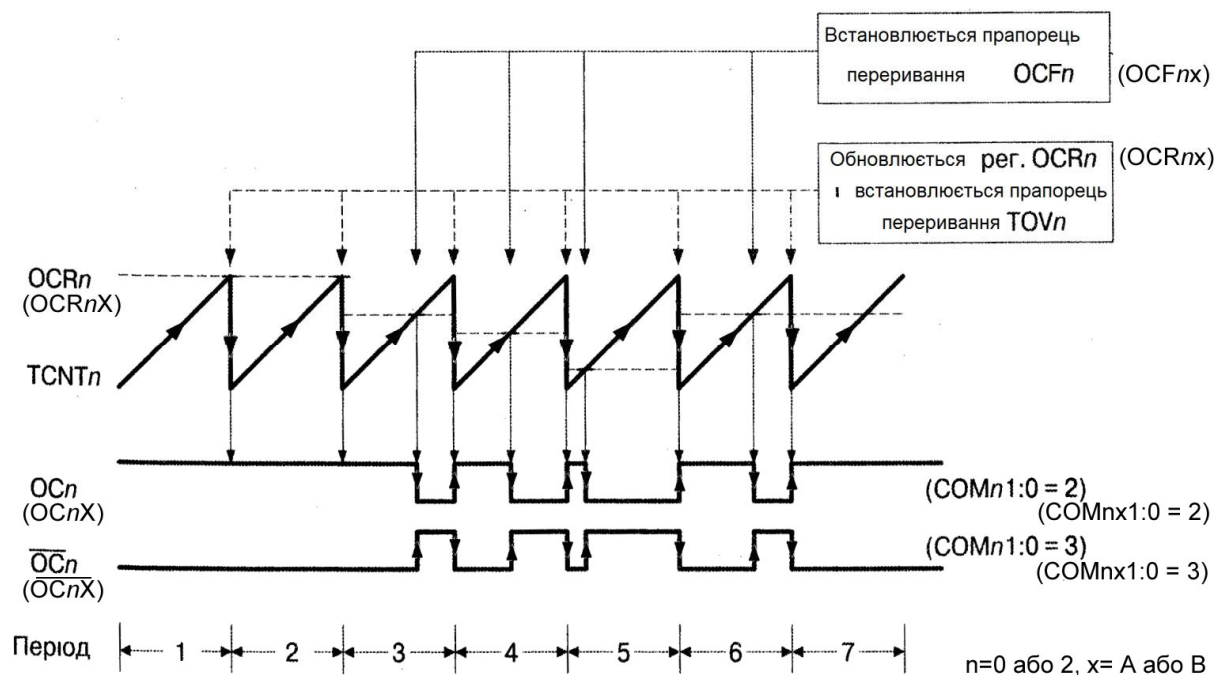


Рисунок 5.11 – Формування ШІМ-сигналу в режимі Fast PWM

Особливістю роботи схеми порівняння в цьому режимі є подвійна буферизація запису в регістр OCR_n (OCR_{nA}/OCR_{nB}), яка полягає в тому, що число, яке записується, насправді зберігається в спеціальному буферному регістрі. А зміна вмісту регістра порівняння відбувається тільки в момент досягнення лічильником максимального значення \$FF. Завдяки такому рішенню виключається поява несиметричних імпульсів сигналу (завад) на виході модулятора, які були б неминучі при безпосередньому запису в регістр порівняння. Поведінка виводу OC_n (OC_{nA}/OC_{nB}) мікроконтролера в цьому режимі також визначається вмістом розрядів COM_{n1}:COM_{n0} (COM_{nX1}:COM_{nX0}) регістра TCCR_n (TCCR_{nA}/TCCR_{nB}) (таблиця 5.16 і рисунок 5.11).

Частота сигналу, що генерується, тривалість імпульсу та період сигналу, які генерується, визначаються виразами:

$$f_{OCn} = f_{clkI/0}/256*N,$$

$$t_{имп} = T_{clkI/0} * N * (1 + OCRnx),$$

$$T_{OCn} = T_{clkI/0} * N * 256,$$

де N — коефіцієнт ділення попереднього дільника, $OCRn$ - вміст регістра порівняння, $f_{clkI/0}, T_{clkI/0}$ - відповідно частота й період генератора тактових імпульсів підсистеми введення/виведення, $TOP = \$FF$.

Окремо варто розглянути випадки, коли в регістрі порівняння перебувають гранично можливі значення ($\$00$ і $\$FF$). У першому випадку, якщо вміст регістра порівняння $OCRn$ ($OCRnA/OCRnB$) дорівнює $\$00$, на виході OCn ($OCnA/OCnB$) при кожному 256-му ($TOP = 255$) такті сигналу f_{clkT0} (f_{clkT2}) будуть спостерігатися короткі викиди. Якщо ж вміст регістра порівняння $OCRnA$ ($OCRnA/OCRnB$) дорівнює $\$FF$ ($TOP = 255$), вивід OCn ($OCnA/OCnB$) перейде в стійкий стан, обумовлений встановленими значеннями розрядів $COMn1:COMn0$ ($COMnx1:COMnx0$) (таблиця 5.16).

Таблиця 5.16 – Поведінка виводу OCn ($OCnA/OCnB$) у режимі Fast PWM

$COMn1$ ($COMnx1$)	$COMn0$ ($COMnx0$)	Поведінка виводу OCn ($OCnA/OCnB$)
0	0	Таймер/лічильник Tn відключений від виводу OCn ($OCnx$)
0	1	$OCnA$: $WGMn2 = 0$ – таймер/лічильник Tn відключений від виводу $OCnA$; $WGMn2 = 1$ – стан виводу змінюється на протилежний, якщо $TCNTn = OCRnA$. $OCnB$: зарезервовано.

Продовження таблиці 5.16

COMn1 (COMnx1)	COMn0 (COMnx0)	Поведінка виводу OCn (OCnA/OCnB)
1	0	Скидається в «0» при рівності регістрів TCNTn й OCRn. Встановлюється в «1» при TCNTn = TOP (неінвертований ШІМ-сигнал)
1	1	Встановлюється в «1» при рівності регістрів TCNTn OCRn. Скидається в «0» при TCNTn = TOP (інвертований ШІМ-сигнал)

Примітка. n = 0 або 2, x = A або B.

5.2.2.4 Режим «ШІМ з точною фазою» (Phase Correct PWM)

Режим «ШІМ з точною фазою» (Phase Correct PWM), як і режим Fast PWM, призначений для генерації сигналів із широтно-імпульсною модуляцією. Однак у цьому режимі лічильний регістр функціонує як реверсивний лічильник, зміна стану якого здійснюється з кожним імпульсом тактового сигналу f_{clkT0} (f_{clkT2}). Стан лічильника спочатку змінюється від \$00 до \$FF, а потім назад до \$00. Відповідно максимальна частота сигналу в цьому режимі у два рази менше максимальної частоти сигналу в режимі Fast PWM. Проте, завдяки «симетричності» зміни стану лічильника, режим Phase Correct PWM краще використовувати для рішення завдань керування двигунами. При досягненні лічильником максимального (TOP) значення відбувається зміна напрямку лічби, але лічильник залишається в цьому стані протягом одного періоду сигналу f_{clkT0} (f_{clkT2}). При досягненні лічильником мінімального значення (\$00) також відбувається зміна напрямку лічби й одночасно встановлюється прапорець переривання TOV0 (TOV2) регістра TIFR. При рівності вмісту лічильного регістра й регістра порівняння OCRn (OCRnA/OCRnB)

встановлюється прапорець OCF_n (OCF_nA/OCF_nB) регістра $TIFR$ і змінюється стан виводу OC_n (OC_nA/OC_nB). Яким чином змінюється стан виводу OC_n (OC_nA/OC_nB) визначається, як звичайно, вмістом розрядів $COM_{n1}:COM_{n0}$ ($COM_{nx1}:COM_{nx0}$) регістра $TCCR_n$ ($TCCR_nA/TCCR_nB$) (таблиця 5.17 і рисунок 5.12).

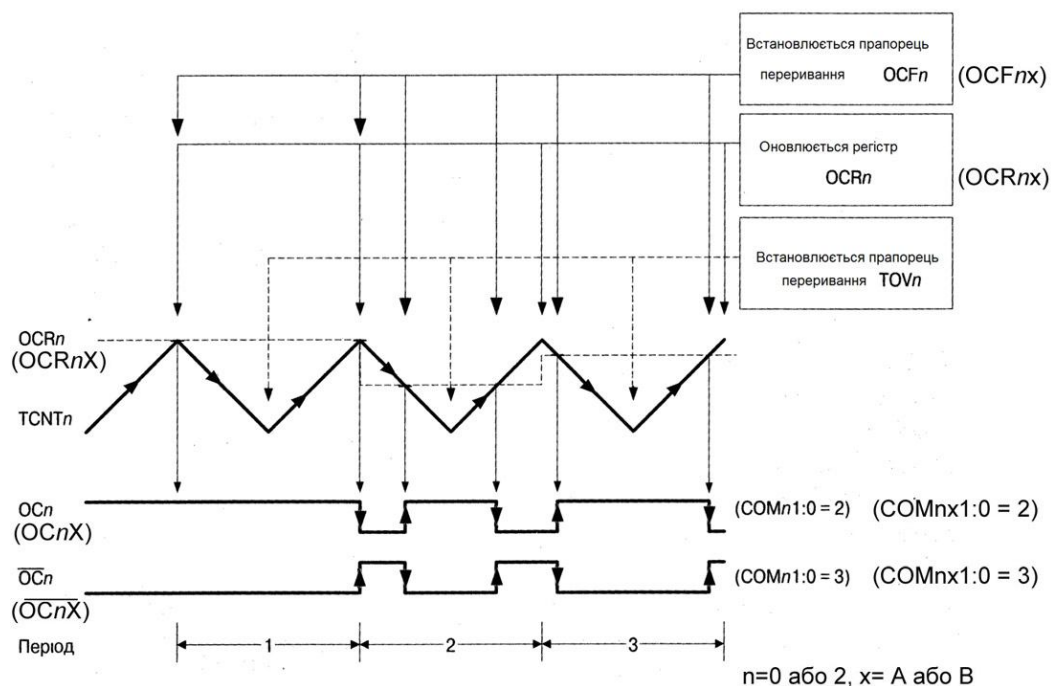


Рисунок 5.12 – Формування ШІМ-сигнала в режимі Phase Correct PWM

Таблиця 5.17 – Поведінка виводу OC_n (OC_nA/OC_nB) у режимі Phase Correct PWM

COM_{n1} (COM_{nx1})	COM_{n0} (COM_{nx0})	Поведінка виводу OC_n (OC_nA/OC_nB)
0	0	Таймер/лічильник T_n відключено від виводу OC_n (OC_nA/OC_nB)

Продовження таблиці 5.17

COMn1 (COMnx1)	COMn0 (COMnx1)	Поведінка виводу OCn (OCnA/OCnB)
0	1	OCnA: WGMn2 = 0 – таймер/лічильник Tn відключено від виводу OCnA; WGMn2 = 1 – стан виводу змінюється на протилежний, якщо TCNTn = OCRnA. OCnB: зарезервовано.
1	0	Скидається в «0» при прямій лічбі й встановлюється в «1» при зворотній лічбі (неінвертований ШІМ-сигнал)
1	1	Встановлюється в «1» при прямій лічбі й скидається в «0» при зворотній лічбі (інвертований ШІМ-сигнал)

Примітка. n = 0 або 2, x = A або B.

Для запобігання несиметричних викидів у цьому режимі теж реалізована подвійна буферизація запису в регістр OCRn (OCRnA/OCRnB). Завдяки цьому дійсна зміна вмісту регістра порівняння відбувається тільки в момент досягнення лічильником максимального значення \$FF. Якщо в регістр порівняння записати значення \$00 або максимальне значення (TOP), то при наступному збігу стану лічильника та вмісту регістра порівняння вихід схеми порівняння перемкнеться в стійкий стан відповідно до таблиці 5.18.

Частота сигналу, що генерується, тривалість імпульсу та період сигналу, які генерується, визначаються виразами

$$f_{OCnx} = f_{clk}/0/512*N,$$

$$t_{\text{imp}} = T_{\text{clkI/O}} * 2 * N * (1 + \text{OCR}_{\text{nx}}),$$

$$T_{\text{OC}_{\text{nx}}} = T_{\text{clkI/O}} * 512 * N,$$

де N — коефіцієнт ділення попереднього дільника, OCR_{nx} — вміст відповідного регістра порівняння, $f_{\text{clkI/O}}$, $T_{\text{clkI/O}}$ - відповідно частота та період генератора тактових імпульсів підсистеми введення/виведення, $\text{TOP}=255$.

Таблиця 5.18 – Стійкі стани виходу схеми порівняння

COM _{n1} (COM _{nx1})	COM _{n0} (COM _{nx0})	Регістр OCR _n (OCR _{nx})	Стан виводу OC _n (OCR _{nx})
1	0	\$00	0
1	0	Максимальне значення	1
1	1	\$00	1
1	1	Максимальне значення	0

Примітка. $n = 0$ або 2 , $x = A$ або B .

5.2.3 Асинхронний режим

У моделях ATmega64x і ATmega128x в асинхронному режимі може працювати таймер/лічильник T0. В інших моделях (крім ATmega8515x) таку можливість має таймер/лічильник T2 (рисунки 5.6 та 5.8, в). В асинхронному режимі на вхід попереднього дільника надходить сигнал від кварцового генератора таймера/лічильника, що дозволяє використовувати таймер/лічильник як годинник реального часу. Задавачем частоти сигналу може бути як кварцовий резонатор, що підключається до виводів TOSC1 та TOSC2 мікроконтролера, так і сигнал від зовнішньої схеми, що подається на вивід TOSC1. Незважаючи на те що тактовий генератор таймера/лічильника налаштований на частоту 32768 Гц, частота

кварцового резонатора або сигналу від зовнішньої схеми може лежати в межах 0...256 кГц. При цьому вона повинна бути в чотири рази менше частоти тактового сигналу мікроконтролера. Безпосередній запис у регістри TCNTn, OCRn (OCRnA/OCRnB) і TCCRn (TCCRnA/TCCRnB) в асинхронному режимі синхронізується з тактовим сигналом таймера/лічильника. При записі числа в кожний із зазначених регістрів воно зберігається в спеціальному тимчасовому регістрі, своєму для кожного регістра таймера/лічильника. А пересилання вмісту тимчасового регістра в робочий регістр таймера/лічильника здійснюються за третім після запису додатним фронтом сигналу на виводі TOSC1. Відповідно запис нового значення можна робити тільки після пересилання вмісту тимчасового регістра в регістр таймера/лічильника. Для визначення моменту дійсної зміни регістрів таймера/лічильника, а також для перемикання таймера/лічильника в асинхронний режим, призначений регістр ASSR. Наявність цього регістра в різних моделях мікроконтролерів зазначено в таблиці 5.10. Формат цього регістра наведено на рисунку 5.13, а опис окремих його розрядів наведено в таблиці 5.19.

	7	6	5	4	3	2	1	0	
	—	—	—	—	AS0	TCN0UB	OCR0UB	TCR0UB	ATmega64x/128x
Зчитування(R)/Запис(W)	R	R	R	R	R/W	R	R	R	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	—	—	—	—	AS2	TCN2UB	OCR2UB	TCR2UB	ATmega8535x
Зчитування(R)/Запис(W)	R	R	R	R	R/W	R	R	R	ATmega8x/16x/32x
Початкове значення	0	0	0	0	0	0	0	0	ATmega162x
	7	6	5	4	3	2	1	0	
	—	—	—	EXCLK	AS2	TCN2UB	OCR2UB	TCR2UB	ATmega165x
Зчитування(R)/Запис(W)	R	R	R	R/W	R/W	R	R	R	ATmega325x/3250x
Початкове значення	0	0	0	0	0	0	0	0	ATmega645x/6450x
	7	6	5	4	3	2	1	0	
	—	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	ATmega48x/88x/168x
Зчитування(R)/Запис(W)	R	R/W	R/W	R	R	R	R	R	ATmega164x/324x/644x
Початкове значення	0	0	0	0	0	0	0	0	ATmega640x/1280x/1281x
									ATmega2560x/2561x

Рисунок 5.13 – Формат регістра ASSR

Необхідно відзначити, що при перемиканні між синхронним й асинхронним режимами, вміст регістрів таймера/лічильника може бути пошкоджено. Щоб цього уникнути, рекомендується діяти в наступній послідовності:

- заборонити переривання від таймера/лічильника;
- перевести його в необхідний режим;
- записати нові значення в регістри TCNTn, OCRn (OCRnx) й TCCRn (TCCRnx);
- у випадку переходу в асинхронний режим чекати, поки прапорці TCNnUB, OCRnUB (OCR2xUB) і TCRnUB (TCR2xUB) не будуть скинуті;
- скинути прапорці переривань таймера/лічильника;
- дозволити переривання (якщо потрібно).

Таблиця 5.19 – Розряди регістра стану асинхронного режиму ASSR

Назва	Опис
ASn	Зміна режиму роботи. Якщо розряд встановлено в «1», на вхід попереднього дільника таймера/лічильника Tn надходять імпульси із кварцового генератора таймера/лічильника (асинхронний режим). У цьому режимі виводи TOSC1 й TOSC2 використовуються для підключення кварцового резонатора й відповідно не можуть використовуватися як контакти введення/виведення загального призначення. Якщо розряд скинуто в «0», на вхід попереднього дільника надходить внутрішній тактовий сигнал мікроконтролера. У цьому випадку виводи TOSC1 й TOSC2 є лініями введення/виведення загального призначення. При зміні стану цього розряду вміст регістрів TCNTn, OCRn й TCCRn може бути пошкоджений.
EXCLK	Дозвіл зовнішнього тактового сигналу. При встановленні розряду в 1 вмикається вхідний буфер зовнішнього тактового сигналу і таймер/лічильник може тактуватися зовнішнім сигналом, що подається на вхід TOSC1. Встановлення цього розряду повинно робитися перед вмиканням асинхронного режиму. Якщо в регістрі ASSR конкретного мікроконтролера цей розряд відсутній, то використання зовнішнього тактового сигналу з цим мікроконтролером не рекомендується.
TCNnUB	Стан відновлення регістра TCNTn. При записі в спеціальний тимчасовий регістр TCNTn цей прапорець встановлюється в «1», а після пересилання значення з цього регістра, прапорець апаратно скидається в «0». У такий спосіб скинутий прапорець TCNnUB означає, що спеціальний тимчасовий регістр TCNTn готовий для запису в нього нового значення. Запис у регістр TCNTn при встановленому прапорці TCNnUB може привести до пошкодження колишнього вмісту регістра й до генерації переривання.

Продовження таблиці 5.19

Назва	Опис
OCRnUB, OCR2AUB, OCR2BUB	Стан відновлення регістра OCRn (OCR2x). При записі в спеціальний тимчасовий регістр порівняння цей прапорець встановлюється в «1», а після пересилання значення, що було записано в цей регістр, прапорець апаратно скидається в «0». Таким чином, скинутий прапорець OCRnUB (OCR2xUB) означає, що спеціальний тимчасовий регістр порівняння готовий для запису в нього нового значення. Запис у регістр порівняння при встановленому прапорці OCRnUB (OCR2xUB) може привести до пошкодження колишнього вмісту регістра та до генерації переривання.
TCRnUB, TCR2AUB, TCR2BUB	Стан відновлення регістра TCCRn (TCCR2x). При записі в спеціальний тимчасовий регістр керування цей прапорець встановлюється в «1», а після пересилання значення, що було записано в цей регістр, прапорець апаратно скидається в «0». Таким чином, скинутий прапорець TCRnUB (TCR2xUB) означає, що спеціальний тимчасовий регістр керування готовий для запису в нього нового значення. Запис у регістр керування при встановленому прапорці TCRnUB (TCR2xUB) може привести до ушкодження колишнього вмісту регістра та до генерації переривання.

Примітка. n = 0 для моделей ATmega64x/128x і n = 2 для інших моделей (крім ATmega8515x), x = A або B.

При роботі таймера/лічильника в асинхронному режимі встановлення прапорців переривань від нього відбувається синхронно з тактовим сигналом мікроконтролера. Для синхронізації потрібно 3 такти плюс один період тактового сигналу таймера/лічильника. Тому до

моменту, коли мікроконтролер зможе прочитати стан лічильника, що викликав встановлення прапорця переривання, він зміниться щонайменше на одиницю. Зміна стану виводу ОСп (ОСпх) відбувається за тактовим сигналом таймера/лічильника й не синхронізується з тактовим сигналом мікроконтролера. Окремо варто сказати про «взаємодію» асинхронного режиму таймерів/лічильників з режимами зниженого енергоспоживання мікроконтролера Power Down, Power Save, Standby й Extended Standby. Перше зауваження стосується використання переривань від таймера/лічильника для «пробудження» мікроконтролера. Якщо переведення мікроконтролера в режим Power Save або Extended Standby здійснюється відразу ж після запису в регістри таймера/лічильника, необхідно переконатися, що операція запису завершена. Найбільш важливо це у випадку, коли для «пробудження» мікроконтролера використовується переривання від блоку порівняння, оскільки під час запису в лічильний регістр, або регістр порівняння робота блоку порівняння заблокована. Відповідно, якщо перехід у сплячий режим відбудеться до закінчення операції запису в зазначені регістри, переривання від схеми порівняння ніколи не відбудеться і мікроконтролер не зможе вийти зі сплячого режиму. Крім того, необхідно бути обережним при повторному переході в режим Power Save або Extended Standby після виходу з них за перериванням від таймера/лічильника. Справа в тому, що в цьому випадку для запуску підсистеми переривань потрібен проміжок часу, рівний одному періоду сигналу на виводі TOSC1. Якщо ж проміжок часу між «пробудженням» і повторним переходом у сплячий режим буде менше зазначеного, генерації переривання й відповідно переходу мікроконтролера в робочий режим не відбудеться. Для формування затримки необхідної тривалості рекомендується після «пробудження» мікроконтролера виконати запис у який-небудь із регістрів таймера/лічильника та дочекатися завершення цієї операції. Після подачі

напруги живлення, а також після «пробудження» мікроконтролера з режиму Power Down або Standby таймер/лічильник рекомендується використовувати тільки через секунду після зазначених подій [2]. Ця затримка необхідна для запуску тактового генератора таймера/лічильника. Відповідно при виході з режиму Power Down або Standby вміст всіх регістрів таймера/лічильника можна вважати загубленим (через нестабільність тактового сигналу під час запуску генератора). Причому це справедливо не тільки при використанні кварцового резонатора, але й при використанні зовнішнього тактового сигналу.

5.3 МОДУЛЯТОР

В деяких мікроконтролерах, наприклад ATmega640x/1280x/1281x/2560x/2561x з'явився новий блок, відсутній у інших моделях, – модулятор (Output Compare Modulator). З його допомогою можна формувати сигнали, з модульованою несучою частотою. Для генерації сигналу несучої частоти використовується блок порівняння А таймера/лічильника T0, а для формування модулюючого сигналу – блок порівняння С таймера/лічильника T1. Структурна схема модулятора зображена на рисунку 5.14 [2].

Виходи блоків порівняння OC1C і OC0A використовують один вивід мікроконтролера – 7-й розряд порта В. Тому модулятор вмикається автоматично при одночасному дозволі роботи цих блоків порівняння (тобто, при значенні розрядів COMnx1:0, відмінному від 0). Тип модуляції («Логічне І» або «Логічне АБО») визначається розрядом PORTB7 регістра PORTB. Спрощена схема модулятора наведена на рисунку 5.15, а часові діаграми роботи модулятора – на рисунку 5.16.

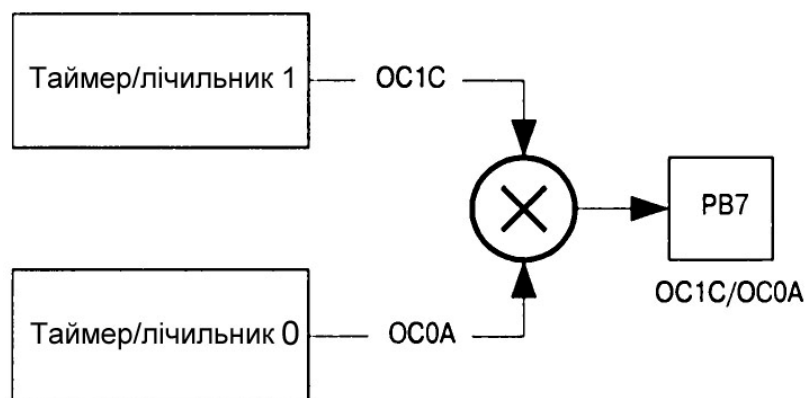


Рисунок 5.14 – Структурна схема модулятора

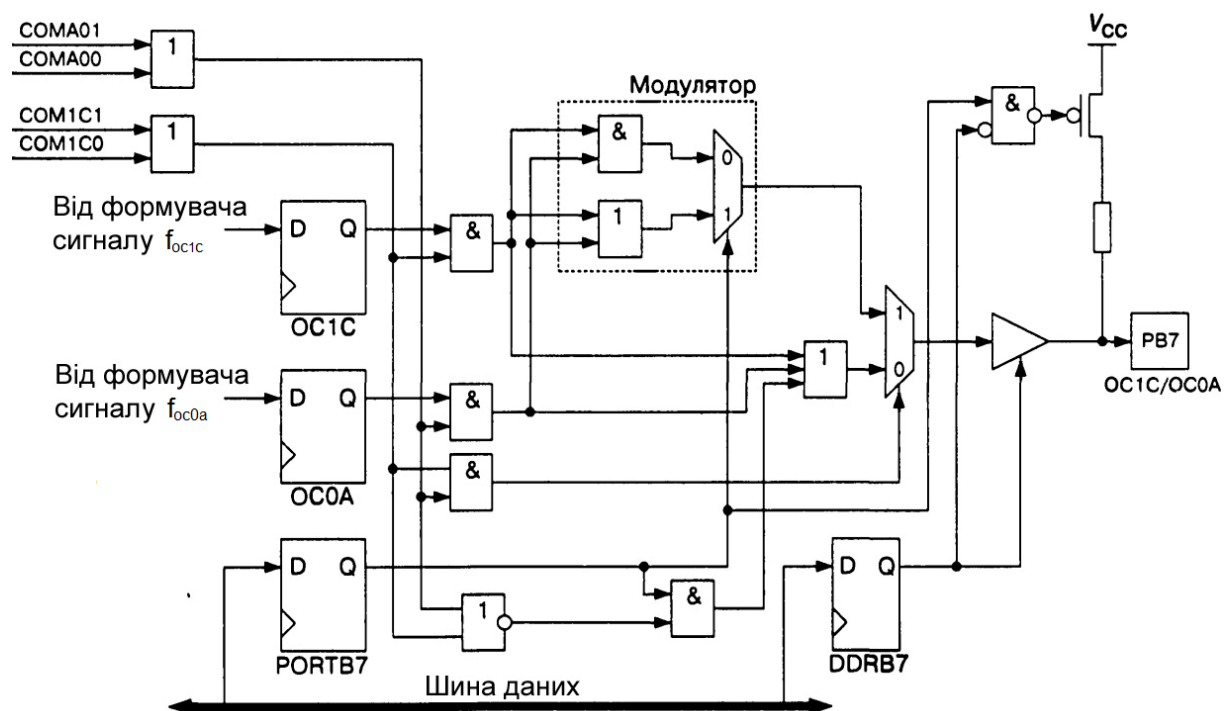


Рисунок 5.15 – Спрощена структурна схема модулятора

Зверніть увагу, що в результаті модуляції знижується роздільна здатність ШІМ-сигналу (OC1C). Коефіцієнт зменшення дорівнює кількості тактів системного тактового сигналу, що вкладаються в один період сигналу несучої частоти (OC1A). У випадку, який продемонстровано на рисунку 5.16, роздільна здатність зменшується у 2 рази. Причину такого

зниження роздільної здатності можна зрозуміти, подивившись на вихідний сигнал модулятора (2-й і 3-й періоди при $PORTB7 = 0$). Не дивлячись на те, що довжина ВИСОКОГО рівня модулюючого сигналу у 2-му періоді на один такт більше, ніж у 3-му періоді, підсумковий сигнал в обох періодах однаковий ($t_{im2} > t_{im3}$, $n_{im2} = n_{im3}$).

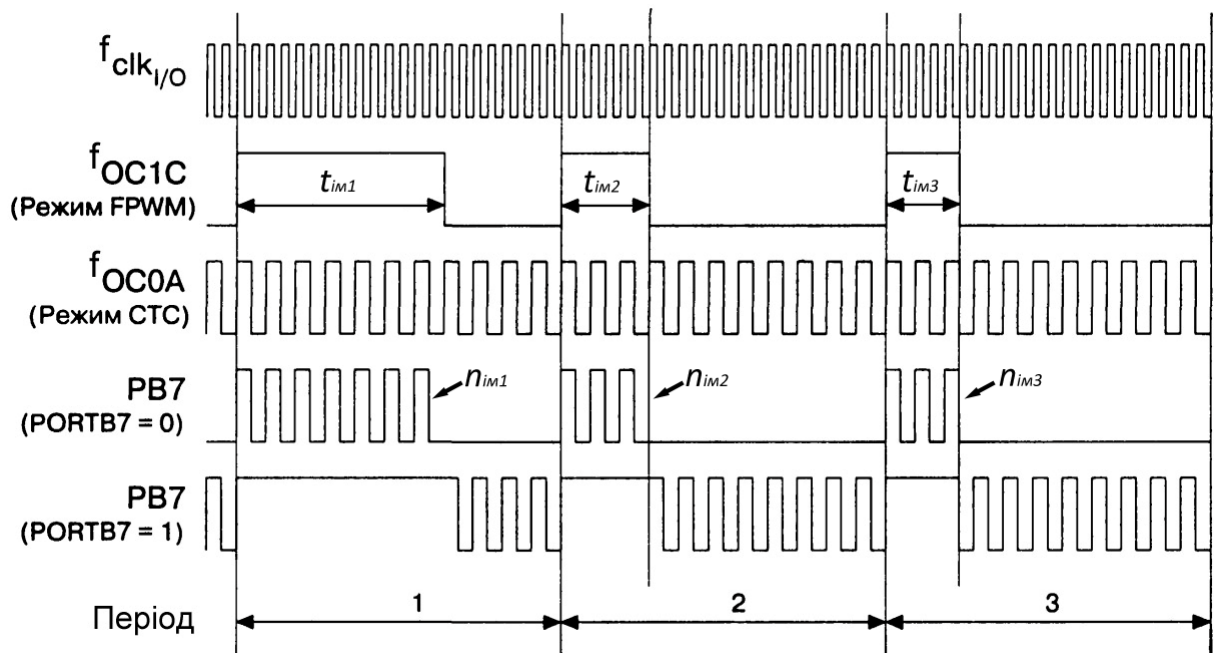


Рисунок 5.16 – Часові діаграми роботи модулятора

Роздільна здатність:

$$R = \frac{\log(TOP + 1)}{\log 2}.$$

Приклад:

$$TOP = 255$$

$$R = \frac{\log_2(256)}{\log_2 2} = \frac{8}{1} = 8.$$

5.4 16-розрядні таймери/лічильники

5.4.1 Загальна характеристика

Шістнадцятирозрядний таймер/лічильник T1 присутній у всіх моделях мікроконтролерів сімейства Mega, а таймер/лічильник T3 - тільки в моделях ATmega162x (відсутній у режимі сумісності з ATmega161x) і ATmega64x/ATmega128x [1,2]. В останніх моделях мікроконтролерів Mega присутні ще два 16-х таймери: T4 та T5. Як і таймери/лічильники T0 і T2, вони можуть використовуватися для формування часових інтервалів, для підрахунку кількості зовнішніх подій, формування сигналів і генерації ШІМ -сигналів (але вже змінної розрядності). На додачу до цього 16-розрядні таймери/лічильники можуть за зовнішнім сигналом зберігати свій поточний стан в окремому регістрі введення/виведення (режим «захоплення»). Таймери/лічильники різних моделей відрізняються тільки кількістю блоків порівняння і відповідно кількістю каналів генерації ШІМ-сигналів. Так, якщо в моделях ATmega64x/ATmega128x/ATmega640x/ATmega1280x/ATmega1281x/ATmega2560x/ATmega2561x таймери/лічильники T1, T3, T4, T5 мають по три блоки порівняння, то в інших моделях мікроконтролерів - тільки по два. Спрощена структурна схема одного з самих розвинених таймерів/лічильників (наприклад, моделі ATmega64x/ATmega128x) приведена на рисунку 5.17.

До складу кожного таймера/лічильника входять наступні регістри введення/виведення (де $n = 1, 3, 4, 5$):

- 16-розрядний лічильний регістр TCNT_n;
- 16-розрядний регістр захоплення ICR_n;
- два або три 16-розрядних регістри порівняння OCR_{nA}, OCR_{nB}, OCR_{nC};
- два або три 8-розрядних регістри керування TCCR_{nA},

TCCRnB, TCCRnC.

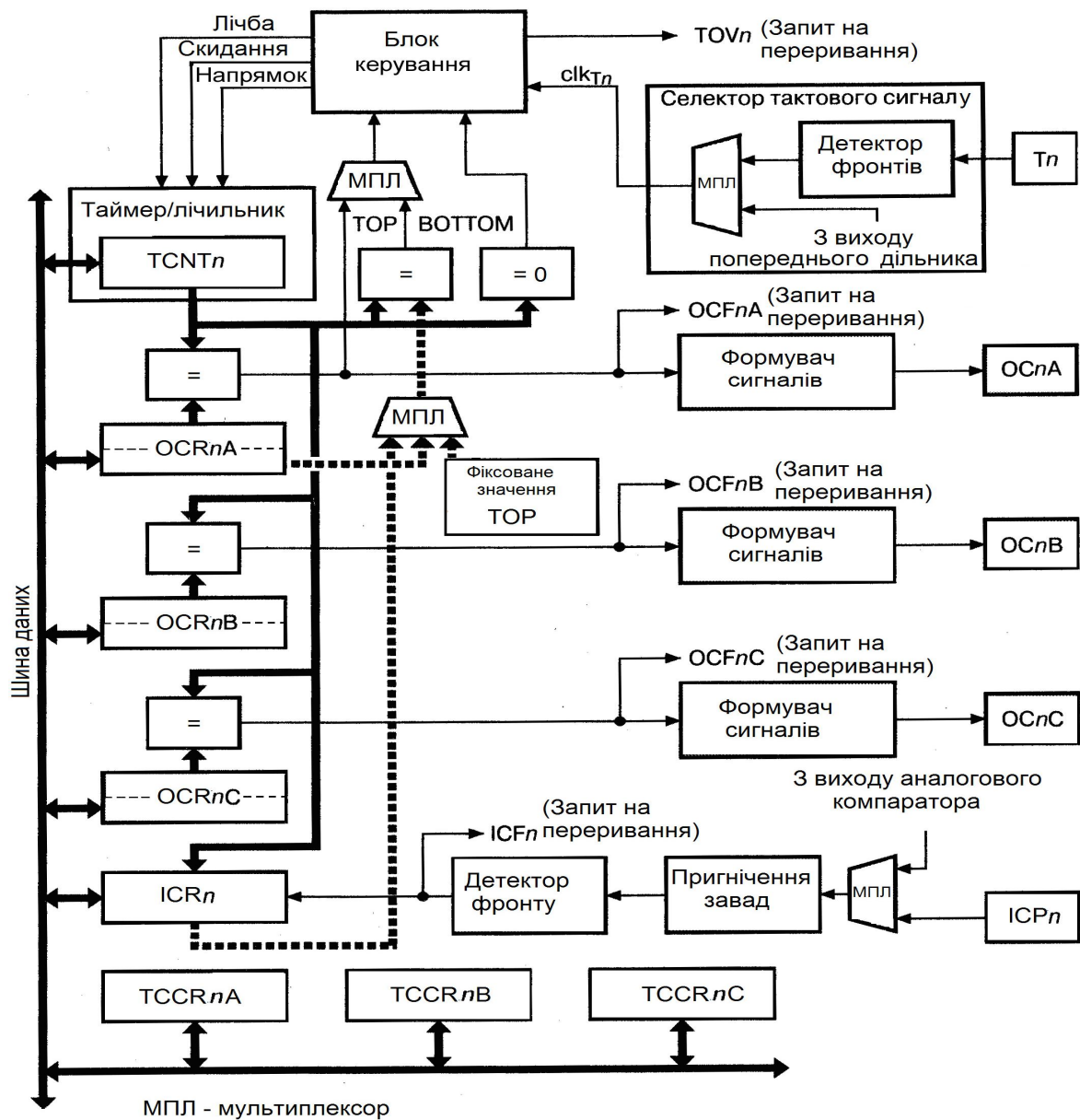


Рисунок 5.17 – Структурна схема 16-розрядних таймерів/лічильників T1, T3-T5

Кожен із 16-розрядних регістрів фізично розміщений в двох регістрах введення/виведення, назви яких утворюються додаванням до назви регістра букви "H" (старший байт) і "L" (молодший байт). Лічильний регістр таймера лічильника TCNT1, наприклад, розміщується в регістрах

TCNT1H:TCNT1L. Адреси всіх регістрів 16-розрядних таймерів/лічильників наведено в таблиці 5.20.

Таблиця 5.20 – Регістри 16-розрядних таймерів/лічильників

Регістр	Адреса	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/3250x, ATmega645x/6450x	ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x
TCCR1A	\$2F(\$4F)	•	•	•	•	•		•				
	(\$80)						•		•	•	•	•
TCCR1B	\$2E (\$4E)	•	•	•	•	•		•				
	(\$81)						•		•	•	•	•
TCCR1C	(\$7A)					•						
	(\$82)						•		•	•	•	•
TCNT1	\$2D:\$2C (\$4D:\$4C)	•	•	•	•	•		•				
	(\$85:\$84)						•		•	•	•	•
OCR1A	\$2B:\$2A (\$4B:\$4A)	•	•	•	•	•		•				
	(\$89:\$88)						•		•	•	•	•
OCR1B	\$29:\$28 (\$49:\$48)	•	•	•	•	•		•				
	(\$8B:\$8A)						•		•	•	•	•
OCR1C	(\$79:\$78)					•						
	(\$8D:\$8C)											•
ICR1	\$27:\$26 (\$47:\$46)		•	•	•	•						
	\$25:\$24 (\$45:\$44)	•						•				
	(\$87:\$86)						•		•	•	•	•
TCCR3A	(\$8B)					•		•				
	(\$90)											•
TCCR3B	(\$8A)					•		•				
	(\$91)											•

Продовження таблиці 5.20

Регістр	Адреса	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/3250x, ATmega645x/6450x	ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x
TCCR3C	(\$8C)					•						
	(\$92)											•
TCNT3	(\$89:\$88)					•		•				
	(\$95:\$94)											•
OCR3A	(\$87:\$86)					•		•				
	(\$99:\$98)											•
OCR3B	(\$85:\$84)					•		•				
	(\$9B:\$9A)											•
OCR3C	(\$83:\$82)					•						
	(\$9D:\$9C)											•
ICR3	(\$81:\$80)					•		•				
	(\$97:\$96)											•
TCCR4A	(\$A0)											•
TCCR4B	(\$A1)											•
TCCR4C	(\$A2)											•
TCNT4	(\$A5:\$A4)											•
OCR4A	(\$A9:\$A8)											•
OCR4B	(\$AB:\$AA)											•
OCR4C	(\$AD:\$AC)											•
ICR4	(\$A7:\$A6)											•
TCCR5A	(\$120)											•
TCCR5B	(\$121)											•
TCCR5C	(\$122)											•
TCNT5	(\$125:\$124)											•
OCR5A	(\$129:\$128)											•
OCR5B	(\$12B:\$12A)											•
OCR5C	(\$12D:\$12C)											•
ICR5	(\$127:\$126)											•

Таймери/лічильники T1, T3, T4, T5 можуть генерувати переривання при виникненні наступних подій:

- при переповненні лічильного регістра;
- при рівності лічильного регістра і регістра порівняння (по одному перериванню на кожен блок порівняння);
- при збереженні лічильного регістра в регістрі захоплення.

Прапорці всіх переривань 16-розрядних таймерів/лічильників знаходяться в регістрах TIFR/ETIFR/TIFRn, а дозвіл/заборона цих переривань здійснюється встановленням/скиданням відповідних прапорців регістрів TIMSK/ETIMSK/TIMSKn.

Лічильний регістр таймера/лічильника TCNTn входить до складу основного блоку модуля - блоку реверсивного лічильника. У залежності від режиму роботи модуля вміст лічильного регістра скидається, інкрементується, або декрементується з кожним імпульсом тактового сигналу таймера/лічильника f_{clk} . Незалежно від того, присутній тактовий сигнал чи ні, регістр доступний у будь-який момент часу, як для зчитування, так і для запису. При цьому будь-яка операція запису в лічильний регістр блокує роботу всіх блоків порівняння на час одного періоду тактового сигналу таймера/лічильника. Після подачі напруги живлення в регістрі TCNTn знаходиться нульове значення. При деяких змінах стану таймера/лічильника, обумовлених режимом його роботи, встановлюється прапорець TOVn відповідного регістра. Дозвіл переривання здійснюється встановленням в "1" розряду TOIEн відповідного регістра маски.

Регістри OCRnA/OCRnB/OCRnC входять до складу блоків порівняння. Під час роботи таймера/лічильника виконується безперервне (у кожному машинному циклі) порівняння цих регістрів з регістром TCNTn. У випадку рівності вмісту регістра порівняння і лічильного

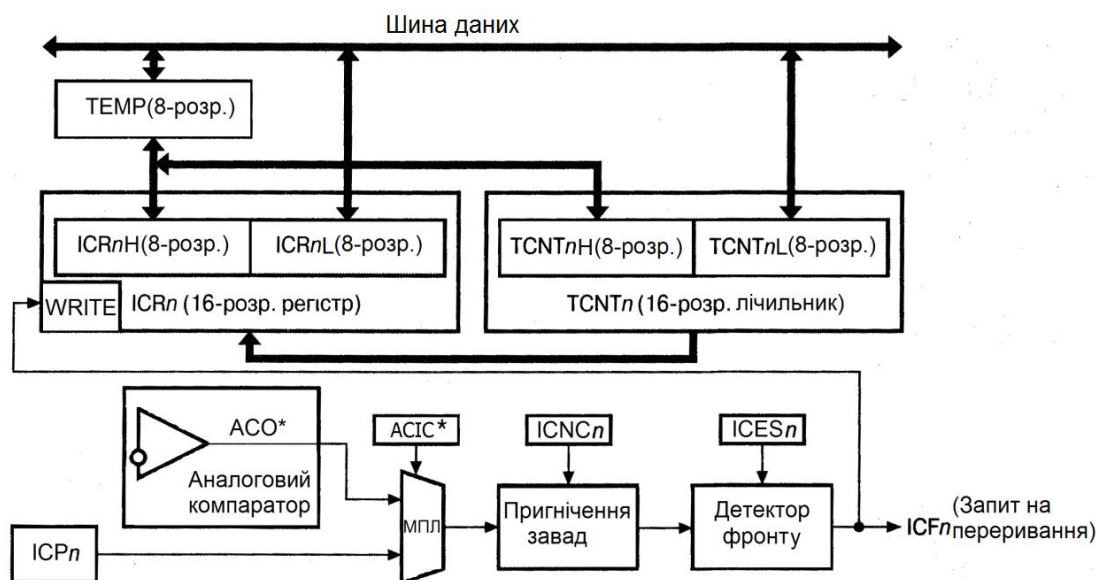
регістра в наступному машинному циклі встановлюється відповідний прапорець OCFnA/OCFnB/OCFnC відповідного регістра прапорців і генерується переривання (якщо воно дозволено). Також при виникненні цієї події може змінюватися стан виводу OCnA/OCnB/OCnC мікроконтролера. Щоб таймер/лічильник міг керувати станом якого-небудь з цих виводів, той повинен бути сконфігурований як вихідний (відповідний розряд регістра DDRx повинен бути встановлений у "1").

Особливістю роботи блоку порівняння в режимах, призначених для формування ШІМ-сигналів, є подвійна буферизація запису в регістри порівняння. Вона полягає в тому, що число, яке записується, насправді тимчасово зберігається в спеціальному буферному регістрі. А зміна вмісту регістра порівняння відбувається тільки при досягненні лічильником максимального або мінімального значення.

Регістр захоплення ICRn входить до складу блоку захоплення, призначення якого - збереження у визначений момент часу стану таймера/лічильника в регістрі захоплення ICRn. Ця дія може виконуватися або за активним фронтом сигналу на виводі ICPn мікроконтролера, або (для таймера/лічильника T1) за сигналом від аналогового компаратора. Одночасно із записом у регістр захоплення встановлюється прапорець ICFn відповідного регістра прапорців і генерується запит на переривання. Дозвіл переривання здійснюється встановленням в "1" розряду TICIEn відповідного регістра маски. Програмно запис у регістр ICRn можливо робити тільки в режимах, у яких регістр захоплення визначає модуль лічби таймера/лічильника (таблиця 5.25). Вивід ICPn у цих режимах відключений і функція захоплення, відповідно, вимкнена. Спрощена структурна схема блоку захоплення приведена на рисунку 5.18.

ТЕМР - 8-розрядний програмно недоступний регістр тимчасового зберігання старшого байта регістрів TCNTn і ICRn (більш докладно див. 5.4.2).

Для керування схемою захоплення використовуються два розряди регістра TCCRnB - ICNCn і ICESn. Розряд ICNCn керує схемою пригнічення завад. Якщо цей розряд скинутий у "0", схема пригнічення завад вимкнена і захоплення виконується за першим активним фронтом на виході мультиплексора (рисунок 5.18). Якщо ж цей розряд встановлений у "1", то з появою активного фронту виконується 4 вибірки з частотою, рівною тактовій частоті мікроконтролера. Захоплення буде виконане тільки в тому випадку, якщо усі вибірки мають рівень, що відповідає активному фронту сигналу (лог. 1 для наростаючого і лог. 0 для спадаючого).



* Вихід аналогового компаратора може впливати на таймер/лічильник Т1 (ACIC=1, див. тему «Аналоговий компаратор»).

Рисунок 5.18 – Структурна схема блоку захоплення

Активний фронт сигналу, тобто фронт, за яким буде виконане збереження вмісту лічильного регістра в регістрі захоплення, визначається станом розряду ICESn. Якщо цей розряд скинуто у "0", то активним є спадаючий фронт. Якщо розряд встановлено у "1", то активним є наростаючий фронт. Для захоплення за сигналом з виводу ICPn, цей вивід повинен бути сконфігурований як вхідний (розряд регістра DDRx, що

відповідає виводу, повинен бути скинутий у "0"). Якщо ж він буде сконфігурований як вихідний, захоплення можна буде здійснювати програмно, керуючи відповідним розрядом порту. Слід розуміти, що між зміною стану входу блоку захоплення і копіюванням лічильного регістра в регістр захоплення таймера/лічильника проходить деякий час. Цю затримку вносить каскад, що складається із синхронізатора (на рисунку не показаний) і детектора фронту. Величина затримки складає 2.5...3.5 тактів. При ввімкненні схеми пригнічення завад затримка збільшується ще на 4 такти [2]. Для керування таймером/лічильником використовуються три регістри керування: TCCRnA, TCCRnB, TCCRnC. Формат цих регістрів приведено на рисунках 5.19...5.21, а опис їхніх розрядів - у таблицях 5.21...5.23

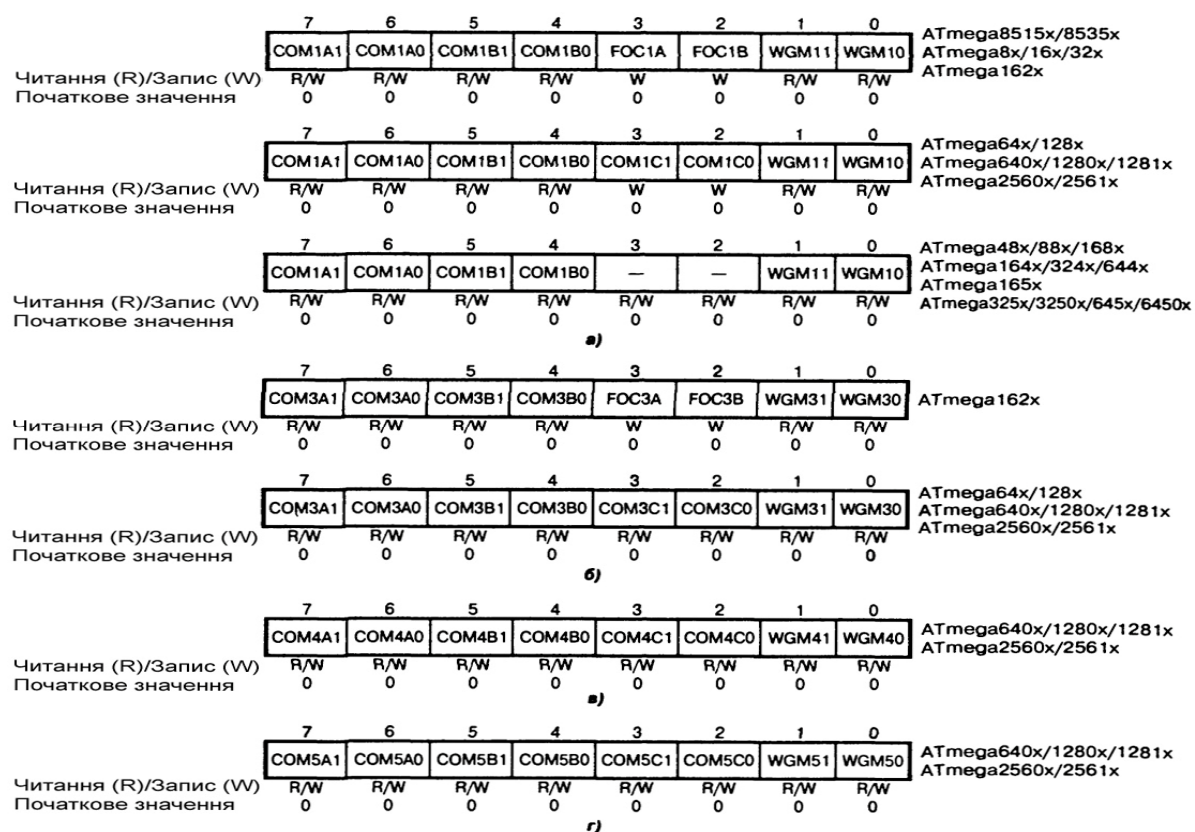


Рисунок 5.19 – Формат регістрів TCCR1A (а), TCCR3A (б), TCCR4A (в), TCCR5A (г)

Таблиця 5.21 – Розряди регістра TCCRnA

Назва	Опис
COMnA1:COMnA0	Режим роботи блоку порівняння x. Ці розряди визначають поведінку виводу OCnx при виникненні події "Збіг". Вплив вмісту цих розрядів на стан виводу залежить від режиму роботи таймера/лічильника
COMnB1:COMnB0	
COMnC1:COMnC0	
WGMn1:WGMn0	Режим роботи таймера/лічильника. Разом з розрядами WGMn3:WGMn2 регістри TCCRnB визначають режим роботи таймера/лічильника Tn (таблиця 5.25)
FOCnx	Примусова зміна стану виводу OCnx. При запису в розряд FOCnx лог. 1 стан виводу OCnx змінюється відповідно із встановленими значеннями розрядів COMn1A:COMn0A регістра TCCRnA. Переривання при цьому не генерується і скидання таймера (в режимі CTC) не виконується. Ця функція доступна лише в тих режимах, котрі не використовуються для генерації сигналу із ШІМ. При читанні розряду завжди повертається 0
FOCnx	

Примітка. $n = 1, 3, 4, 5$; $x = A, B$ або C .

	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	—	WGM13	WGM12	CS12	CS11	CS10	Все модели
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
а)									
	7	6	5	4	3	2	1	0	
	ICNC3	ICES3	—	WGM33	WGM32	CS32	CS31	CS30	ATmega64x/128x ATmega162x ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
б)									
	7	6	5	4	3	2	1	0	
	ICNC4	ICES4	—	WGM43	WGM42	CS42	CS41	CS40	ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
в)									
	7	6	5	4	3	2	1	0	
	ICNC5	ICES5	—	WGM53	WGM52	CS52	CS51	CS50	ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
г)									

Рисунок 5.20 – Формат регістрів TCCR1B (а), TCCR3B (б), TCCR4B (в), TCCR5B (г)

	7	6	5	4	3	2	1	0	
	FOC1A	FOC1B	FOC1C	—	—	—	—	—	ATmega64x/128x ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
а)									
	7	6	5	4	3	2	1	0	
	FOC1A	FOC1B	—	—	—	—	—	—	ATmega164x/324x/644x ATmega48x/88x/168x ATmega165x/325x/3250x ATmega645/6450x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
б)									
	7	6	5	4	3	2	1	0	
	FOC3A	FOC3B	FOC3C	—	—	—	—	—	ATmega64x/128x ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
в)									
	7	6	5	4	3	2	1	0	
	FOC4A	FOC4B	FOC4C	—	—	—	—	—	ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
г)									
	7	6	5	4	3	2	1	0	
	FOC5A	FOC5B	FOC5C	—	—	—	—	—	ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
д)									

Рисунок 5.21 – Формат регістрів TCCR1C (а), TCCR3C (б), TCCR4C (в), TCCR5C (г)

Таблиця 5.22 – Розряди регістра TCCRnB

Розряд	Назва	Опис
7	ICNCn	Керування схемою пригнічення завад блоку захоплення. Якщо розряд скинуто у "0", схема пригнічення завад вимкнена (захоплення виконується за першим активним фронтом). Якщо розряд встановлено у "1", схема пригнічення завад увімкнена і захоплення виконується тільки у випадку 4-х однакових вибірок, що відповідають активному фронту сигналу
6	ICESn	Вибір активного фронту сигналу захоплення. Якщо розряд ICESn скинутий у "0", збереження лічильного регістра в регістрі захоплення виконується за спадаючим фронтом сигналу. Якщо розряд встановлено у "1", збереження лічильного регістра в регістрі захоплення виконується за наростаючим фронтом сигналу. Одночасно зі збереженням лічильного регістра встановлюється також прапорець переривання ICFn регістра прапорців
5	-	Не використовується, читається як "0"
4,3	WGMn3: WGMn2	Режим роботи таймера/лічильника. Разом із розрядами WGMn1:WGMn0 регістра TCCRnA визначають режим роботи таймера/лічильника Tn
2...0	CSn2...CSn0	Керування тактовим сигналом. Ці розряди визначають джерело тактового сигналу таймерів/лічильників Tn.

Примітка. n=1,3,4,5.

Таблиця 5.23 – Розряди регістра TCCRnC

Розряд	Назва	Опис
7	FOCnA	Примусова зміна стану виводу OCnx. При запису в розряд FOCnx: лог. 1, стан виводу OCnx змінюється відповідно до встановлень розрядів OCn1x: OCn0x регістра TCCRnA. Переривання при цьому не генерується і скидання таймера (у режимі CTC) не виконується. Ця функція доступна тільки в тих режимах, що не використовуються для генерації ШІМ-сигналу. При зчитуванні розряду завжди повертається "0"
6	FOCnB	
5	FOCnC	
4...0	-	Не використовується, читається як "0"

Примітка. $n=1,3,4,5$; $x = A, B$ або C .

5.4.2 Звернення до 16-розрядних регістрів

Кожен 16-розрядний регістр таймерів/лічильників фізично розміщується в двох 8-розрядних регістрах. Відповідно для звернення до них потрібно виконати по дві операції зчитування або запису. Для того щоб запис або зчитування обох байтів вмісту 16-розрядного регістра відбувалися одночасно, у складі кожного таймера/лічильника є спеціальний 8-розрядний регістр TEMP (рисунок 5.18), призначений для зберігання значення старшого байта (цей регістр використовується тільки процесором і програмно недоступний). Для виконання циклу запису 16-розрядного регістра першим повинен бути завантажений старший байт, що записується в регістр TEMP. При наступному записі молодшого байта він з'єднується з вмістом регістра TEMP, і обидва байти одночасно (у тому самому машинному циклі) записуються в 16-розрядний регістр. Якщо потрібно змінити кілька 16-розрядних регістрів таймера/лічильника, а

старші байти всіх значень, що записуються, однакові, завантаження старшого байта достатньо виконати тільки один раз. Для виконання циклу зчитування 16-розрядного регістра першим повинен бути зчитаний молодший байт. При його зчитуванні вміст старшого байта записується в регістр TEMP. При наступному зчитуванні старшого байта повертається значення, збережене в регістрі TEMP. Виняток складають тільки регістри порівняння OCRn A/B/C, при зчитуванні яких регістр TEMP не використовується. При виконанні циклу звернення до 16-розрядного регістра таймера/лічильника переривання повинні бути заборонені. У іншому випадку, якщо переривання відбудеться між двома командами звернення до 16-розрядного регістра, а в підпрограмі обробки цього переривання теж буде зроблено звернення до одного із 16-розрядних регістрів того ж таймера/лічильника, вміст регістра TEMP буде змінено. Як наслідок, результат звернення до 16-розрядного регістра в основній програмі буде неправильним.

5.4.3 Керування тактовим сигналом

Формування тактового сигналу таймерів/лічильників f_{clkTn} ($n=1,3,4,5$) здійснюється блоком попереднього дільника, що був розглянутий у 5.1.1.4. У якості тактового сигналу f_{clkTn} таймерів/лічильників T1, T3, T4 і T5 може використовуватися:

- системний тактовий сигнал ($f_{\text{clkTn}} = f_{\text{clk I/O}}$);
- масштабований системний тактовий сигнал ($f_{\text{clkTn}} = f_{\text{clk I/O}}/N$, де N - коефіцієнт ділення попереднього дільника);
- зовнішній сигнал, що надходить на вхід Tn мікроконтролера ($f_{\text{clkTn}} = f_{\text{clk EXT}}$).

Виняток складає лише таймер/лічильник T3 моделей ATmega162x, що не може працювати від зовнішнього тактового сигналу.

Вибір джерела тактового сигналу, а також запуск і зупинка таймерів/лічильників здійснюються за допомогою розрядів CSn2...CSn0 регістра керування таймером TCCRnB відповідно до таблиці 5.24.

Таблиця 5.24 – Вибір джерела тактового сигналу таймерів/лічильників Tn

CSn2	CSn1	CSn0	Джерело тактового сигналу	
			ТЗ в моделях ATmega162x	Інші
0	0	0	Таймер/лічильник зупинений	Таймер/лічильник зупинений
0	0	1	$f_{clkI/0}$	$f_{clkI/0}$
0	1	0	$f_{clkI/0}/8$	$f_{clkI/0}/8$
0	1	1	$f_{clkI/0}/64$	$f_{clkI/0}/64$
1	0	0	$f_{clkI/0}/256$	$f_{clkI/0}/256$
1	0	1	$f_{clkI/0}/1024$	$f_{clkI/0}/1024$
1	1	0	$f_{clkI/0}/16$	Вивід Tn, лічба виконується за спадаючим фронтом
1	1	1	$f_{clkI/0}/32$	Вивід Tn, лічба виконується за наростаючим фронтом

Примітка. $n = 1, 3, 4$ або 5 .

5.4.4 Режими роботи

Режим роботи таймера/лічильника T1, T3, T4, T5 визначається станом розрядів WGMn3:WGMn2 (CTC1 у моделях ATmega161x/163x/323x) регістра TCCRnB і розрядів WGMn1:WGMn0 (PWM11:10 у моделях ATmega161x/163x/323x) регістра TCCRnA. Залежність режиму роботи таймерів/лічильників від стану цих розрядів

показано в таблиці 5.25. У моделях ATmega161x/163x/323x режими 8...15 відсутні.

5.4.4.1 Режим Normal

Це найбільш простий режим роботи таймерів/лічильників. У цьому режимі лічильний регістр функціонує як звичайний лічильник, що підсумовує. З кожним імпульсом тактового сигналу f_{clkTn} (f_{clkTn}) здійснюється інкремент лічильного регістра.

При переході через значення \$FFFF виникає переповнення, і лічба продовжується зі значення \$0000. У тому ж такті сигналу f_{clkTn} , у якому скидається в нуль регістр TCNTn, встановлюється в "1" прапорець переривання за переповненням TOVn. Блоки порівняння обох таймерів у цьому режимі можуть використовуватися як для генерації переривань, так і для формування сигналів. Поведінка виходів OCnA/OCnB/OCnC кожного з блоків порівняння таймерів/лічильників Tn визначається станом розрядів COMnx1:COMnx0) регістрів TCCRnA, як показано в таблиці 5.26.

Стан виходу будь-якого блоку порівняння також може бути змінено примусово, записом лог. 1 у розряд FOCnA/FOCnB/FOCnC регістра TCCRnC або регістра TCCRnA (рисунки 5.19, 5.21). Переривання при цьому не генерується.

Таблиця 5.25 – Режими роботи 16-розрядних таймерів/лічильників

Номер режиму	WGMn3	WGMn2	WGMn1	WGMn0	Режим роботи таймера/лічильника Tn	Модуль лічби (TOP)	Оновлення регістрів TOVn	Момент встановлення прапорця TOVn
0	0	0	0	0	Normal	\$FFFF	Негайно	\$FFFF
1	0	0	0	1	Phase correct PWM, 8-розрядний	\$00FF	При TOP	\$0000
2	0	0	1	0	Phase correct PWM, 9-розрядний	\$01FF	При TOP	\$0000
3	0	0	1	1	Phase correct PWM, 10-розрядний	\$03FF	При TOP	\$0000
4	0	1	0	0	CTC (скидання за збігом)	OCRnA	Негайно	\$FFFF
5	0	1	0	1	Fast PWM, 8-розрядний	\$00FF	При TOP	При TOP
6	0	1	1	0	Fast PWM, 9-розрядний	\$01FF	При TOP	При TOP
7	0	1	1	1	Fast PWM, 10-розрядний	\$03FF	При TOP	При TOP
8	1	0	0	0	Phase and Frequency Correct PWM	ICRn	\$0000	\$0000
9	1	0	0	1	Phase and Frequency Correct PWM	OCRnA	\$0000	\$0000
10	1	0	1	0	Phase correct PWM	ICRn	При TOP	\$0000
11	1	0	1	1	Phase correct PWM	OCRnA	При TOP	\$0000
12	1	1	0	0	CTC (скидання за збігом)	ICRn	Негайно	\$FFFF
13	1	1	0	1	Зарезервовано	–	–	–
14	1	1	1	0	Fast PWM	ICRn	При TOP	При TOP
15	1	1	1	1	Fast PWM	OCRnA	При TOP	При TOP

Примітка. $n = 1, 3, 4$ або 5 .

Таблиця 5.26 – Керування виводом OCnA/OCnB/OCnC в режимі Normal

COMnx1	COMnx0	Опис
0	0	Таймер/лічильник Tn відключено від виводу OCnx
0	1	Стан виводу змінюється на протилежний
1	0	Вивід скидається в «0»
1	1	Вивід встановлюється в «1»

Примітка. $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

5.4.4.2 Режим «Скидання за збігом» (Chop on Timer Coincidence)

У цьому режимі лічильний регістр теж функціонує як звичайний лічильник, що підсумовує, інкремент якого здійснюється кожним імпульсом тактового сигналу f_{clkTn} . Однак максимально можливе значення лічильного регістра і, отже, роздільна здатність лічильника визначається або регістром порівняння блоку "А" OCRnA, або регістром захоплення ICRn (таблиця 5.25). А після досягнення максимального значення лічба продовжується зі значення \$0000. У тому ж такті сигналу f_{clkTn} , у якому скидається в нуль лічильний регістр, встановлюється прапорець переривання TOVn. Часові діаграми для цього режиму роботи таймера/лічильника приведені на рисунку 5.22.

При досягненні лічильником максимального значення встановлюється прапорець:

- OCFnA, якщо модуль лічби визначається регістром порівняння OCRnA, режим 4;
- ICFnA, якщо модуль лічби визначається регістром захоплення ICRnA, режим 12.

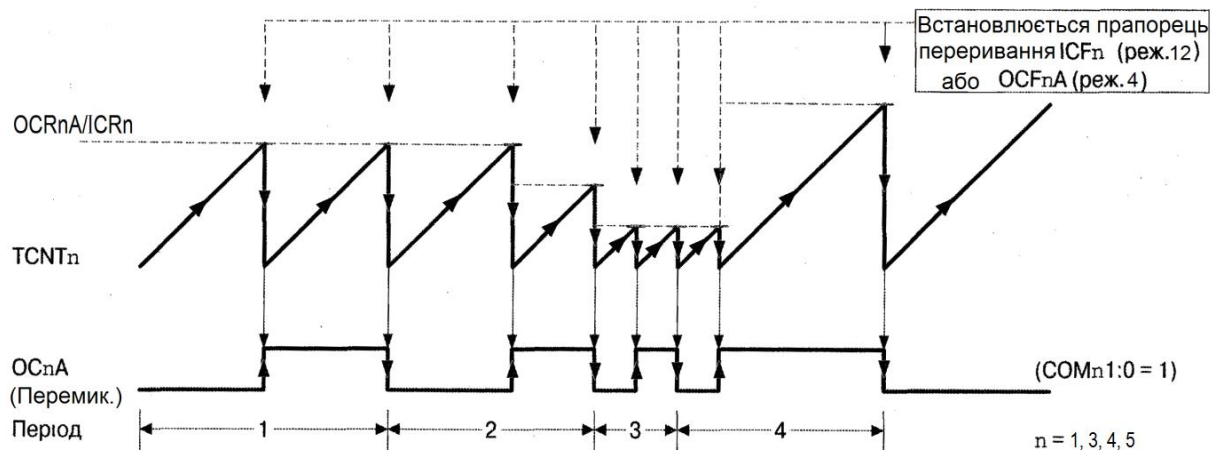


Рисунок 5.22 – Часові діаграми для режиму CTC

Одночасно з встановленням прапорця може змінюватися стан виводу OCnx мікроконтролера. Поведінка виводу визначається станом розрядів COMnx1:COMnx0 регістрів TCCRnA, як зазначено в таблиці 5.27.

Для генерації сигналу заданої частоти необхідно записати в розряди COMnx1:COMnx0 значення "01" (переключення стану виводу). Частота сигналу, що генерується, тривалість імпульсу та період сигналу, які генерується, визначаються виразами:

$$f_{OCnx} = f_{clkI/O} / 2 * N(1+Y),$$

$$t_{имп} = N * (1+Y) * T_{clkI/O},$$

$$T_{OCnx} = 2 * t_{имп} = 2 * N * (1 + Y) * T_{clkI/O},$$

де N - коефіцієнт ділення попереднього дільника, Y - модуль лічби, обумовлений значенням, що знаходиться в регістрі OCRnA або ICRnA, $f_{clkI/O}$, $T_{clkI/O}$ – відповідно частота та період генератора тактових імпульсів підсистеми введення/виведення.

Як і в режимі Normal, стан виводу OCnx при необхідності може бути змінено примусово записом лог. 1 у розряд FOCnx регістра TCCRnC або регістра TCCRnA (рисунки 5.19, 5.21). Переривання при цьому не генерується і скидання лічильного регістра не виконується.

Таблиця 5.27 – Керування виводом OCnA в режимі CTC

COMnx1	COMnx0	Опис
0	0	Таймер/лічильник Tn відключено від виводу OCnx
0	1	Стан виводу змінюється на протилежний
1	0	Вивід скидається в «0»
1	1	Вивід встановлюється в «1»

Примітка. $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

5.4.4.3 Режим «Швидкодіючий ШІМ» (Fast PWM)

Режим Fast PWM ("Швидкодіючий ШІМ") дозволяє генерувати високочастотний сигнал із широтно-імпульсною модуляцією. Цей режим практично повністю ідентичний однойменному режиму 8-розрядних таймерів/лічильників. Відмінність полягає тільки в тому, що максимальне значення, до якого буде лічити лічильник, може програмно змінюватися (таблиця 5.25).

Лічильний регістр у цьому режимі функціонує як лічильник, що підсумовує, інкремент якого здійснюється кожним імпульсом тактового сигналу f_{clkTn} . Стан лічильника змінюється від 0000 до максимального значення, після чого лічильний регістр скидається і цикл повторюється. У залежності від встановлень розрядів WGMn3:0 (CTC1, PWM11, PWM10 у моделях ATmega161x/163x/323x) максимальне значення лічильника (роздільна здатність ШІМ-сигналу) є або фіксованим значенням, або визначається вмістом окремих регістрів таймера/лічильника (таблиця 5.25). Роздільна здатність R визначається виразом:

$$R = \log(TOP + 1) / \log 2, \text{ де } TOP\text{-модуль лічби.}$$

Роздільна здатність модулятора у режимі Fast PWM приведена у

таблиці 5.28. При досягненні лічильником максимального значення встановлюється прапорець переривання TOV_n. Одночасно з ним встановлюється прапорець ICF_n (режим 14) або OCF_n (режим 15). При рівності вмісту лічильного регістра і якого-небудь регістра порівняння встановлюється відповідний прапорець переривання OCF_{nA}/OCF_{nB}/OCF_{nC}. Одночасно змінюється стан виходу блоку порівняння OC_{nA}/OC_{nB}/OC_{nC}. Поведінка цих виходів визначається вмістом розрядів COM_{n1}:COM_{n0} регістрів TCCR_{nA} (таблиця 5.29). Часові діаграми для випадку, коли модуль лічби визначається вмістом регістра OCR_{nA}, показані на рисунку 5.23.

Таблиця 5.28 – Роздільна здатність модулятора у режимі Fast PWM

Номер режиму	WGM _{n3}	WGM _{n2} (CTC1)*	WGM _{n1} (PWM11)*	WGM _{n0} (PWM10)*	Роздільна здатність	Модуль лічби (TOP)
5	0	1	0	1	8 розрядів	\$00FF
6	0	1	1	0	9 розрядів	\$01FF
7	0	1	1	1	10 розрядів	\$03FF
14**	1	1	1	0	змінна (2...16)	ICR _n (\$0003...\$FFFF)
15**	1	1	1	1	змінна (2...16)	OCR _{nA} (\$0003...\$FFFF)
* В моделях ATmega161х/163х/323х.						
** В моделях ATmega161х/163х/323х відсутні.						

Примітка. n = 1, 3, 4 або 5.

При роботі з яким-небудь фіксованим значенням модуля лічильника для задання модуля лічби рекомендується використовувати регістр захоплення ICR_n. При цьому регістр OCR_{nA} може бути використано для формування ШІМ-сигналу. Якщо в процесі формування ШІМ-сигналу його частота змінюється дуже часто, для задання модуля лічби лічильника рекомендується використовувати регістр порівняння OCR_{nA}. В цьому

випадку за рахунок буферизації запису в регістрі порівняння виключається можливість появи несиметричних імпульсів сигналу на виході модулятора.

Таблиця 5.29 – Поведінка виводу OCnA/OCnB/OCnC в режимі Fast PWM

COMnx1	COMnx0	Опис
0	0	Таймер/лічильник Tn відключений від виводу OCnx
0	1	WGMn3 = «0»: таймер/лічильник Tn відключено від виводу OCnA/OCnB/OCnC*; WGMn3 = «1»: стан виводу OCnA змінюється на протилежний
1	0	Скидається в "0" при рівності лічильного регістра і відповідного регістра порівняння. Встановлюється в "1" при досягненні лічильником максимального значення (неінвертований ШІМ-сигнал)
1	1	Встановлюється в "1" при рівності лічильного регістра і відповідного регістра порівняння. Скидається в "0" при досягненні лічильником максимального значення (неінвертований ШІМ-сигнал)
* Також для моделей ATmega161x/163x/323x.		

Примітка. $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

Якщо вміст регістра порівняння дорівнює модулю лічби, то вихід відповідного блоку порівняння переключиться в стійкий стан, обумовлений встановленнями розрядів COMnx1:COMnx0 (рисунок 5.23).

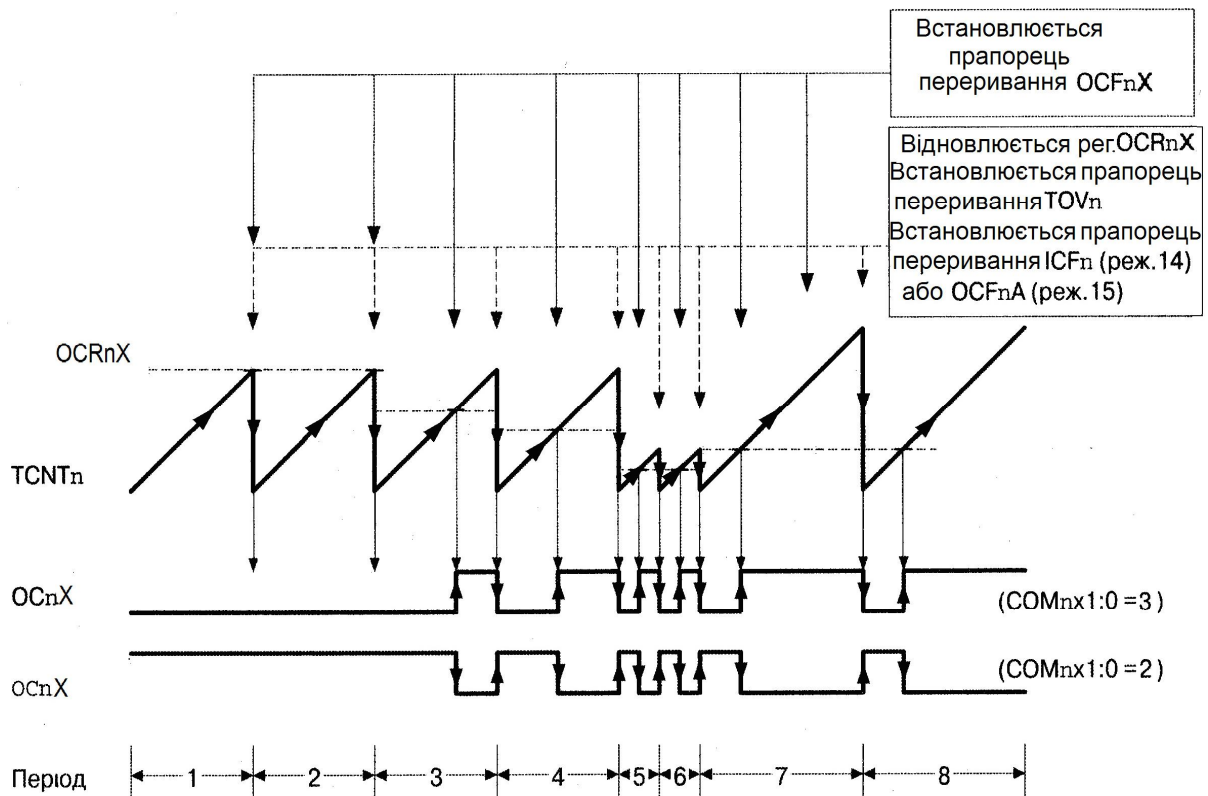


Рисунок 5.23 – Формування ШІМ-сигналу в режимі Fast PWM

Частота сигналу, що генерується, тривалість імпульсу та період сигналу, які генерується, визначаються виразами:

$$f_{OCnX} = f_{clkI/O} / N * (TOP + 1),$$

$$t_{imp} = T_{clkI/O} * N * (1 + OCRnX),$$

$$T_{OCnX} = T_{clkI/O} * N * (TOP + 1),$$

де N - коефіцієнт ділення попереднього дільника, TOP - модуль лічби, $OCRnX$ - вміст регістра $OCRnX$; $f_{clkI/O}$, $T_{clkI/O}$ - відповідно частота та період генератора тактових імпульсів підсистеми введення/виведення; $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

При необхідності блок порівняння "A" у цьому режимі може також використовуватися для генерації сигналу «Меандр». Для цього необхідно записати до розрядів $COMnA1:COMnA0$ значення "01" та встановити $WGMn3=1$, що задає переключення стану виходу $OCnA$ при виникненні

події "Збіг" (таблиця 5.10).

5.4.4.4 Режим «ШІМ з корекцією фази» (Phase Correct PWM)

Режим Phase Corect PWM («ШІМ з корекцією фази»), як і режим Fast PWM призначений для генерації сигналів із широтно-імпульсною модуляцією. Однак у цьому режимі лічильний регістр функціонує як реверсивний лічильник, стан якого спочатку змінюється від \$0000 до максимального значення, а потім назад до \$0000. Відповідно максимальна частота сигналу в цьому режимі в два рази менше максимальної частоти сигналу в режимі Fast PWM.

У залежності від встановлень розрядів WGMn3:0 (CTC1, PWM11, PWM10 у моделях ATmega161x/163x/323x) максимальне значення лічильника (роздільна здатність модулятора ШІМ-сигналу) є або фіксованим значенням, або визначається вмістом визначених регістрів таймера/лічильника (таблиця 5.30). Роздільна здатність R визначається виразом $R = \log(\text{TOP} + 1)/\log 2$, де TOP-модуль лічби.

Як і в режимі Fast PWM, при роботі з якими-небудь фіксованими значеннями модуля лічби, що змінюються інколи, для його задання рекомендується використовувати регістр захоплення. При цьому регістр OCRnA може використовуватися для формування ШІМ-сигналу. Якщо ж у процесі формування ШІМ-сигналу його частота змінюється дуже часто, для задання модуля лічби рекомендується використовувати регістр порівняння OCRnA.

При досягненні лічильником максимального значення відбувається зміна напрямку лічби, але лічильник залишається в цьому стані протягом одного періоду сигналу f_{clkTn} . У цьому ж такті відбувається відновлення вмісту регістра порівняння.

Таблиця 5.30 – Роздільна здатність модулятора в режимі Phase Correct PWM

Номер режиму	WGMn3	WGMn2 (CTC1)*	WGMn1 (PWM11)*	WGMn0 (PWM10)*	Роздільна здатність	Модуль лічби (TOP)
1	0	0	0	1	8 розрядів	\$00FF
2	0	0	1	0	9 розрядів	\$01FF
3	0	0	1	1	10 розрядів	\$03FF
10**	1	0	1	0	Змінна (2...16)	ICRn (\$0003...\$FFFF)
11**	1	0	1	1	Змінна (2...16)	OCRnA (\$0003...\$FFFF)
* В моделях ATmega161x/163x/323x. ** В моделях ATmega161x/163x/323x ці режими відсутні.						

Примітка. n = 1, 3, 4 або 5.

Якщо модуль лічби визначається регістром ICRn (режим 10) або OCRnA (режим 11), одночасно з відновленням регістра порівняння встановлюється прапорець ICFn або OCFn відповідно. При досягненні лічильником мінімального значення (\$0000) також відбувається зміна напрямку лічби й одночасно встановлюється прапорець переривання TOVn. При рівності вмісту лічильного регістра і якого-небудь регістра порівняння встановлюється відповідний прапорець OCFnA/OCFnB/OCFnC регістра TIFR. Одночасно змінюється стан виходу блоку порівняння OCNx1/OCNx0/OCNx0. Поведінка виводу визначається вмістом розрядів COMnx1:COMnx0 регістрів TCCRnA (таблиця 5.31). Часові діаграми для випадку, коли модуль лічби визначається вмістом регістра ICRn або OCRnA, показані на рисунку 5.24.

Слід розуміти, що при зміні модуля лічби під час роботи таймера/лічильника, на виході блоків порівняння можуть з'явитися імпульси, несиметричні відносно середини періоду модуляції. Оскільки відновлення вмісту регістра порівняння відбувається в момент досягнення лічильником максимального значення, період ШІМ-сигналу дорівнює часу між цими моментами. При цьому час прямої лічби визначається попереднім значенням модуля лічби, а час зворотної лічби - новим значенням. Якщо ці значення різні, то час прямої і зворотної лічби також відрізняються. Результатом цього і є несиметричні імпульси на виході блоків порівняння, як показано на рисунку 5.24 (3-й період сигналу). Тому при частій зміні модуля лічби під час роботи таймера/лічильника рекомендується використовувати режим Phase and Frequency Correct PWM, описаний у наступному підрозділі. Якщо ж використовується постійне значення модуля лічби, між цими двома режимами немає ніякої різниці.

Якщо значення, що знаходиться в регістрі порівняння, дорівнює 0000 або модулю лічби (TOP), то при наступному збігу стану лічильника і вмісту регістра порівняння вихід схеми порівняння переключиться в стійкий стан відповідно до таблиці 5.32. Частота, тривалість імпульсу та період сигналу, що генерується, визначаються виразами:

$$f_{OCnx} = f_{clkI/O} / 2 * N * (TOP + 1),$$

$$t_{imp} = T_{clkI/O} * 2 * N * (1 + OCRnx),$$

$T_{OCnx} = T_{clkI/O} * 2 * N * (TOP + 1)$, де N - коефіцієнт ділення попереднього дільника, TOP - модуль лічби, OCRnx - вміст регістра OCRnx, $f_{clkI/O}$, $T_{clkI/O}$ - відповідно частота й період генератора тактових імпульсів підсистеми введення/виведення; $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

Таблиця 5.31 – Поведінка виводу OCnA/OCnB/OCnC в режимі Phase Correct PWM

COMnx1	COMnx0	Опис
0	0	Таймер/лічильник Tn відключено від виводу OCnA/OCnB/OCnC
0	1	WGMn3 = «0»: таймер/лічильник Tn відключено від виводу OCnA/OCnB/OCnC*; WGMn3 = «1»: стан виводу OCnA змінюється на протилежний
1	0	Скидається в "0" при прямій лічбі і встановлюється в "1" при зворотній лічбі (неінвертований ШІМ-сигнал)
1	1	Встановлюється в "1" при прямій лічбі і скидається в "0" при зворотній лічбі (інвертований ШІМ-сигнал)
* Також для моделей ATmega161x/163x/323x.		

Примітка. n = 1, 3, 4 або 5; x = A, B або C.

Таблиця 5.32 – Стійкі стани виходу схеми порівняння

COMnx1	COMnx0	Регістр OCRnx	Стан виводу OCnx
1	0	\$0000	0
1	0	TOP	1
1	1	\$0000	1
1	1	TOP	0

Примітка. n = 1, 3, 4 або 5; x = A, B або C.

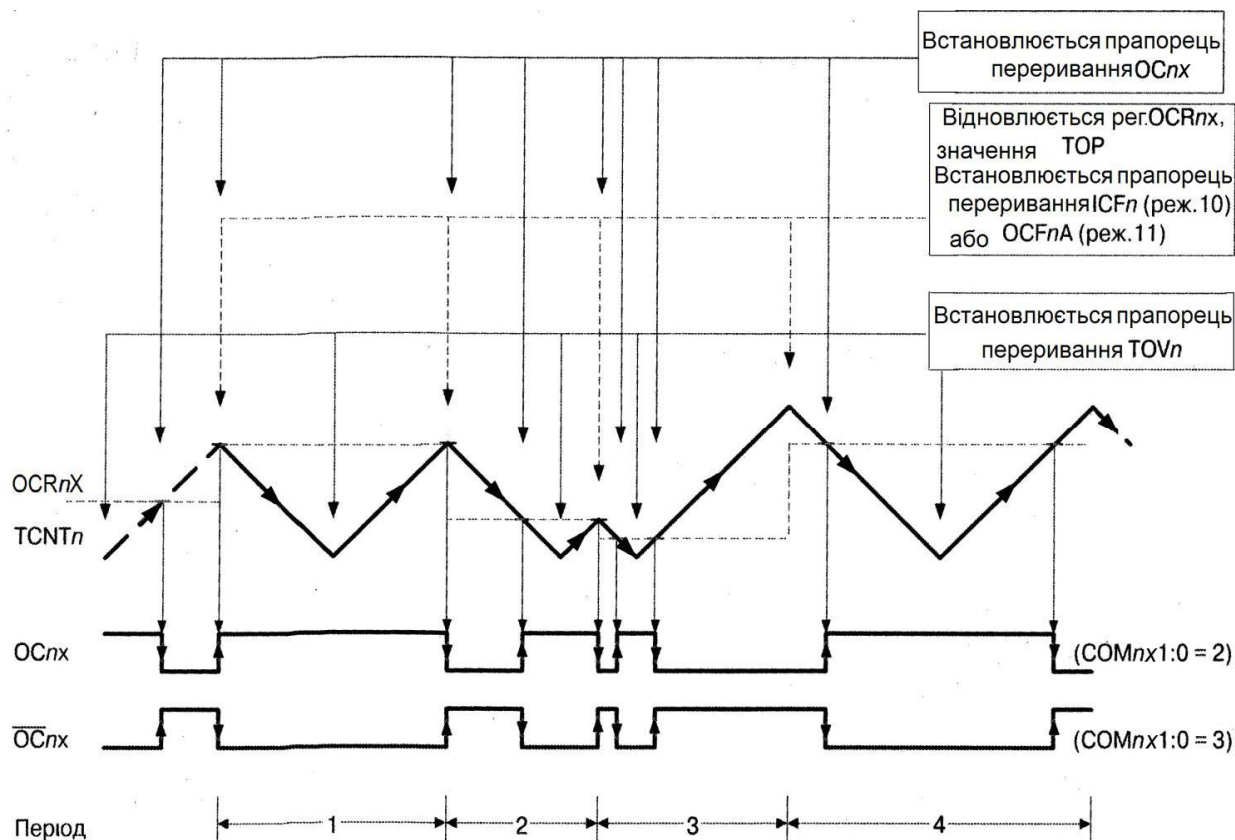


Рисунок 5.24 – Формування ШІМ-сигналу в режимі Phase Correct PWM

5.4.4.5 Режим «ШІМ з корекцією фази та частоти» (Phase and Frequency Correct PWM)

Режим Phase and Frequency Correct PWM («ШІМ з корекцією фази та частоти»), відсутній у моделях ATmega161x/163x/323x, дуже схожий на режим Phase Corect PWM. Єдина принципова різниця між ними - момент відновлення вмісту регістра порівняння (рисунок 5.25). Максимальне значення лічильника (роздільна здатність ШІМ-сигналу) у цьому режимі може визначатися тільки регістрами ICRn або OCRnA таймера/лічильника Tn, як показано в таблиці 5.25. Роздільна здатність визначається виразом $R = \log(TOP+1)/\log 2$, де TOP — модуль лічби (таблиця 5.33). Як і в інших режимах, при роботі з якими-небудь фіксованими значеннями модуля лічби, що змінюються рідко, для задання його рекомендується

використовувати регістр захоплення ICRn. При цьому регістр OCRnA може використовуватися для формування ШІМ-сигналу. Якщо ж у процесі формування ШІМ-сигналу його частота змінюється дуже часто, для модуля лічби рекомендується використовувати регістр порівняння OCRnA.

При досягненні лічильником максимального значення відбувається зміна напрямку лічби, але лічильник залишається в цьому стані протягом одного періоду сигналу f_{clkTn} . У цьому ж такті встановлюється прапорець ICFn або OCFnA (залежить від того, який з регістрів використовується для задання модуля лічби). При досягненні лічильником мінімального значення (\$0000) напрям лічби знову змінюється. При цьому встановлюється прапорець переривання TOVn відповідного регістра і відбувається відновлення вмісту регістра порівняння.

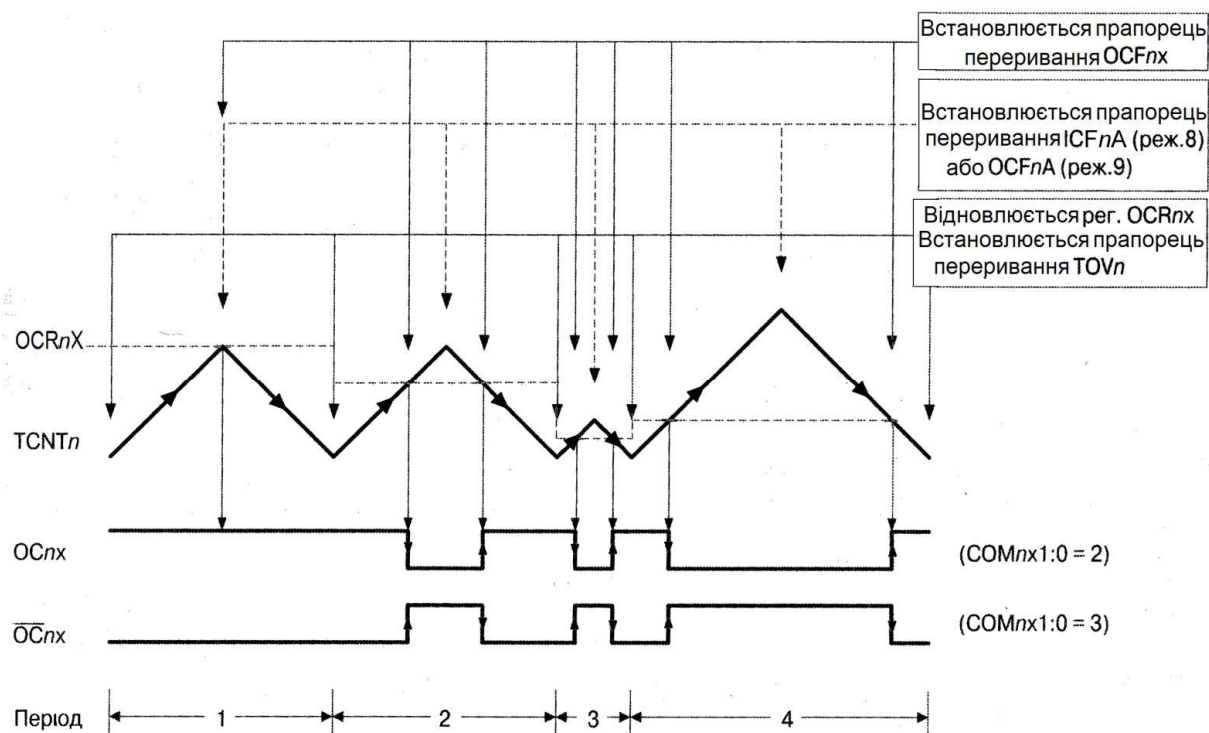


Рисунок 5.25 – Формування ШІМ-сигналу в режимі Phase and Frequency Correct PWM

Таблиця 5.33 – Роздільна здатність модулятора в режимі Phase and Frequency Correct PWM

Номер режиму	WGMn3	WGMn2	WGMn1	WGMn0	Роздільна здатність	Модуль лічби (TOP)
8	1	0	1	0	Змінна (2...16)	ICRnA (\$0003...\$FFFF)
9	1	0	1	1	Змінна (2... 16)	OCRnA (\$0003...\$FFFF)

Примітка. $n = 1, 3, 4$ або 5 .

При рівності вмісту лічильного регістра і якого-небудь регістра порівняння встановлюється відповідний прапорець OCFnA/OCFnB/OCFnC. Одночасно змінюється стан виходу блоку порівняння OCnA/OCnB/OCnC. Звичайно поведінка виводу визначається вмістом розрядів COMnx1:COMnx0 регістрів TCCRnA (таблиця 5.34). Якщо порівняти рисунок 5.24 і рисунок 5.25 можна побачити, що в режимі Phase and Frequency Correct PWM кожен період сигналу є повністю симетричним. Це є наслідком того, що відновлення вмісту регістра порівняння відбувається в момент досягнення лічильником мінімального значення. Тому час прямої лічби завжди дорівнює часу зворотної лічби, вихідні імпульси симетричні відносно періоду і, відповідно, частота сигналу, що генерується, залишається сталою.

Частота, тривалість імпульсу та період сигналу, що генерується, визначаються виразами:

$$f_{OCnx} = f_{clkI/O} / 2 * N * (TOP + 1),$$

$$t_{imp} = T_{clkI/O} * 2 * N * (1 + OCRn),$$

$T_{OCnx} = T_{clkI/O} * 2 * N * (TOP + 1)$, де N - коефіцієнт ділення попереднього дільника, TOP – модуль лічби, OCRnx – вміст регістра OCRnx, $f_{clkI/O}$, $T_{clkI/O}$ -

відповідно частота та період генератора тактових імпульсів підсистеми введення/виведення; $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

Якщо значення, що знаходиться в регістрі порівняння, дорівнює 0000 або модулю лічби (TOP), то при наступному збігу стану лічильника і вмісту регістра порівняння вихід схеми порівняння переключиться в стійкий стан відповідно до таблиці 5.32.

Таблиця 5.34 – Поведінка виводу OCnA/OCnB/OCnC в режимі Phase and Frequency Correct PWM

COMnx1	COMnx0	Опис
0	0	Таймер/лічильник Tn відключено від виводу OCnA/OCnB/OCnC
0	1	WGMn3 = «0»: таймер/лічильник Tn відключено від виводу OCnA/OCnB/OCnC; WGMn3 = «1»: стан виводу OCnA змінюється на протилежний
1	0	Скидається в "0" при прямій лічбі і встановлюється в "1" при зворотній лічбі (неінвертований ШІМ-сигнал)
1	1	Встановлюється в "1" при прямій лічбі і скидається в "0" при зворотній лічбі (інвертований ШІМ-сигнал)

Примітка. $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

5.5 ВАРТОВИЙ ТАЙМЕР

5.5.1 СТАНДАРТНИЙ ВАРТОВИЙ ТАЙМЕР

Всі AVR-мікроконтролери мають вартовий таймер, основна функція якого - захист пристрою від збоїв. Завдяки вартовому таймеру можна перервати виконання програми, що зациклилася або вийти з інших

непередбачених ситуацій, що перешкоджають нормальному виконанню програми. В мікроконтролерах сімейства можна зустріти вартовий таймер у двох виконаннях: стандартному та розширеному [2]. Структурна схема вартового таймера у стандартному режимі приведена на рисунку 5.26.

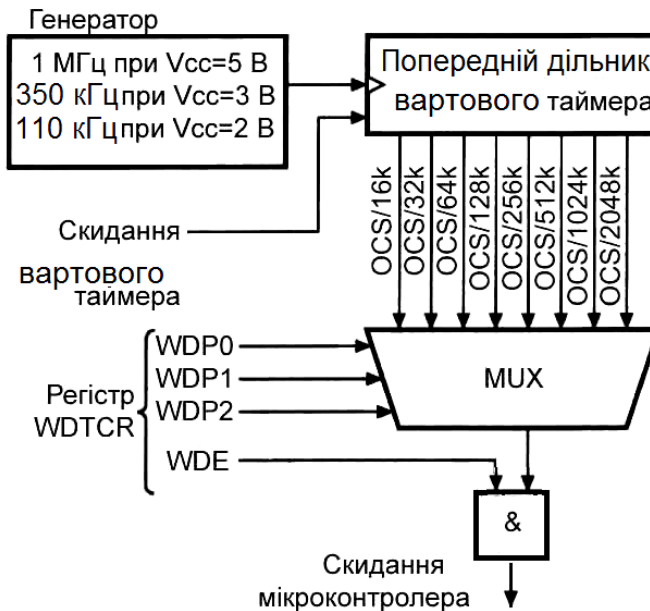


Рисунок 5.26 - Структурна схема вартового таймера

Вартовий таймер має незалежний тактовий генератор і працює навіть у режимі Power Down. Частота цього генератора залежить від напруги живлення пристрою, температури, технологічного розкиду. Типові значення частот рівні: 1 МГц при $V_{cc} = 5.0$ В, 350 кГц при $V_{cc} = 3.0$ В і 110 кГц при $V_{cc} = 2.0$ В. Якщо вартовий таймер включено, то через визначені проміжки часу (при настанні тайм-ауту) виконується скидання мікроконтролера. При нормальному виконанні програми вартовий таймер повинен періодично (через проміжки часу, менші його періоду) скидатися командою WDR. Для керування вартовим таймером у більшості моделей призначений регістр WDTCR (Watch Dog Timer Control Register), розташований за адресою \$21 (\$41). Формат цього регістра приведено на рисунку 5.27, а короткий опис розрядів цього регістра приведено в таблиці

5.35.

	7	6	5	4	3	2	1	0
\$21/\$01	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0
Читання (R)/Запис(W)	R	R	R	R/W	R/W	R/W	R/W	R/W
Початкове значення	0	0	0	0	0	0	0	0

Рисунок 5.27 – Формат регістра WDTCR

Таблиця 5.35 – Розряди регістра WDTCR

Розряд	Назва	Опис
7..5	-	Зарезервовані
4	WDTOE (WDCE*)	Дозвіл вимкнення вартового таймера
3	WDE	Дозвіл вартового таймера ("1" - включений)
2..0	WDP2..WDP0	Коефіцієнт ділення попереднього дільника вартового таймера

*В деяких мікроконтролерах це біт дозволу зміни конфігурації вартового таймера

Для ввімкнення/вимкнення вартового таймера використовуються два розряди регістра WDTCR - WDE і WDTOE. Якщо розряд WDE (Watch Dog Enable) встановлений у "1", вартовий таймер включений, якщо розряд WDE скинутий у "0", вартовий таймер вимкнений. Безпосередньо перед включенням таймера рекомендується також виконувати його скидання командою WDR. Щоб уникнути ненавмисного вимикання вартового таймера, використовується розряд WDTOE (Watch Dog Timer On Enable). Справа в тім, що вимкнення вартового таймера (скидання розряду WDE) можна здійснити тільки при встановленому розряді WDTOE, який апаратно скидається через 4 машинних цикли після установки в "1". За рахунок цього практично виключається можливість випадкового

вимикання вартового таймера. Виходячи зі сказаного, для вимикання вартового таймера необхідно одною командою записати лог. 1 у розряди WDE і WDTOE та протягом наступних чотирьох машинних циклів записати лог. 0 у розряд WDE. Період настання тайм-ауту вартового таймера задається за допомогою розрядів WDP2...WDPO регістра WDTCSR відповідно до таблиці 5.36.

Таблиця 5.36 – Задання періоду вартового таймера

WDP2	WDP1	WDP0	Число тактів генератора	Період настання тайм-ауту (типове значення)	
				V _{cc} = 3.0В	V _{cc} = 5.0В
0	0	0	16К	17.3 мс	16.3 мс
0	0	1	32К	34.3 мс	32.5 мс
0	1	0	64К	68.5 мс	65 мс
0	1	1	128К	0.14 с	0.13 с
1	0	0	256К	0.27 с	0.26 с
1	0	1	512К	0.55 с	0.52 с
1	1	0	1024К	1.1 с	1.0 с
1	1	1	2048К	2.2 с	2.1 с

В моделях ATmega8515x/8535x/8x/64x/128x/162x/165x/325x/3250x/645x/6450x передбачено декілька так званих рівнів безпеки, кожен із яких накладає деякі обмеження на зміну конфігурації вартового таймера. Для вибору конкретного рівня майже у всіх вказаних моделях використовуються дві конфігураційні комірки (у моделях ATmega8x, ATmega165x, ATmega325x/3250x/645x/6450x - одна). Перша комірка, загальна для всіх моделей, має назву WDTON. Другою коміркою, що визначає рівень безпеки, є комірка, котра переводить мікроконтролер у режим сумісності з іншими моделями. Відповідно, в моделях ATmega8515x це комірка S8515C, ATmega8535x – S8535C, ATmega162x – M161C, ATmega64x/128x – M103C. Відповідність між цими комірками і конфігурацією вартового таймера відображена у таблиці 5.37.

Режим керування у моделях, що розглядаються, характеризується трьома рівнями: 0, 1 і 2.

Рівень 0. В цьому режимі керування вартовим таймером відбувається так само, як і в більш ранніх моделях. При ввімкненні мікроконтролера вартовий таймер вимкнено, однак він може бути увімкнений у будь-який момент записом лог.1 у розряд WDE регістра WDTCSR.

Рівень 1. Для вимкнення вартового таймера або для зміни періоду тайм-ауту необхідно виконати наступні дії:

1. Однією командою записати лог.1 у розряди WDE і WDTCE.
2. Протягом наступних чотирьох тактів записати (однією командою) необхідні значення в розряди WDE і WDP2:0, одночасно скидаючи розряд WDCE.

Рівень 2. В цьому режимі вартовий таймер увімкнено постійно (розряд WDE завжди читається як 1) і не може бути вимкнений. Для зміни періоду тайм-ауту необхідно виконати наступні дії:

1. Однією командою записати лог.1 у розряди WDE і WDTCE.
2. Протягом наступних чотирьох тактів записати необхідні значення в розряди WDP2:0, одночасно скидаючи розряд WDCE. Значення, яке записується у розряд WDE є несуттєвим.

Таблиця 5.37 – Конфігурація вартового таймера

S8515C (ATmega8515x), S8535C (ATmega8535x), M161C (ATmega162x), M103C (ATmega64x/128x)	WDTON	Рівень	Початковий стан вартового таймера	Вимкнення вартового таймера	Зміна періоду тайм-аута
1	1	1	Вимкнений	Послідовність команд	Послідовність команд
1	0	2	Увімкнений	Завжди увімкнений	Послідовність команд
0	1	0 ¹⁾	Вимкнений	Послідовність команд	Без обмежень
0	0	2 ¹⁾	Увімкнений	Завжди увімкнений	Послідовність команд
¹⁾ Відсутній у моделях ATmega8x, ATmega165x і ATmega325x/3250x/645x/6450x					

Період тайм-ауту вартового таймера задається за допомогою розрядів WDP2...WDP0 регістра WDTCSR згідно таблиці 5.36.

Як і раніше, перед записом розрядів WDP2:0 рекомендується або заборонити вартовий таймер, або скинути його.

5.5.2 РОЗШИРЕНИЙ ВАРТОВИЙ ТАЙМЕР

У моделях ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega640x/1280x/1281x/2560x/2561x реалізовано розширений вартовий таймер, головною відмінністю якого від звичайного таймера є можливість

роботи у декількох режимах (рисунок 5.28). Частота генератора розширеного вартового таймера була зменшена до 128 кГц, завдяки чому максимальний період тайм-ауту збільшився до 8 с.

Для керування розширеним вартовим таймером використовується регістр керування WDTCSR, розташований за адресою (\$60). Формат цього регістра наведено на рисунку 5.29, а опис розрядів – у таблиці 5.38.

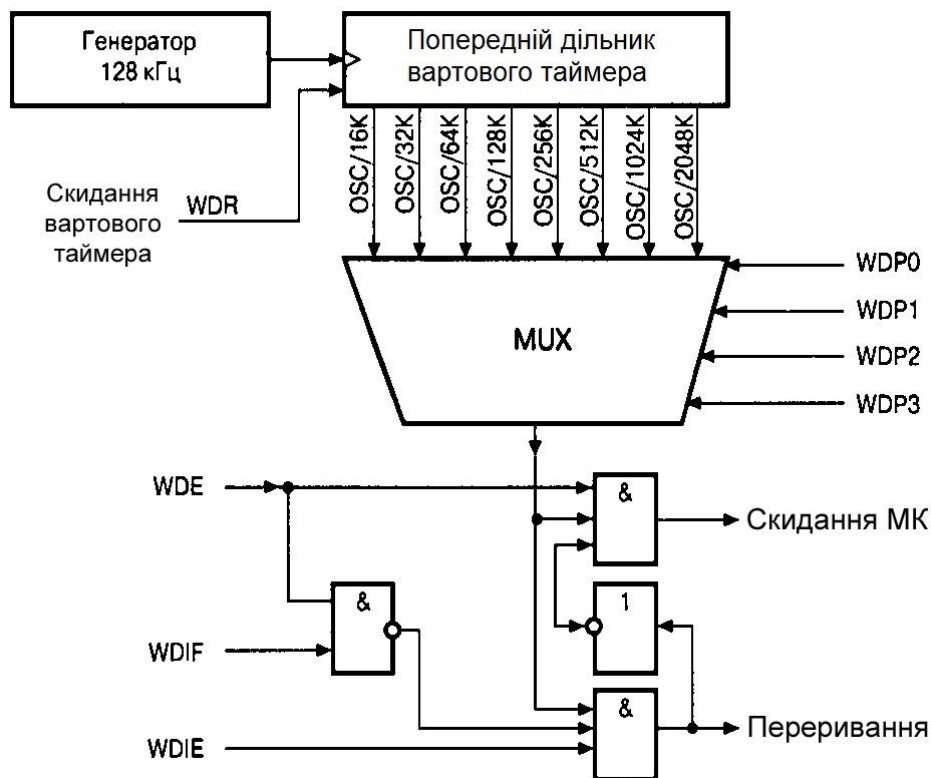


Рисунок 5.28 – Структурна схема розширеного вартового таймера

Для увімкнення/вимкнення вартового таймера використовується 2 розряди регістра WDTCSR – WDE і WDCE. Якщо розряд WDE встановлено в 1, вартовий таймер увімкнено, якщо в 0 – вимкнено. Для скидання розряду WDE необхідно виконати наступні дії:

1. Однією командою записати лог.1 у розряди WDE і WDTCE.
2. Протягом наступних чотирьох тактів записати лог. 0 у розряд WDE.

Зверніть увагу на те, що стан прапорця скидання WDRF регістра

MCUSR відмінює стан розряду WDE. Це означає, що розряд WDE встановлено завжди, коли встановлено прапорець WDRF, тому перед скиданням WDE необхідно також скинути WDRF.

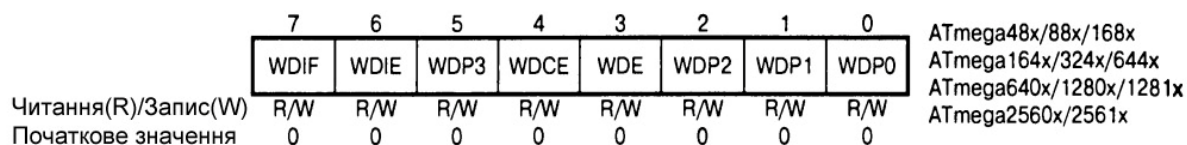


Рисунок 5.29 – Формат регістра WDTCSR

Таблиця 5.38 – Біти регістра WDTCSR

Біт	Назва	Характеристика
7	WDIF	Прапорець переривання від вартового таймера
6	WDIE	Дозвіл переривання від вартового таймера
5	WDP3	Коефіцієнт ділення попереднього дільника вартового таймера (спільне з розрядами WDP2...WDP0)
4	WDCE	Дозвіл зміни конфігурації вартового таймера
3	WDE	Дозвіл скидання за тайм-аутом вартового таймера
2	WDP2	Коефіцієнт ділення попереднього дільника вартового таймера
1	WDP1	
0	WDP0	

Як вже зазначалося, розширений вартовий таймер може працювати в декількох режимах (таблиця 5.39). В режимі скидання він працює як стандартний вартовий таймер, викликаючи скидання мікроконтролера через задані проміжки часу. Як правило, цей режим використовується для попередження зависання програми через збій в роботі мікроконтролера.

Таблиця 5.39 – Режими роботи вартового таймера

WDTON	WDE	WDIE	Режим	Дія за тайм-аутом
1	0	0	Зупинений	Немає дій
1	0	1	Режим переривання	Переривання
1	1	0	Режим скидання	Скидання
1	1	1	Режим переривання і скидання	Переривання, потім переключення в режим скидання
0	X	X	Режим скидання	Скидання

Більш детально робота розширеного таймера розглянута в [2].

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Для чого може використовуватись таймер/лічильник?
- 2) Які регістри відповідають за дозвіл/заборону переривань від таймерів/лічильників?
- 3) Яким чином задається режим роботи таймера/лічильника T0 (T2)?
- 4) Чим відрізняється режим роботи CTC від режиму Normal?
- 5) Для рішення яких завдань краще використання режиму Phase Correct PWM ніж режиму Fast PWM?
- 6) Які таймери/лічильники можуть працювати в асинхронному режимі?
- 7) Що може виступати у якості задавача частоти при роботі таймера/лічильника в асинхронному режимі?
- 8) Для чого призначений модулятор?
- 9) Для чого можуть використовуватись 16-розрядні таймери/лічильники?
- 10) При виникненні яких подій таймери/лічильники T1, T3, T4, T5 можуть генерувати переривання?
- 11) За допомогою яких регістрів виконується керування таймером/лічильником?
- 12) Які сигнали можуть використовуватись у якості тактового сигналу f_{clk} таймерів/лічильників T1, T3, T4 і T5?
- 13) Яким чином здійснюється вибір джерела тактового сигналу, а також запуск і зупинка таймерів/лічильників?
- 14) Як увімкнути режим роботи Normal таймера/лічильника T3? Для чого може використовуватись даний режим роботи?
- 15) Що таке роздільна здатність лічильника таймера?
- 16) У чому полягає різниця між роботою 8-розрядних та 16-розрядних таймерів/лічильників у режимі Fast PWM?

- 17) Завдяки чому в режимі Phase and Frequency Correct PWM кожен період сигналу є повністю симетричним?
- 18) У чому полягає функція вартового таймера?
- 19) Чим відрізняється розширений вартовий таймер від звичайного?

ЛІТЕРАТУРА [1...10]

6 СИСТЕМА ПЕРЕРИВАНЬ

6.1 СИСТЕМА ПЕРЕРИВАНЬ AVR-МІКРОКОНТРОЛЕРІВ

6.1.1 Загальні відомості

Переривання припиняє нормальний хід програми для виконання пріоритетної задачі, обумовленої внутрішньою або зовнішньою подією мікроконтролера. При виникненні переривання мікроконтролер зберігає у стеку адресу наступної команди (вміст лічильника команд PC) і завантажує в нього адресу відповідного вектора переривання. За цією адресою повинна знаходитися команда відносного переходу до підпрограми обробки переривання. Крім того, останньою командою підпрограми обробки переривання повинна бути команда RETI, що забезпечує повернення в основну програму і відновлення стану попередньо збереженого лічильника команд.

Оскільки основними джерелами переривань є різні периферійні пристрої мікроконтролерів, кількість переривань залежить від конкретної моделі.

6.1.2 Таблиця векторів переривань

AVR-мікроконтролери сімейства Mega мають багаторівневу систему пріоритетних переривань [1,2]. Молодші адреси пам'яті програм, починаючи з адреси \$0001 (моделі ATmega8515x/8535x/8x/48x/88x) або \$0002 (інші моделі), відведені під таблицю векторів переривання. Кожному перериванню відповідає своя адреса в цій таблиці і саме ця адреса завантажується в лічильник команд при виникненні переривання. Положення вектора в таблиці визначає також і пріоритет відповідного

переривання: чим менше адреса, тим вище пріоритет переривання. Розмір вектора переривання залежить від обсягу пам'яті програм мікроконтролера і становить 1 слово для моделей з об'ємом пам'яті менше 16 Кбайт і 2 слова для інших моделей. Відповідно, для переходу до підпрограм обробки переривань у моделях ATmega8515x/8535x, ATmega8x і ATmega48x/88x використовуються команди RJMP, а в інших моделях - команди JMP.

Практично в усіх мікроконтролерах сімейства Mega, за винятком моделі ATmega48x, положення таблиці векторів переривань може бути змінено. Таблиця може розташовуватися не тільки на початку пам'яті програм, але також і на початку області завантажувача, причому переміщення таблиці може бути здійснено безпосередньо в ході виконання програми.

Залежно від моделі (таблиця 6.1) для керування розміщенням таблиці переривань використовується або регістр керування мікроконтролера MCUCR, розташований за адресою \$35 (\$55), або загальний регістр керування перериваннями GICR, розташований за адресою \$3B (\$5B).

Таблиця 6.1 – Регістри для керування розміщенням таблиці переривань

Назва	Опис	Адреса	Модель
MCUCR	Регістр керування мікроконтролера	\$35 (\$55)	ATmega64x/128x, ATmega88x/168x, ATmega164x/324x/644x, ATmega165x/325x/3250x/645x/6450x, ATmega640x/1280x/1281x/2560x/2561x
GICR	Загальний регістр керування перериваннями	\$3B (\$5B)	ATmega8515x/8535x, ATmega8x/16x/32x, ATmega162x

Для керування таблицею переривань в цих регістрах використовуються два молодших розряди: IVSEL (1-й розряд) і IVCE (0-й

розряд). Стан прапорця IVSEL визначає положення таблиці в пам'яті програм. Якщо прапорець скинутий в 0, то таблиця векторів переривань розташовується на початку пам'яті програм, якщо встановлений у 1 - на початку області завантажувача [1,2]. Конкретне значення початкової адреси області завантажувача залежить від значень конфігураційних комірок BOOTSZ1 і BOOTSZ0 (крім моделей ATmega48x). Розряд IVCE призначений для дозволу зміни прапорця IVSEL.

Для зміни положення таблиці векторів переривань необхідно виконати наступні дії:

1. Встановити розряд IVCE в 1.
2. Протягом наступних чотирьох тактів занести необхідне значення в розряд IVSEL, при цьому розряд IVCE скидається в 0. В іншому випадку розряд IVCE буде скинутий апаратно після закінчення чотирьох тактів, забороняючи подальшу зміну прапорця IVSEL.

На час виконання описаної послідовності переривання автоматично забороняються та дозволяються тільки після скидання прапорця IVCE. Стан прапорця I регістра SREG при цьому не змінюється.

Розмір таблиці векторів переривань залежить від моделі мікроконтролера і становить від 16 (моделі ATmega8515x) до 56 (моделі ATmega640x/1280x/2560x) векторів. Розподіл адрес таблиці векторів переривань для різних мікроконтролерів сімейства наведено в [2]. При розміщенні векторів переривань в області завантажувача до значень, наведених у таблицях, слід додати значення початкової адреси області завантажувача.

Якщо переривання в роботі мікроконтролера не передбачаються, то на місці таблиці векторів переривань може бути розміщена частина основної програми. Далі наведено фрагменти лістингів для різних положень векторів скидання і переривань (на прикладі моделей ATmega128x, розмір області завантажувача - 8 Кбайт).

Конфігураційна комірка BOOTRST визначає положення вектора скидання: 0 – вектор в області завантажувача; 1 – за адресою \$0000.

1. Вектор скидання і таблиця векторів переривань розташовуються на початку пам'яті програм (BOOTRST = 1, IVSEL = 0)

Address	Labels	Code	Comments
\$0000		jmp RESET	; Оброблювач скидання
\$0002		jmp EXT_INT0	; Оброблювач IRQ0
\$0004		jmp EXT_INT1	; Оброблювач IRQ1
...		...	...
\$0044		jmp SPM_RDY	; Оброблювач переривання ; "готовність SPM"
\$0046	RESET:	ldi r16, high (RAMEND)	; Початок основної програми
\$0047		out SPH, r16	
\$0048		ldi r16, low (RAMEND)	
\$0049		out SPL, r16	; Проініціалізували ; покажчик стека
\$004A		Sei	; Дозволили переривання
\$004B		<команда> xxx	
...		...	

2. Вектор скидання розташовується на початку пам'яті програм, а таблиця векторів переривань - на початку області завантажувача (BOOTRST = 1, IVSEL = 1, BOOTSZ1 = BOOTSZ0 = 0)

Address	Labels	Code	Comments
\$0000	RESET:	ldi r16, high (RAMEND)	; Початок основної програми
\$0001		out SPH, r16	
\$0002		ldi r16, low (RAMEND)	
\$0003		out SPL, r16	; Проініціалізували ; покажчик стека
\$0004		Sei	; Дозволили переривання
\$0005		<команда> xxx	

```

...
.org $F002
$F002      jmp EXT_INT0      ; Оброблювач IRQ0
$F004      jmp EXT_INT1      ; Оброблювач IRQ1
...
$F044      jmp SPM_RDY       ; Оброблювач переривання
                                ; "готовність SPM"

```

3. Вектор скидання розташовується на початку області завантажувача, а таблиця векторів переривань - на початку пам'яті програм (BOOTRST = 0, 1VSEL = 0)

Address	Labels	Code	Comments
.org \$0002			
\$0002		jmp EXT_INT0	; Оброблювач IRQ0
\$0004		jmp EXT_INT1	; Оброблювач IRQ1
...		...	...
\$0044		jmp SPM_RDY	; Оброблювач переривання ; "готовність SPM"
...		...	...
.org \$F000			
\$F000	RESET:	ldi r16, high (RAMEND)	; Початок основної програми
\$F001		out SPH, r16	
\$F002		ldi r16, low (RAMEND)	
\$F003		out SPL, r16	; Проініціалізували ; показчик стека
\$F004		Sei	; Дозволили переривання
\$F005		<команда> xxx	

4. Вектор скидання і таблиця векторів переривань розташовуються в області завантажувача (BOOTRST = 0, IVSEL = 1)

Address	Labels	Code	Comments
.org \$F000			
\$F000		jmp RESET	; Оброблювач скидання
\$F002		jmp EXT_INT0	; Оброблювач IRQ0
\$F004		jmp EXT_INT1	; Оброблювач IRQ1
...		...	...
\$F044		jmp SPM_RDY	; Оброблювач переривання ; "готовність SPM"
...		...	...
\$F046	RESET:	ldi r16, high (RAMEND)	; Початок основної програми
\$F047		out SPH, r16	
\$F048		ldi r16, low (RAMEND)	
\$F049		out SPL, r16	; Проініціалізували ; показник стека
\$F04A		Sei	; Дозволили переривання
\$F04B		<команда> xxx	

Нижче в якості прикладу (таблиця 6.2) наведено таблицю векторів переривань деяких AVR-мікроконтролерів сімейства Mega.

6.1.3 Обробка переривань

Для глобального дозволу переривань прапорець I регістра SREG повинний бути встановлений в «1». А дозвіл або заборона (маскування) окремих переривань виконується встановленням або скиданням відповідних розрядів регістрів масок переривань, розглянуто нижче.

Таблиця 6.2 - Таблиця векторів переривань моделей ATmega640x/1280x/
1281x/2560x/2561x

Джерело	Опис	№	Адреса
INT0	Зовнішнє переривання 0	1	\$0002
INT1	Зовнішнє переривання 1	2	\$0004
INT2	Зовнішнє переривання 2	3	\$0006
INT3	Зовнішнє переривання 3	4	\$0008
INT4	Зовнішнє переривання 4	5	\$000A
INT5	Зовнішнє переривання 5	6	\$000C
INT6	Зовнішнє переривання 6	7	\$000E
INT7	Зовнішнє переривання 7	8	\$0010
PCINT0	Переривання 0 за зміною стану виводів	9	\$0012
PCINT1	Переривання 1 за зміною стану виводів	10	\$0014
PCINT2	Переривання 2 за зміною стану виводів	11	\$0016 ¹⁾
WDT	Тайм-аут вартового таймера	12	\$0018
TIMER2 COMPA	Співпадіння А таймера/лічильника T2	13	\$001A
TIMER2 COMPB	Співпадіння В таймера/лічильника T2	14	\$001C
TIMER2 OVF	Переповнення таймера/лічильника T2	15	\$001E
TIMER1 CAPT	Захоплення таймера/лічильника T1	16	\$0020
TIMER1 COMPA	Співпадіння А таймера/лічильника T1	17	\$0022
TIMER1 COMPB	Співпадіння В таймера/лічильника T1	18	\$0024
TIMER1 COMPC	Співпадіння С таймера/лічильника T1	19	\$0026
TIMER1 OVF	Переповнення таймера/лічильника T1	20	\$0028
TIMER0 COMPA	Співпадіння А таймера/лічильника T0	21	\$002A
TIMER0 COMPB	Співпадіння В таймера/лічильника T0	22	\$002C
TIMER0 OVF	Переповнення таймера/лічильника T0	23	\$002E
SPI, STC	Передачу по SPI завершено	24	\$0030
USART0, RX	USART0, прийом завершено	25	\$0032

Продовження таблиці 6.2

Джерело	Опис	№	Адреса
USART0, UDRE	Регістр даних USART0 порожній	26	\$0034
USART0, TX	USART0, передачу завершено	27	\$0036
ANA_COMP	Аналоговий компаратор	28	\$0038
ADC	Перетворення АЦП завершено	29	\$003A
EE_RDY	EEPROM готово	30	\$003C
TIMER3 CAPT	Захоплення таймера/лічильника T3	31	\$003E
TIMER3 COMPA	Співпадіння А таймера/лічильника T3	32	\$0040
TIMER3 COMPB	Співпадіння В таймера/лічильника T3	33	\$0042
TIMER3 COMPC	Співпадіння С таймера/лічильника T3	34	\$0044
TIMER3 OVF	Переповнення таймера/лічильника T3	35	\$0046
USART1, RX	USART1, прийом завершено	36	\$0048
USART1, UDRE	Регістр даних USART порожній	37	\$004A
USART1, TX	USART1, передачу завершено	38	\$004C
TWI	Переривання від модуля TWI	39	\$004E
SPM_RDY	Готовність SPM	40	\$0050
TIMER4 CAPT	Захоплення таймера/лічильника T4	41	\$0052 ¹⁾
TIMER4 COMPA	Співпадіння А таймера/лічильника T4	42	\$0054
TIMER4 COMPB	Співпадіння В таймера/лічильника T4	43	\$0056
TIMER4 COMPC	Співпадіння С таймера/лічильника T4	44	\$0058
TIMER4 OVF	Переповнення таймера/лічильника T4	45	\$005A
TIMER5 CAPT	Захоплення таймера/лічильника T5	46	\$005C ¹⁾
TIMER5 COMPA	Співпадіння А таймера/лічильника T5	47	\$005E
TIMER5 COMPB	Співпадіння В таймера/лічильника T5	48	\$0060
TIMER5 COMPC	Співпадіння С таймера/лічильника T5	49	\$0062
TIMER5 OVF	Переповнення таймера/лічильника T5	50	\$0064
USART2, RX	USART2, прийом завершено	51	\$0066 ¹⁾
USART2, UDRE	Регістр даних USART2 порожній	52	\$0068 ¹⁾
USART2, TX	USART2, передачу завершено	53	\$006A ¹⁾

Продовження таблиці 6.2

Джерело	Опис	№	Адреса
USART3, RX	USART3, прийом завершено	54	\$006C ¹⁾
USART3, UDRE	Регістр даних USART3 порожній	55	\$006E ¹⁾
USART3, TX	USART3, передачу завершено	56	\$0070 ¹⁾
¹⁾ Тільки в моделях ATmega640х/1280х/2560х			

Обробка переривань здійснюється наступним чином:

- при виконанні умов, необхідних для генерації переривання, прапорець, що відповідає цьому перериванню, встановлюється в «1», а прапорець I апаратно скидається, забороняючи тим самим обробку наступних переривань. Однак у підпрограмі обробки переривання цей прапорець можна буде встановити в «1» для дозволу вкладених переривань;

- якщо переривання дозволено (прапорець дозволу переривання встановлено) у лічильник команд завантажується адреса вектора відповідного переривання. При цьому прапорець переривання апаратно скидається. Ряд прапорців переривань може бути також скинутий записом лог. «1» у розряд регістра, який відповідає прапорцю. Якщо ж переривання заборонено (прапорець дозволу переривання скинутий), прапорець переривання залишається у стані лог. «1» до дозволу переривання (у цьому випадку він буде скинутий апаратно), або до програмного скидання цього прапорця;

- виконується підпрограма обробки переривання;

- виконується команда повернення з переривання RETI, при цьому прапорець I апаратно встановлюється в «1», дозволяючи обробку наступних переривань;

– центральний процесор автоматично відновлює вміст лічильника команд. Потім основна програма продовжує своє виконання з того місця, де вона була перервана.

При виклику підпрограм обробки переривань вміст регістра стану SREG не зберігається. Тому користувач повинен самостійно запам'ятовувати вміст цього регістра при вході в підпрограму обробки переривання (якщо це необхідно) і відновлювати його значення перед викликом команди RETI.

Варто пам'ятати, що для переривань, викликаних динамічними подіями (наприклад, для переривання, яке генерується при рівності вмісту лічильного регістра і регістра порівняння таймера), прапорець переривання встановлюється тільки в момент виникнення події. Якщо прапорець переривання скинуто, а умови генерації переривання присутні, прапорець буде встановлено тільки в момент виникнення наступної події.

З іншого боку, для зовнішніх переривань, які генеруються за рівнем, прапорці не передбачені, тому інформація про ці переривання буде зберігатися доти, доки наявна подія, яка викликає переривання.

AVR-мікроконтролери сімейства Classic підтримують чергу переривань. Вона працює наступним чином: якщо умови генерації одного або більше переривань виникають у той час, коли прапорець загального дозволу переривань скинутий (усі переривання заборонено), відповідні прапорці встановлюються в «1» і залишаються в цьому стані до встановлення прапорця загального дозволу переривань. Після дозволу переривань виконується їхня обробка відповідно до пріоритету.

Частина часу відгуку для будь-якого переривання становить 5 тактів в моделях ATmega2560x/2561x і 4 такти - в інших моделях. Протягом цього часу відбувається збереження лічильника команд у стеку. Протягом наступних двох (для моделей ATmega8515x/8535x, ATmega8x і ATmega48x/88x) або трьох тактів (для інших моделей) виконується

команда переходу до підпрограми обробки переривання. Якщо переривання відбудеться під час виконання команди, яка триває кілька циклів, то генерація переривання відбудеться тільки після виконання цієї команди. Якщо ж переривання відбудеться під час перебування мікроконтролера у «сплячому» режимі, то час відгуку збільшується ще на 4 або 5 тактів.

Повернення в основну програму займає 4 такти (в моделях ATmega2560x/2561x - 5 тактів), протягом яких відбувається відновлення лічильника команд зі стека. Після виходу з переривання процесор завжди виконує одну команду основної програми, перш ніж обслужити будь-яке відкладене переривання.

6.1.4 Зовнішні переривання

В мікроконтролерах сімейства Mega є два різновиди зовнішніх переривань. Переривання першого типу (так звані «звичайні») генеруються при появі на вході зовнішнього переривання заданого сигналу. Ці переривання присутні у всіх мікроконтролерах сімейства (конкретне число таких переривань залежить від моделі). Переривання другого типу генеруються при будь-якій зміні стану певних виводів мікроконтролера. Наявність тих чи інших зовнішніх переривань в різних моделях показано в таблиці 6.3.

Зверніть увагу, що на певних входах зовнішніх переривань виявлення фронтів сигналів відбувається асинхронно, тобто не вимагає наявності тактового сигналу $f_{clk/O}$. Зовнішні переривання за низьким рівнем і переривання за зміною стану виводів завжди реєструються асинхронно. Відповідно, всі ці переривання можуть використовуватися для виведення мікроконтролера зі «сплячих» режимів, відмінних від режиму Idle.

Таблиця 6.3 – Зовнішні переривання

Зовнішні переривання	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/645x	ATmega3250x/6450x	ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x
INT0	•	•	•	•	• ¹⁾	•	•	• ¹⁾	•	•	•	• ¹⁾
INT1	•	•	•	•	• ¹⁾	•	•	• ¹⁾				• ¹⁾
INT2	• ¹⁾	• ¹⁾		• ¹⁾	• ¹⁾		• ¹⁾	• ¹⁾				• ¹⁾
INT3					• ¹⁾							• ¹⁾
INT4					•							•
INT5					•							•
INT6					•							•
INT7					•							•
PCIO						•	•	•	•	•	•	•
PC11						•	•	•	•	•	•	•
PC12						•		•			•	•
PC13								•			•	
¹⁾ Асинхронне виявлення фронтів сигналу												

Для дозволу/заборони «звичайних» зовнішніх переривань в залежності від моделі (таблиця 6.4) використовується або загальний регістр керування перериваннями GICR, або регістр маскуванія зовнішніх переривань EIMSK. Для дозволу/заборони переривань за зміною стану виводів використовуються або ті ж самі регістри, або окремий регістр PCICR.

Таблиця 6.4 –Регістри для дозволу/заборони зовнішніх переривань

Модель	Переривання	Регістр	Адреса
ATmega8515x/8535x	INT0...INT2	GICR	\$3B (\$5B)
ATmega8x	INT0, INT1	GICR	\$3B (\$5B)
ATmega16x/32x	INT0...INT2	GICR	\$3B (\$5B)
ATmega64x/128x	INT0...INT7	EIMSK	\$39 (\$59)
ATmega48x/88x/168x	INT0, INT1	EIMSK	\$1D (\$3D)
	PCI0...PCI2	PCICR	\$68
ATmega162x	INT0...INT2, PCI0, PCI1	GICR	\$3B (\$5B)
ATmega164x/324x/644x	INT0...INT2	EIMSK	\$1D (\$3D)
	PCI0...PCI3	PCICR	\$68
ATmega165x/325x/645x	INT0, PCI0, PCI1	EIMSK	\$1D (\$3D)
ATmega3250x/6450x	INT0, PCI0...PCI3	EIMSK	\$1D (\$3D)
ATmega640x, ATmega1280x/1281x/2560x/2561x	INT0...INT7	EIMSK	\$1D (\$3D)
	PCI0...PCI2	PCICR	\$68

Формати регістрів GICR, EIMSK і PCICR різних моделей показані на рисунку 6.1; описи їх розрядів, які стосуються переривань, наведено в таблиці 6.5.

Для індикації настання «звичайних» зовнішніх переривань в мікроконтролерах Mega використовується або загальний регістр прапорців переривань GIFR, або регістр прапорців зовнішніх переривань EIFR (таблиця 6.6). Для індикації переривань за зміною стану виводів використовуються або ті ж самі регістри, або окремий регістр PCIFR.

	7	6	5	4	3	2	1	0	
	INT1	INT0	—	—	—	—	IVSEL	IVSE	ATmega8x
Читання (R)/Запис (W)	R/W	R/W	R	R	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega8515x ATmega8535x ATmega16x ATmega32x
	INT1	INT0	INT2	—	—	—	IVSEL	IVSE	
Читання (R)/Запис (W)	R/W	R/W	R/W	R	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega162x
	INT1	INT0	INT2	PCIE1	PCIE0	—	IVSEL	IVSE	
Читання (R)/Запис (W)	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
а)									
	7	6	5	4	3	2	1	0	ATmega64x/128x ATmega640x ATmega1280x/1281x ATmega2560x/2561x
	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	
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	
	7	6	5	4	3	2	1	0	ATmega48x/88x/168x
	—	—	—	—	—	—	INT1	INT0	
Читання (R)/Запис (W)	R	R	R	R	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega164x/324x/644x
	—	—	—	—	—	INT2	INT1	INT0	
Читання (R)/Запис (W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega165x
	PCIE1	PCIE0	—	—	—	—	—	INT0	
Читання (R)/Запис (W)	R/W	R/W	R	R	R	R	R	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega325x/645x
	—	—	PCIE1	PCIE0	—	—	—	INT0	
Читання (R)/Запис (W)	R	R	R/W	R/W	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega3250x/6450x
	PCIE3	PCIE2	PCIE1	PCIE0	—	—	—	INT0	
Читання (R)/Запис (W)	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
б)									
	7	6	5	4	3	2	1	0	ATmega48x/88x/168x ATmega640x ATmega1280x/1281x ATmega2560x/2561x
	—	—	—	—	—	PCIE2	PCIE1	PCIE0	
Читання (R)/Запис (W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega164x/324x/644x
	—	—	—	—	PCIE3	PCIE2	PCIE1	PCIE0	
Читання (R)/Запис (W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
в)									

Рисунок 6.1 – Формат регістрів для дозволу/заборони зовнішніх переривань: GICR (а), EIMSK (б) и PCICR (в)

Таблиця 6.5 – Розряди регістрів GICR, EIMSK та PCICR

Назва	Опис
INT7	<u>Дозвіл зовнішнього переривання INT7.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання з виводу INT7. Умова генерації переривання визначається вмістом розрядів ISC71 і ISC70 регістра EICRB
INT6	<u>Дозвіл зовнішнього переривання INT6.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання з виводу INT6. Умова генерації переривання визначається вмістом розрядів ISC61 і ISC60 регістра EICRB
INT5	<u>Дозвіл зовнішнього переривання INT5.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання з виводу INT5. Умова генерації переривання визначається вмістом розрядів ISC51 і ISC50 регістра EICRB
INT4	<u>Дозвіл зовнішнього переривання INT4.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання з виводу INT4. Умова генерації переривання визначається вмістом розрядів ISC41 і ISC40 регістра EICRB
INT3	<u>Дозвіл зовнішнього переривання INT3.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання з виводу INT3. Умова генерації переривання визначається вмістом розрядів ISC31 і ISC30 регістра EICRA
INT2	<u>Дозвіл зовнішнього переривання INT2.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання з виводу INT2. Умова генерації переривання визначається вмістом розрядів ISC21 і ISC20 регістра EICRA або розряду ISC2 регістра EMCUCR (ATmega8515x і ATmega162x) або MCUCSR (ATmega8535x і ATmega16x/32x)
INT1	<u>Дозвіл зовнішнього переривання INT1.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання з виводу INT1. Умова генерації переривання визначається вмістом розрядів ISC11 і ISC10 регістра EICRA або MCUCR
INT0	<u>Дозвіл зовнішнього переривання INT0.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання з виводу INT0. Умова генерації переривання визначається вмістом розрядів ISC01 і ISC00 регістра EICRA або MCUCR
PCIE3	<u>Дозвіл переривання за зміною стану виводів 3-ї групи.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, дозволяється зовнішнє переривання за зміною стану виводів PCINT31...24 мікроконтролера. До виникнення переривання призводить будь-яка зміна сигналу на відповідному виводі

Продовження таблиці 6.5

Назва	Опис
PCIE2	<u>Дозвіл переривання за зміною стану виводів 2-ї групи.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, то дозволяється зовнішнє переривання за зміною стану виводів PCINT23...16 мікроконтролера. До виникнення переривання призводить будь-яка зміна сигналу на відповідному виводі
PCIE1	<u>Дозвіл переривання за зміною стану виводів 1-ї групи.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, то дозволяється зовнішнє переривання за зміною стану виводів PCINT15...8 мікроконтролера. До виникнення переривання призводить будь-яка зміна сигналу на відповідному виводі
PCIE0	<u>Дозвіл переривання за зміною стану виводів 0-ї групи.</u> Якщо в цьому розряді записана лог. 1 і прапорець І регістра SREG також встановлений в 1, то дозволяється зовнішнє переривання за зміною стану виводів PCINT7...0 мікроконтролера. До виникнення переривання призводить будь-яка зміна сигналу на відповідному виводі

Таблиця 6.6 – Регістри прапорців зовнішніх переривань

Модель	Переривання	Регістр	Адреса
ATmega8515x/8535x	INT0...INT2	GIFR	\$3A (\$5A)
ATmega8x	INT0, INT1	GIFR	\$3A (\$5A)
ATmega16x/32x	INT0...INT2	GIFR	\$3A (\$5A)
ATmega64x/128x	INT0...INT7	EIFR	\$38 (\$58)
ATmega48x/88x/168x	INT0, INT1	EIFR	\$1C (\$3C)
	PCI0...PCI2	PCIFR	\$1B (\$3B)
ATmega162x	INT0...INT2, PCI0, PCI1	GIFR	\$3A (\$5A)
ATmega164x/324x/644x	INT0...INT2	EIFR	\$1C (\$3C)
	PCI0...PCI3	PCIFR	\$1B (\$3B)
ATmega165x/325x/645x	INT0, PCI0, PCI1	EIFR	\$1C (\$3C)
ATmega3250x/6450x	INT0, PCI0...PCI3	EIFR	\$1C (\$3C)
ATmega640x, ATmega1280x/1281x/2560x/2561x	INT0...INT7	EIFR	\$1C (\$3C)
	PCI0...PCI2	PCIFR	\$1B (\$3B)

Формати регістрів GIFR, EIFR і PCIFR різних моделей показані на рисунку 6.2, а описи їх розрядів наведені у таблиці 6.7.

	7	6	5	4	3	2	1	0	
	INTF1	INTF0	—	—	—	—	—	—	ATmega8x
Читання (R)/Запис (W)	R/W	R/W	R	R	R	R	R	R	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega851x ATmega8535x ATmega16x ATmega32x
	INTF1	INTF0	INTF2	—	—	—	—	—	
Читання (R)/Запис (W)	R/W	R/W	R/W	R	R	R	R	R	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega162x
	INTF1	INTF0	INTF2	PCIF1	PCIF0	—	—	—	
Читання (R)/Запис (W)	R/W	R/W	R/W	R/W	R/W	R	R	R	
Початкове значення	0	0	0	0	0	0	0	0	
а)									
	7	6	5	4	3	2	1	0	ATmega64x/128x ATmega640x ATmega1280x/1281x ATmega2560x/2561x
	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0	
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	
	7	6	5	4	3	2	1	0	ATmega48x/88x/168x
	—	—	—	—	—	—	INTF1	INTF0	
Читання (R)/Запис (W)	R	R	R	R	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega164x/324x/644x
	—	—	—	—	—	INTF2	INTF1	INTF0	
Читання (R)/Запис (W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega165x
	PCIF1	PCIF0	—	—	—	—	—	INTF0	
Читання (R)/Запис (W)	R/W	R/W	R	R	R	R	R	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega325x/645x
	—	—	PCIF1	PCIF0	—	—	—	INTF0	
Читання (R)/Запис (W)	R	R	R/W	R/W	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega3250x/6450x
	PCIF3	PCIF2	PCIF1	PCIF0	—	—	—	INTF0	
Читання (R)/Запис (W)	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
б)									
	7	6	5	4	3	2	1	0	ATmega48x/88x/168x ATmega640x ATmega1280x/1281x ATmega2560x/2561x
	—	—	—	—	—	PCIF2	PCIF1	PCIF0	
Читання (R)/Запис (W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega164x/324x/644x
	—	—	—	—	PCIF3	PCIF2	PCIF1	PCIF0	
Читання (R)/Запис (W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
в)									

Рисунок 6.2 – Формат регістрів прапорців зовнішніх переривань: GIFR (а), EIFR (б) та PCIFR (в)

Таблиця 6.7 – Біти регістрів GIFR, EIFR і PCIFR

Назва	Опис
INTF7	<u>Прапорець зовнішнього переривання INT7</u> . Якщо в результаті події на виводі INT7 сформувався запит на зовнішнє переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Прапорець INTF7 скинутий постійно, якщо генерація переривання повинна відбуватися за низьким рівнем сигналу на виводі INT7
INTF6	<u>Прапорець зовнішнього переривання INT6</u> . Якщо в результаті події на виводі INT6 сформувався запит на зовнішнє переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Прапорець INTF6 скинутий постійно, якщо генерація переривання повинна відбуватися за низьким рівнем сигналу на виводі INT6
INTF5	<u>Прапорець зовнішнього переривання INT5</u> . Якщо в результаті події на виводі INT5 сформувався запит на зовнішнє переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Прапорець INTF5 скинутий постійно, якщо генерація переривання повинна відбуватися за низьким рівнем сигналу на виводі INT5
INTF4	<u>Прапорець зовнішнього переривання INT4</u> . Якщо в результаті події на виводі INT4 сформувався запит на зовнішнє переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Прапорець INTF4 скинутий постійно, якщо генерація переривання повинна відбуватися за низьким рівнем сигналу на виводі INT4
INTF3	<u>Прапорець зовнішнього переривання INT3</u> . Якщо в результаті події на виводі INT3 сформувався запит на зовнішнє переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Прапорець INTF3 скинутий постійно, якщо генерація переривання повинна відбуватися за низьким рівнем сигналу на виводі INT3
INTF2	<u>Прапорець зовнішнього переривання INT2</u> . Якщо в результаті події на виводі INT2 сформувався запит на зовнішнє переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Прапорець INTF2 скинутий постійно, якщо генерація переривання повинна відбуватися за низьким рівнем сигналу на виводі INT2

Продовження таблиці 6.7

INTF1	<u>Прапорець зовнішнього переривання INT1.</u> Якщо в результаті події на виводі INT1 сформувався запит на зовнішнє переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Прапорець INTF1 скинутий постійно, якщо генерація переривання повинна відбуватися за низьким рівнем сигналу на виводі INT1
INTF0	<u>Прапорець зовнішнього переривання INT0.</u> Якщо в результаті події на виводі INT0 сформувався запит на зовнішнє переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Прапорець INTF0 скинутий постійно, якщо генерація переривання повинна відбуватися за низьким рівнем сигналу на виводі INT0
PCIF3	<u>Прапорець переривання за зміною стану виводів 3-ї групи.</u> Якщо в результаті події на будь-якому з виводів PCINT31...24 сформувався запит на переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1
PCIF2	<u>Прапорець переривання за зміною стану виводів 2-ї групи.</u> Якщо в результаті події на будь-якому з виводів PCINT23...16 сформувався запит на переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1
PCIF1	<u>Прапорець переривання за зміною стану виводів 1-ї групи.</u> Якщо в результаті події на будь-якому з виводів PCINT15...8 сформувався запит на переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1
PCIF0	<u>Прапорець переривання за зміною стану виводів 0-ї групи.</u> Якщо в результаті події на будь-якому з виводів PCINT7...0 сформувався запит на переривання, цей розряд встановлюється в 1. Прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1

Слід мати на увазі, що в ранніх моделях мікроконтролерів, таких як ATmega8515x/8535x, ATmega8x/16x/32x і ATmega64x/128x, для програмного скидання прапорців переривань не можна використовувати команду SBI (встановити розряд регістра введення/виведення), тому що при її виконанні будуть скинуті всі прапорці в регістрі. Це пов'язано з тим, що команда спочатку зчитує вміст регістра, потім змінює вказаний розряд і записує результат назад в регістр. Аналогічно, при виконанні команди CBI

(скинути розряд регістра введення/виведення), будуть скинуті всі розряди регістра, крім зазначеного. В інших моделях ці команди впливають тільки на заданий розряд, тому для скидання прапорця переривання в даних моделях можна використовувати команду SBI.

Переривання INT0...INT7 можуть бути згенеровані за наростаючим/спадаючим фронтом сигналу або при появі НИЗЬКОГО рівня на вході (крім переривання INT2 в моделях ATmega8515x/8535x, ATmega16x/32x і ATmega162x). Переривання PCI0...PCI3 генеруються при зміні стану будь-якого заданого виводу відповідної групи.

Умови генерації переривань INT0...INT3 визначаються станом певних розрядів регістра MCUCR або EICRA, а умова генерації переривання INT2 в моделях ATmega8515x/8535x, ATmega16x/32x і ATmega162x визначається станом розряду ISC2 регістра EMCUCR або MCUCSR. Умови генерації переривань INT7...INT4 визначаються регістром EICRB. Вибір виводів, зміна стану яких викликає генерацію переривань PCI0...PCI3, здійснюється за допомогою регістрів PCMSK3...PCMSK0. Відповідність регістрів конкретним моделям наведено в таблиці 6.8.

Таблиця 6.8 – Регістри, які визначають умови генерації зовнішніх переривань

Модель	Переривання	Регістр	Адреса
ATmega8515x	INT0, INT1	MCUCR	\$35 (\$55)
	INT2	EMCUCR	\$36 (\$56)
ATmega8535x	INT0, INT1	MCUCR	\$35 (\$55)
	INT2	MCUCSR	\$34 (\$54)
ATmega8x	INT0, INT1	MCUCR	\$35 (\$55)

Продовження таблиці 6.8

ATmega16x/32x	INT0, INT1	MCUCR	\$35 (\$55)
	INT2	MCUCSR	\$34 (\$54)
ATmega64x/128x	INT0...INT3	EICRA	\$6A
	INT4...INT7	EICRB	\$3A (\$5A)
ATmega48x/88x/168x	INT0, INT1	EICRA	\$69
	PCI0	PCMSK0	\$6B
	PCI1	PCMSK1	\$6C
	PCI2	PCMSK2	\$6D
ATmega162x	INT0, INT1	MCUCR	\$35 (\$55)
	INT2	EMCUCR	\$36 (\$56)
	PCI0	PCMSK0	\$6B
	PCI1	PCMSK1	\$6C
ATmega164x/324x/644x	INT0...INT2	EICRA	\$69
	PCI0	PCMSK0	\$6B
	PCI1	PCMSK1	\$6C
	PCI2	PCMSK2	\$6D
	PCI3	PCMSK3	\$73
ATmega165x/325x/645x	INT0	EICRA	\$69
	PCI0	PCMSK0	\$6B
	PCI1	PCMSK1	\$6C
ATmega325x/3250x, ATmega645x/6450x	INT0	EICRA	\$69
	PCI0	PCMSK0	\$6B
	PCI1	PCMSK1	\$6C
	PCI2	PCMSK2	\$6D
	PCI3	PCMSK3	\$73
ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x	INT0...INT3	EICRA	\$69
	INT4...INT7	EICRB	\$6A
	PCI0	PCMSK0	\$6B
	PCI1	PCMSK1	\$6C
	PCI2	PCMSK2	\$6D

Формати регістрів, перелічених у попередній таблиці, показані на рисунках 6.3 та 6.4, а опис їх розрядів наведено в таблицях 6.9...6.11.

	7	6	5	4	3	2	1	0	ATmega8515x/8535x ATmega8x ATmega16x/32x ATmega162x
	X	X	X	X	ISC11	ISC10	ISC01	ISC00	
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	
а)									
	7	6	5	4	3	2	1	0	ATmega8535x ATmega16x/32x
	X	ISC2	—	X	WDRF	BORF	EXTRF	PORF	
Читання (R)/Запис (W)	X	R/W	R	X	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
б)									
	7	6	5	4	3	2	1	0	ATmega8515x ATmega162x
	SM0	SRL2	SRL1	SRL0	SRW01	SRW00	SRW11	ISC2	
Читання (R)/Запис (W)	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
в)									
	7	6	5	4	3	2	1	0	ATmega48x/88x/168x
	—	—	—	—	ISC11	ISC10	ISC01	ISC00	
Читання (R)/Запис (W)	R	R	R	R	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
г)									
	7	6	5	4	3	2	1	0	ATmega164x/324x/644x
	—	—	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	
Читання (R)/Запис (W)	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
д)									
	7	6	5	4	3	2	1	0	ATmega165x ATmega325x/3250x ATmega645x/6450x
	—	—	—	—	—	—	ISC01	ISC00	
Читання (R)/Запис (W)	R	R	R	R	R	R	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
е)									
	7	6	5	4	3	2	1	0	ATmega64x/128x ATmega640x ATmega1280x/1281x ATmega2560x/2561x
	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	
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	
ж)									
	7	6	5	4	3	2	1	0	ATmega64x/128x ATmega640x ATmega1280x/1281x ATmega2560x/2561x
	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	
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 6.3 – Формат регістрів, які визначають умови генерації зовнішніх переривань: MCUCR (а), MCUCSR (б), EMCUCR (в), EICRA (г) і EICRB(д)

PCMSK0	7	6	5	4	3	2	1	0
Читання (R)/Запис (W)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
Початкове значення	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0	0	0	0	0	0	0	0

PCMSK1	7	6	5	4	3	2	1	0
Читання (R)/Запис (W)	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
Початкове значення	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0	0	0	0	0	0	0	0

PCMSK2	7	6	5	4	3	2	1	0
Читання (R)/Запис (W)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
Початкове значення	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0	0	0	0	0	0	0	0

PCMSK3	7	6	5	4	3	2	1	0
Читання (R)/Запис (W)	PCINT31	PCINT30	PCINT29	PCINT28	PCINT27	PCINT26	PCINT25	PCINT24
Початкове значення	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0	0	0	0	0	0	0	0

Рисунок 6.4 – Формат регістрів PCMSK0... PCMSK3

Таблиця 6.9 – Визначення умов генерації зовнішніх переривань (регістри MCUCR, EICRA та EICRB)

Регістр	Біт	Опис		
MCUCR/ EICRA	ISC31, ISC30	Визначають умови генерації зовнішніх переривань INT0...INT7 наступним чином:		
	ISC21, ISC20			
	ISC11, ISC10			
	ISC01, ISC00			
EICRB	ISC71, ISC70	ISCn1	ISCn1	Умова
	ISC61, ISC60	0	0	За НИЗЬКИМ рівнем на виводі INTn
	ISC51, ISC50	0	1	Зарезервовано
	ISC41, ISC40	1	0	За спадаючим фронтом сигналу на виводі INTn
		1	1	За наростаючим фронтом сигналу на виводі INTn

Примітка. Мінімальна тривалість імпульсу, яка гарантує генерацію переривання, дорівнює періоду тактового сигналу мікроконтролера, а в моделях ATmega164x/324x/644x – 50 нс.

Таблиця 6.10 – Визначення умови генерації асинхронного зовнішнього переривання INT2 (реєстри MCUCSR та EMCUCR)

Регістр	Біт	Опис	
MCUCSR/ EMCUCR	ISC2	Визначає умову генерації зовнішнього асинхронного переривання INT2 наступним чином:	
		ISC2	Умова
		0	За спадаючим фронтом сигналу на виводі INT2
		1	За наростаючим фронтом сигналу на виводі INT2

Примітка. Мінімальна тривалість імпульсу, яка гарантує генерацію переривання, становить 50 нс.

Таблиця 6.11 – Визначення виводів, які викликають генерацію переривань за зміною стану виводів (реєстри PCMSK3...0)

Регістр	Біт	Опис
PCMSK0	PCINT7 ... PCINT0	Визначають умову генерації переривання типу PC10. Якщо який-небудь розряд встановлений в 1, то зміна стану відповідного виводу викличе генерацію переривання
PCMSK1	PCINT15 ... PCINT8	Визначають умову генерації переривання типу PC11. Якщо який-небудь розряд встановлений в 1, то зміна стану відповідного виводу викличе генерацію переривання
PCMSK2	PCINT23 ... PCINT16	Визначають умову генерації переривання типу PC12. Якщо який-небудь розряд встановлений в 1, то зміна стану відповідного виводу викличе генерацію переривання
PCMSK3	PCINT31 ... PCINT24	Визначають умову генерації переривання типу PC13. Якщо який-небудь розряд встановлений в 1, то зміна стану відповідного виводу викличе генерацію переривання

На закінчення слід зазначити, що всі зовнішні переривання генеруються навіть у тому випадку, якщо відповідні виводи сконфігуровані як виходи. Ця особливість мікроконтролерів дозволяє генерувати переривання програмно.

6.1.5 ОСОБЛИВОСТІ ВИКОРИСТАННЯ МОДУЛЯ ПЕРЕРИВАНЬ У МІКРОКОНТРОЛЕРАХ XMEGA

6.1.5.1 Загальні відомості

Крім зовнішніх, більшість переривань є внутрішніми і сигналізують про зміну стану окремих периферійних модулів. Кожний модуль може підтримувати одне або декілька переривань і кожне з цих переривань може бути окремо дозволено або заборонено. Запит на переривання генерується в тому випадку, якщо в будь-який момент після його дозволу виявляється відповідна йому умова. У кожного переривання є окремий вектор переривання [12,13].

Мікроконтролери Xmega містять програмований багаторівневий контролер переривань (PMIC-контролер), який відповідає за обробку запитів переривань з урахуванням їх пріоритетів і рівнів. Після того, як запит переривання підтверджується PMIC-контролером, у лічильник програми завантажується адреса вектора переривання, а потім виконується процедура обробки переривання.

Кожне з переривань підтримує можливість налаштування рівня його пріоритету: низький, середній і високий. Переривання середнього рівня здатні переривати виконання процедури обробки переривання низького рівня. У свою чергу, переривання високого рівня можуть переривати процедури обробки переривань низького і середнього рівнів. Пріоритетність обробки переривань однакового рівня залежить від числового значення адреси вектора переривання: чим менше адреса вектора переривання, тим вище його пріоритет. Опціонально, пріоритетність переривань низького рівня може визначатися карусельним методом (round-robin), який гарантує обробку кожного з переривань за певну кількість часу.

Також передбачена підтримка переривань, що не маскуються (NMI).

6.1.5.2 Принцип дії

Щоб генерація переривань стала можливою, необхідно встановити розряд загального дозволу переривань (розряд I) у регістрі статусу мікроконтролера. Після підтвердження переривання, втрачається можливість скидання розряду I. Крім того, перед тим як використовувати той чи інший рівень переривань, його роботу потрібно дозволити.

Після дозволу переривання і виявлення його умови, РМІС-контролер приймає запит на обробку переривання. Далі, в залежності від рівня і пріоритету, переривання переходить або в статус "підтверджено", або в статус "відправлено". Якщо переривання підтверджується, то в лічильник програми завантажується вектор переривання. За адресою вектора переривання зазвичай знаходиться команда переходу до процедури обробки переривання. Після виходу з процедури обробки переривання, відновлюється виконання програми з того ж місця, де виникло переривання. Перед переходом до обслуговування відправленого переривання виконується одна інструкція.

Регістр статусу РМІС-контролера містить всю необхідну інформацію про його стан, яка гарантує повернення РМІС-контролера на коректний рівень переривання після виконання інструкції RETI (повернення з переривання) в кінці процедури обробки переривання. Під час виходу з переривання, РМІС-контролер перейде в той стан, в якому він був перед введенням переривання. Автоматичне збереження регістра статусу (SREG) при надходженні запиту на переривання не передбачено. Інструкцію RET (повернення з підпрограми) не можна використовувати для повернення з процедури обробки переривання, тому що вона не повертає РМІС-контролер в коректний стан.

Всі переривання і вектор скидання мають окрему адресу в просторі пам'яті програм. За найменшою адресою пам'яті програм розташований вектор скидання. У кожного з переривань є окремі розряди, за допомогою яких можна дозволити або заборонити переривання, а також вибрати його рівень його пріоритету. Дані розряди знаходяться в регістрах керування тих пристроїв введення/виведення (ПВВ), які підтримують генерацію переривань. Більш детально про це див. у розділах, присвячених опису таких ПВВ.

У кожного з переривань є пов'язаний з ним прапорець переривання. Даний прапорець встановлюється, коли виникає умова для генерації переривання, незалежно від того, дозволено воно чи ні. Прапорці більшості переривань автоматично скидаються при переході на вектор переривання. Прапорець переривання можна скинути і програмно. Для цього необхідно виконати запис в нього одиниці. Є декілька прапорців переривань, які не скидаються автоматично при переході на вектор переривання. Також є ще ряд переривань, прапорці яких автоматично скидаються при виконанні читання або запису пов'язаного з ними регістра. Подробиці знаходяться в описі кожного окремого прапорця переривання.

Якщо під час обробки або відправлення запиту одного переривання виникає умова для генерації ще одного, але більш низькопріоритетного переривання, прапорець цього переривання запам'ятовується у встановленому стані до тих пір, поки до нього не дійде черга. Якщо ж переривання не дозволено, але виникає умова для його генерації, прапорець переривання буде також встановлений і збережений в такому стані до дозволу переривання або програмного скидання цього прапорця. Аналогічним чином, якщо виникає одне або декілька умов переривань, коли скинуто розряд загального дозволу переривань, будуть встановлені відповідні прапорці переривань і запам'ятовуються в такому стані до загального дозволу переривань. Надалі черговість обробки всіх

відправлених запитів буде виконуватися з урахуванням їх пріоритету.

Передбачена можливість блокування переривань при виконанні коду програми із захищеного сектора, наприклад, коли запрограмовані розряди Boot Lock. Така можливість покращує захищеність програми (подробиці див. в темі програмування пам'яті програм).

Переривання автоматично блокуються на 4 цикли системної синхронізації після виконання запису коректної сигнатури в регістр захисту від зміни конфігурації.

Переривання, що не маскуються, (NMI) повністю реалізовані на апаратному рівні. Не передбачена можливість вибору, яке переривання відноситься до NMI, а яке буде звичайним. Перед тим як використовувати переривання, що не маскуються, їх необхідно дозволити. Які саме NMI підтримуються мікроконтролером, необхідно уточнити в документації.

Переривання, що не маскується, виконується незалежно від встановлення розряду I і не змінює його стан. Жодне з інших переривань не може перервати обробку NMI-переривання. При одночасному надходженні запитів на обробку декількох NMI-переривань, вони будуть оброблені з урахуванням їх пріоритету, який визначається за значеннями адрес векторів переривань: чим менша адреса, тим вищий пріоритет переривання.

6.1.5.3 Затримка реагування на переривання

Затримка реагування на переривання для всіх дозволених переривань становить не менше 5 циклів синхронізації ЦПП. Протягом цих 5 циклів синхронізації виконується поміщення лічильника програми в стек. Після завершення цих 5 циклів синхронізації, виконується процедура обробки переривання, на перехід до якої з вектора переривання необхідно ще три цикли синхронізації.

Якщо переривання виникає в момент виконання багатоциклової інструкції, то спочатку завершується виконання цієї інструкції, а вже потім обслуговується переривання. Якщо переривання виникає, коли МК-р знаходиться в економічному режимі роботи, то затримка реагування на переривання збільшується ще на 5 циклів синхронізації. Крім того, до цього часу також потрібно додати час, необхідний мікроконтролеру для виходу з обраного економічного режиму роботи.

Вихід з процедури обробки переривання вимагає п'ять циклів синхронізації. Протягом цих п'яти циклів синхронізації, зі стека витягується адреса повернення в лічильник програми та інкрементується показник стека.

6.1.5.4 Рівні переривань

Для кожного з джерел переривань можна окремо задати рівень переривання. Під час отримання запиту на переривання РМІС-контролер також приймає інформацію про його рівень. Рівні переривань і відповідні їм значення розрядів наведені в таблиці 6.12.

Рівень переривання, яке запитується, порівнюється з поточними рівнем і статусом у контролері переривання. Якщо надійшов запит на переривання, який має більш високий рівень, ніж той, що обробляється в даний момент, то обробка низькорівневого переривання буде перервана. По завершенні обробки переривання з більш високим рівнем, буде продовжено виконання процедури обробки більш низькорівневого переривання.

Таблиця 6.12 – Рівні переривань

Налаштування рівнів переривань	Групові конфігурації	Опис
00	OFF	Переривання відключено
01	LO	Переривання низького рівня
10	MED	Переривання середнього рівня
11	HI	Переривання високого рівня

6.1.5.5 Пріоритети переривань

Всі переривання, крім рівня, також мають пріоритет. При одночасному надходженні запитів від декількох джерел переривань, черговість їх обробки буде визначатися з урахуванням їх рівня і пріоритету. Пріоритезація переривань може бути статичною (рисунки 6.5) або динамічною (карусельною, round-robin) (рисунки 6.6). Пріоритезація переривань високого і середнього рівнів, а також переривань, що не маскуються, є тільки статичною, а пріоритезація низькорівневих переривань може бути як статичною, так і динамічною (задається користувачем).

6.1.5.5.1 Статичне задання пріоритетів

Вектори переривань мають фіксовані адреси. Коли застосовується статична пріоритезація, черговість обробки переривань одного і того ж рівня буде залежати від значення адреси вектора переривання: чим менше адреса, тим вищий пріоритет переривання (рисунки 6.5). Таблицю векторів переривань із зазначенням базових адрес для всіх вбудованих модулів і ПБВ, які підтримують генерацію переривань, можна знайти в документації на МК-р.

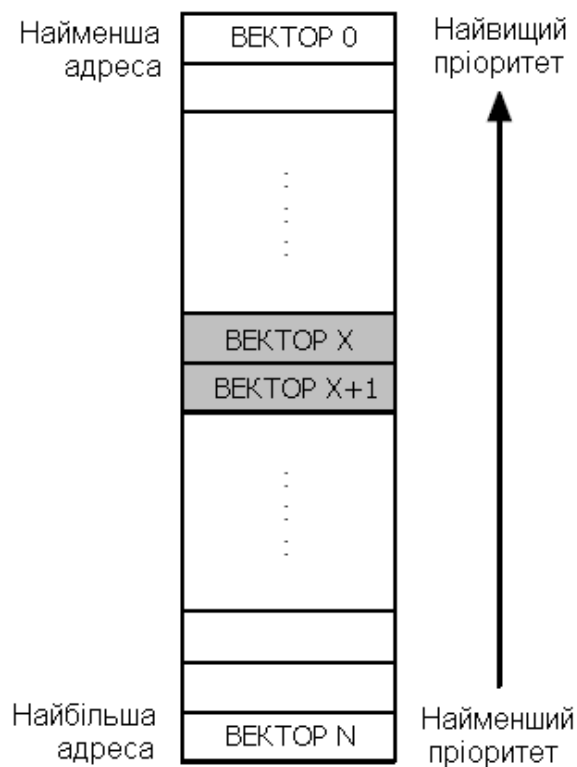


Рисунок 6.5 – Статичне задання пріоритетів

6.1.5.5.2 Динамічне задання пріоритетів

У застосуваннях з частою генерацією низькорівневих переривань статична пріоритезація може призвести до ситуації, коли одні переривання обробляються часто, а до інших - черга просто не доходить. Для виключення таких ситуацій передбачена опціональна динамічна пріоритезація низькорівневих переривань. Коли вона дозволяється, в момент надходження запитів від одного або декількох джерел низькорівневих переривань, найменший пріоритет буде мати те переривання, яке оброблялося останнім (рисунок 6.6).

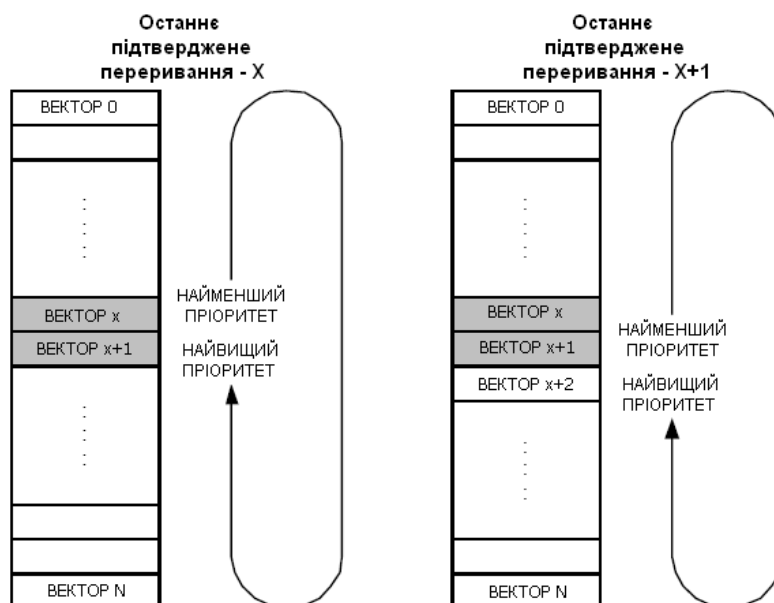


Рисунок 6.6 – Динамічне задання пріоритетів

6.1.5.6 Опис керуючих регістрів

6.1.5.6.1 Регістр статусу (STATUS – регістр)

Біт	7	6	5	4	3	2	1	0	
+0x00	NMIEX	-	-	-	-	HILVLEX	MEDLVLEX	LOLVLEX	STATUS
Чит./запис	Чит.	Чит.	Чит.	Чит.	Чит.	Чит.	Чит.	Чит.	
Початк. значення	0	0	0	0	0	0	0	0	

Біт 7 - NMIEX: обробка переривання, що не маскується.

Даний прапорець встановлюється на час обробки переривання, що не маскується. Скидання прапорця виконується при виконанні інструкції повернення з процедури обробки переривання (RETI).

Біти 6:3 - Res: резервні розряди.

Дані розряди є резервними і завжди зчитуються з нульовими значеннями. Для сумісності з новішими МК, виконуючи запис в цей регістр, завжди записуйте нулі в резервні розряди.

Біт 2 - HILVLEX: обробка переривання високого рівня.

Даний прапорець знаходиться у високому стані, коли виконується обробка переривання високого рівня або коли обробка переривання перервана перериванням, що не маскується. Прапорець автоматично скидається при виході з процедури обробки переривання (тобто при виконанні інструкції RETI).

Біт 1 - MEDLVLEX: обробка переривання середнього рівня.

Даний прапорець знаходиться у високому стані, коли виконується обробка переривання середнього рівня або коли обробка переривання перервана перериванням, що не маскується, або перериванням високого рівня. Прапорець автоматично скидається при виході з процедури обробки переривання (тобто при виконанні інструкції RETI).

Біт 0 - LOLVLEX: обробка переривання низького рівня.

Даний прапорець знаходиться у високому стані, коли виконується обробка переривання середнього рівня або коли обробка переривання перервана перериванням, що не маскується, або перериванням більш високого рівня. Прапорець автоматично скидається при виході з процедури обробки переривання (тобто при виконанні інструкції RETI).

6.1.5.6.2 Регістр пріоритету переривань (INTPRI–регістр)

Біт	7	6	5	4	3	2	1	0
+0x01	INTPRI[7:0]							INTPRI
Чит./ запис	Чит./ зап.	Чит./ зап.	Чит./ зап.	Чит./ зап.	Чит./ зап.	Чит./ зап.	Чит./ зап.	Чит./ зап.
Початк. значення	0	0	0	0	0	0	0	0

Біт 7:0 - INTPRI: пріоритет переривання.

Після дозволу динамічної пріоритезації, в даному регістрі буде

зберігатися вектор останнього підтвердженого низькорівневого переривання. Вектор переривання, що зберігається в цьому регістрі, має найменший пріоритет у разі надходження запитів від ще одного або декількох джерел низькорівневих переривань. Записом у цей регістр можна впливати на черговість обробки переривань. Після відключення динамічної пріоритезації вміст регістра не обнуляється, тому, якщо необхідно використовувати статичну пріоритезацію, в цей регістр необхідно записати нульове значення.

6.1.5.6.3 Регістр керування (CTRL - регістр)

Біт	7	6	5	4	3	2	1	0	
+0x00	RREN	IVSEL	-	-	-	HILVLEN	MEDLVLEN	LOLVLEN	CTRL
Чит./ запис	Чит./ зап.	Чит./ зап.	Чит.	Чит.	Чит.	Чит./ зап.	Чит./ зап.	Чит./ зап.	
Початк. значення	0	0	0	0	0	0	0	0	

Біт 7 - RREN: дозвіл динамічної пріоритезації.

Встановлення розряду RREN призводить до дозволу динамічної пріоритезації низькорівневих переривань. Якщо цей розряд дорівнює нулю, використовується статична пріоритезація, відповідно до якої пріоритет переривання визначається за адресою вектора переривання, причому чим менше адреса, тим вищий пріоритет переривання.

Біт 6 - IVSEL: настройка векторів переривань.

За замовчуванням, розряд IVSEL дорівнює нулю і вектори переривань знаходяться на початку сектора прикладної програми Flash-

пам'яті. Якщо ж у цей розряд записати одиницю, то вектори переривань будуть перенесені на початок завантажувального сектора Flash-пам'яті. Значення абсолютних адрес можна знайти в документації на МК.

Даний розряд захищений механізмом захисту від зміни конфігурації.

Біти 5:3 - Res: резервні розряди.

Дані розряди не використовуються і зарезервовані для використання в майбутньому. Для сумісності з новішими МК, виконуючи запис в цей регістр, завжди записуйте нулі в резервні розряди.

Біт 2 - HILVLEN: дозвіл переривань високого рівня.

Встановлення даного розряду дозволяє роботу переривань високого рівня. Якщо ж даний розряд буде дорівнювати 0, то переривання високого рівня будуть ігноруватися.

Біт 1 - MEDLVLEN: дозвіл переривань середнього рівня.

Встановлення даного розряду дозволяє роботу переривань середнього рівня. Якщо ж даний розряд буде дорівнювати 0, то переривання середнього рівня будуть ігноруватися.

Біт 0 - LOLVLEN: дозвіл переривань низького рівня.

Встановлення даного розряду дозволяє роботу переривань низького рівня. Якщо ж даний розряд буде дорівнювати 0, то переривання низького рівня будуть ігноруватися.

6.1.5.7 Таблиця векторів переривань мікроконтролера XMEGA A3

Як приклад в таблиці 6.13 наведено вектори переривань мікроконтролера XMEGA A3.

Таблиця 6.13 – Вектори переривань мікроконтролера XMEGA A3

Програмна адреса (базова адреса)	Джерело	Опис переривання
0x000	RESET	
0x002	OSCF_INT_vect	Переривання, що не маскується, при виявленні відмови кварцового генератора
0x004	PORTC_INT_base	Базова адреса переривань порту C
0x008	PORTR_INT_base	Базова адреса переривань порту R
0x00C	DMA_INT_base	Базова адреса переривань DMA-контролера
0x014	RTC_INT_base	Базова адреса переривань лічильника реального часу
0x018	TWIC_INT_base	Базова адреса переривань інтерфейсу TWI (порт C)
0x01C	TCC0_INT_base	Базова адреса переривань таймера-лічильника 0 (порт C)
0x028	TCC1_INT_base	Базова адреса переривань таймера-лічильника 1 (порт C)
0x030	SPIC_INT_vect	Базова адреса переривань інтерфейсу SPI (порт C)
0x032	USARTC0_INT_base	Базова адреса переривань інтерфейсу USART 0 (порт C)
0x03D	USARTC1_INT_base	Базова адреса переривань інтерфейсу USART 1 (порт C)
0x03E	AES_INT_vect	Вектор переривань AES
0x040	NVM_INT_base	Базова адреса переривань енергонезалежної пам'яті
0x044	PORTB_INT_base	Базова адреса переривань (порт B)
0x048	ACB_INT_base	Базова адреса переривань аналогового компаратора (порт B)
0x04E	ADCB_INT_base	Базова адреса переривань АЦП (порт B)
0x056	PORTE_INT_base	Базова адреса переривань порту E
0x05A	TWIE_INT_base	Базова адреса переривань інтерфейсу TWI (порт E)
0x05E	TCE0_INT_base	Базова адреса переривань таймера-лічильника 0 (порт E)
0x06A	TCE1_INT_base	Базова адреса переривань таймера-лічильника 1 (порт E)
0x072	SPIE_INT_vect	Базова адреса переривань інтерфейсу SPI (порт E)
0x074	USARTE0_INT_base	Базова адреса переривань інтерфейсу USART 0 (порт E)
0x07A	USARTE1_INT_base	Базова адреса переривань інтерфейсу USART 1 (порт E)

Продовження таблиці 6.13

Програмна адреса (базова адреса)	Джерело	Опис переривання
0x080	PORTD_INT_base	Базова адреса переривань порту D
0x084	PORTA_INT_base	Базова адреса переривань порту A
0x088	ACA_INT_base	Базова адреса переривань аналогового компаратора (порт A)
0x08E	ADCA_INT_base	Базова адреса переривань АЦП (порт A)
0x09A	TCD0_INT_base	Базова адреса переривань таймера-лічильника 0 (порт D)
0x0A6	TCD1_INT_base	Базова адреса переривань таймера-лічильника 1 (порт D)
0x0AE	SPID_INT_vector	Базова адреса переривань інтерфейсу SPI (порт D)
0x0B0	USARTD0_INT_base	Базова адреса переривань інтерфейсу USART 0 (порт D)
0x0B6	USARTD1_INT_base	Базова адреса переривань інтерфейсу USART 1 (порт D)
0x0D0	PORTF_INT_base	Базова адреса переривань порту F
0x0D8	TCF0_INT_base	Базова адреса переривань таймера-лічильника 0 (порт F)
0x0EE	USARTF0_INT_base	Базова адреса переривань інтерфейсу USART 0 (порт F)

6.1.5.8 Кількість переривань в мікроконтролерах XMEGA сімейств A1, A2 і A3

Таблиця 6.14 – Кількість переривань в мікроконтролерах XMEGA сімейств A1, A2 і A3

Сімейство	A1	A3	A4
Число переривань	45	36	29
NMI	1	1	1

У всіх представлених вище сімействах присутнє лише одне переривання, що не маскується, - відмова кварцового генератора.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Для чого необхідні переривання?
- 2) Що відбувається із вмістом лічильника команд РС при виникненні переривання?
- 3) Як виконується команда RETI?
- 4) Що визначає положення вектора переривання в таблиці векторів переривань?
- 5) Де може розташовуватися таблиця векторів переривань?
- 6) Яким чином здійснюється керування розміщенням таблиці векторів переривань?
- 7) Як здійснити глобальний дозвіл/заборону переривань? Як здійснити дозвіл/заборону окремого переривання?
- 8) Яким чином виконується обробка переривань?
- 9) Які два різновиди зовнішніх переривань є в мікроконтролерах сімейства Mega?
- 10) Які регістри використовуються для дозволу/заборони зовнішніх переривань?
- 11) Які регістри використовуються для індикації настання зовнішніх переривань?

ЛІТЕРАТУРА [1...10]

7 АНАЛОГОВИЙ КОМПАРАТОР

7.1 Загальні відомості

Компаратором називається електронний пристрій, призначений для порівняння двох сигналів (напруг). У залежності від форми представлення порівнюваних сигналів компаратори поділяються на:

- аналогові компаратори (АК);
- цифрові компаратори (ЦК) [19].

Нижче описуються АК, що призначені для порівняння двох аналогових напруг, одна з яких виконує функцію еталонної: $U_{\text{ЕТ}}$, а інша порівнюється з еталонною: $U_{\text{Х}}$.

Звичайно АК включає власне аналоговий компаратор (ВАК) і схему формування рівнів (СФР) (рисунок 7.1).

ВАК виконує порівняння двох напруг: $U_{\text{ЕТ}}$ і $U_{\text{Х}}$ і формує на виході дискретний сигнал, що приймає одне з двох значень: $+U_{\text{НАС}}$; $-U_{\text{НАС}}$ [18, 19].

СФР перетворює значення: $+U_{\text{НАС}}$; $-U_{\text{НАС}}$ у рівні цифрових сигналів ТТЛШ/КМОН – схем (ТТЛШ – транзисторно-транзисторна логіка з переходами Шоттки, КМОН – комплементарний метал-оксид-напівпровідник).

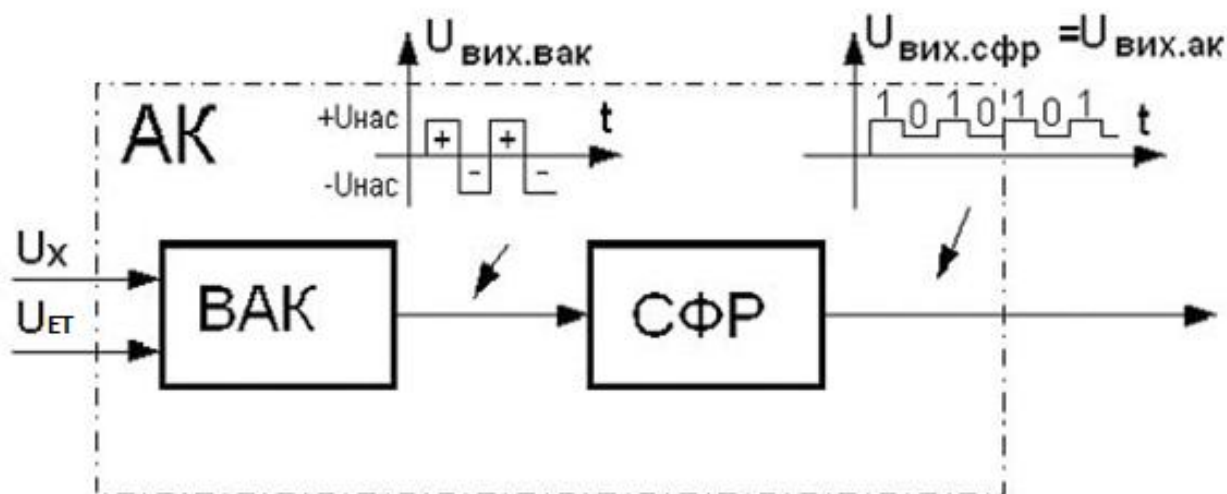


Рисунок 7.1 – Структура аналогового компаратора

Звичайно значення $+U_{\text{нас}}$ перетворюється в рівень логічної одиниці (1), а значення $(-U_{\text{нас}})$ – у рівень логічного нуля (0).

Найбільш широко застосовуються два варіанти схемного виконання АК:

- на інтегральній мікросхемі операційного підсилювача (ІМС ОП);
- на спеціалізованій мікросхемі аналогового компаратора.

7.2 Аналогові компаратори на інтегральній мікросхемі операційного підсилювача

7.2.1 Загальна характеристика інтегральної мікросхеми операційного підсилювача

Термін «операційний підсилювач» (ОП) вперше був використаний в обчислювальній техніці, де він ототожнювався із поняттям «розв’язуючий підсилювач». Із цим терміном незмінно пов’язувалась будь-яка математична операція: підсумовування, диференціювання, інтегрування і т. ін., яку підсилювач міг виконувати завдяки введенню від’ємного

зворотного зв'язку (ВЗЗ) певного вигляду. У даний час зміст цього терміну істотно розширився і дещо змінився. Під «операційним підсилювачем» прийнято розуміти універсальний електронний підсилювач, який може виконувати різні функції і дозволяє без порушення його працездатності вводити ЗЗ різного типу. Цей універсальний підсилювач існує у вигляді окремої інтегральної мікросхеми операційного підсилювача (ІМС ОП) або може бути частиною інтегральної мікросхеми. На рисунку 7.2 наведені два варіанти позначення ІМС ОП на електричних схемах.

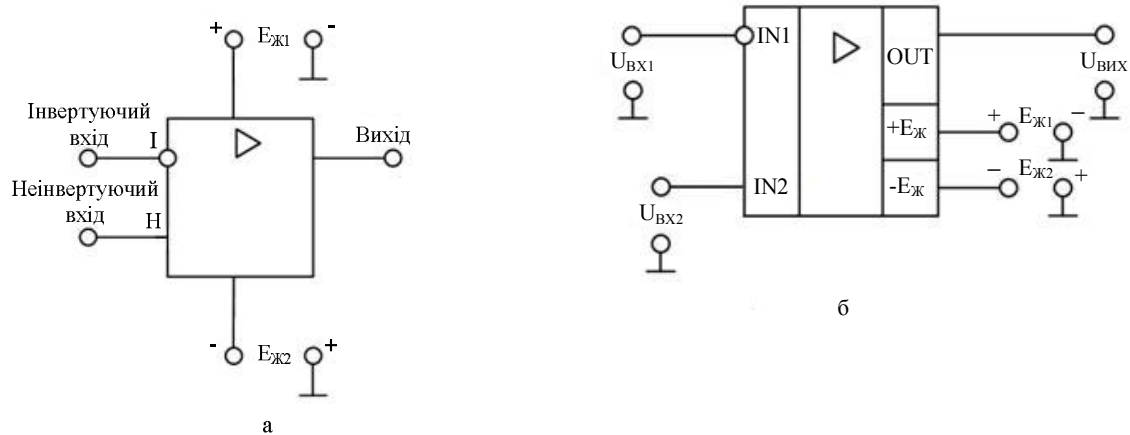


Рисунок 7.2 – Умовні позначення ІМС ОП на електричних схемах

У технічній документації ІМС ОП позначають прямокутником, у якого інвертуючий вхід позначають маленьким колом. За принципом дії ІМС ОП схожий із звичайним підсилювачем. Як і звичайний підсилювач, він призначений для підсилення напруги або потужності вхідного сигналу. Тоді як властивості і параметри звичайного підсилювача повністю визначені його схемою, властивості і параметри ІМС ОП визначаються переважно параметрами ланцюга ЗЗ. ІМС ОП виконують за схемою підсилювачів постійного струму із безпосереднім зв'язком між окремими каскадами із диференціальним входом і біполярним, по відношенню до амплітуди підсилювального сигналу, виходом. Це забезпечує нульові

потенціали на вході і виході ІМС ОП за відсутності сигналів, що керують. ІМС ОП характеризується великим коефіцієнтом підсилення, високим вхідним і низьким вихідним опорами. У даний час ІМС ОП виконуються, як правило, у вигляді монолітних інтегральних мікросхем, і за розмірами і ціною не відрізняються від окремо взятого транзистора. Завдяки практично ідеальним характеристикам ІМС ОП реалізація різних схем на їх основі виявляється значно простішою, ніж на окремих транзисторах. Тому ІМС ОП витісняють окремі транзистори як елементи схем у багатьох областях лінійної схемотехніки. Більшість ІМС ОП промислового типу мають один вихід (рисунк 7.2). При цьому вихідна напруга $U_{ВНХ}$ знаходиться у фазі із напругою $U_{ВХ2}$ і протифазі до напруги $U_{ВХ1}$. Напруга, що безпосередньо прикладена між входами, дорівнює різниці напруг $U_{ВХ1}$ і $U_{ВХ2}$. Ця напруга дорівнює нулю, якщо $U_{ВХ1}$ і $U_{ВХ2}$ мають навіть значні, але рівні значення. Тому напруги $U_{ВХ1}$ і $U_{ВХ2}$ по відношенню до спільної точки називаються напругою загального вигляду, а їх різниця - диференціальною напругою.

Вихідна напруга ОП обчислюється за формулою

$$U_{ВНХ} = \Delta U \cdot K_{U.ІМСОП}, \quad (7.1)$$

де ΔU — різниця напруг між входами, яка обчислюється за формулою

$$\Delta U' = U_{ВХ1} - U_{ВХ2}, \quad (7.2)$$

або

$$\Delta U'' = U_{ВХ2} - U_{ВХ1}, \quad (7.3)$$

$K_{U.ІМСОП}$ — коефіцієнт підсилення напруги інтегральної мікросхеми ОП (ІМС ОП).

Основною характеристикою ІМС ОП є передатна характеристика, яка наведена на рисунку 7.3.

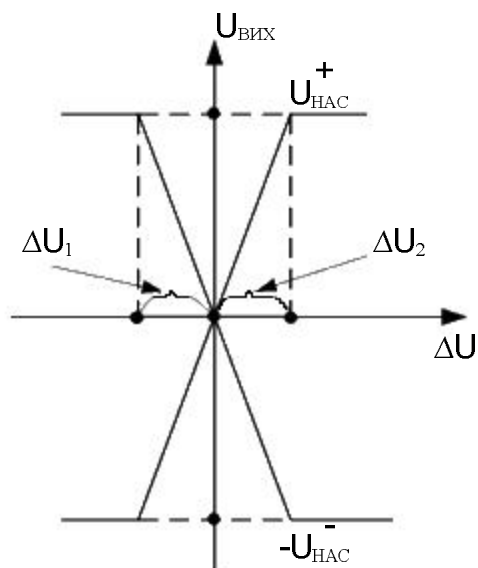


Рисунок 7.3 – Передатна характеристика ІМС ОП

На цьому рисунку суміщені дві передатні характеристики тієї ж самої ІМС ОП, вигляд яких залежить від того, як обчислюється ΔU :

$$\Delta U' = U_I - U_H,$$

$$\Delta U'' = U_H - U_I.$$

На рисунку 7.4 ці дві характеристики зображені окремо. Наведені на рисунках 7.3, 7.4 передатні характеристики ІМС ОП мають три характерні ділянки:

- лінійна ($U_{ВІХ}$ лінійно залежить від ΔU);
- насичення плюс ($U_{ВІХ} = +U_{НАС}^+$);
- насичення мінус ($U_{ВІХ} = -U_{НАС}^-$).

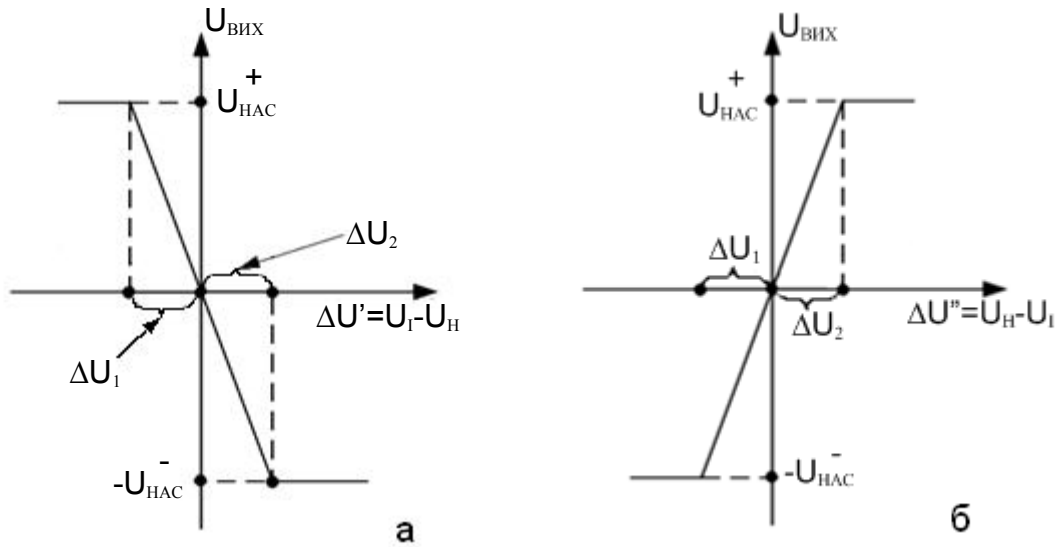


Рисунок 7.4 – Передатні характеристики ІМС ОП при:

$$а - \Delta U' = U_I - U_H; б - \Delta U'' = U_H - U_I$$

Значення напруг $|\Delta U_1| \approx \Delta U_2$, які відповідають границям лінійної ділянки, дуже малі

$$\Delta U_1 = \Delta U_2 = \frac{U_{НАС}^+}{K_{U.ІМСОП}}. \quad (7.4)$$

Умовні позначення на рисунках 7.3, 7.4:

$U_{ВНХ}^+$ - напруга насичення плюс, величина якої залежить від напруги живлення ІМС ОП (наприклад, якщо мікросхема живиться напругами $E_{Ж1} = +15B$, а $E_{Ж2} = -15B$, то $U_{НАС}^+ = +11B$, а $-U_{НАС}^- = -11B$);

$K_{U.ІМСОП}$ - коефіцієнт підсилення напруги ІМС ОП (дорівнює десяткам ... сотням тисяч). Наприклад, якщо

$$U_{НАС}^+ = 11B, \text{ а } K_{U.ІМСОП} = 10000, \text{ то } \Delta U_1 = \frac{11}{10000} = 1.1 \cdot 10^{-3} B = 1.1 мВ.$$

Лінійна ділянка передатної характеристики використовується у підсилювачах, а нелінійні ділянки - в імпульсних пристроях (компараторах, тригерах Шмітта, генераторах і т. ін.).

При використанні ІМС ОП в імпульсних пристроях вихідна напруга залежить від співвідношення потенціалів входів: І - інвертуючого на Н - неінвертуючого.

Якщо потенціал входу «І» більш додатний (менш від'ємний) відносно потенціалу на вході «Н» і різниця $\Delta U = (U_I - U_H) > (|U_1| = U_2)$, то $U_{ВИХ} = -U_{НАС}^-$.

Наприклад,

$U_I = 4В$, $U_H = 2В$, то $U_{ВИХ} = -U_{НАС}^-$; $U_I = -4В$, $U_H = -2В$, то $U_{ВИХ} = +U_{НАС}^+$.

Як правило

$$+U_{ВИХ}^+ = |-U_{НАС}^-|, \quad (7.5)$$

тому при посиленні на напруги насичення використовують, або $+U_{НАС}$, або $(-U_{НАС})$. Інтегральні мікросхеми операційних підсилювачів широко використовуються в сучасних електронних пристроях, тому що вони мають: високий коефіцієнт підсилення напруги (десятки ... сотні тисяч); великий вхідний опір (сотні ... тисячі кОм); малий вихідний опір (долі ... одиниці Ом); широку смугу частот пропускання (мегагерци); високий коефіцієнт придушення синфазного сигналу. При аналізі електронних схем на ІМС ОП останню вважають ідеальною, якщо:

$$\left. \begin{aligned} -K_{У.ІМСОП} &\rightarrow \infty; \\ -R_{ВХ} &\rightarrow \infty; \\ -R_{ВИХ} &\rightarrow 0; \\ -\Delta f_{ПРОПУСКАНИЯ} &\rightarrow \infty. \end{aligned} \right\} \quad (7.6)$$

На рисунку 7.5 наведена спрощена структурна схема ІМС ОП.

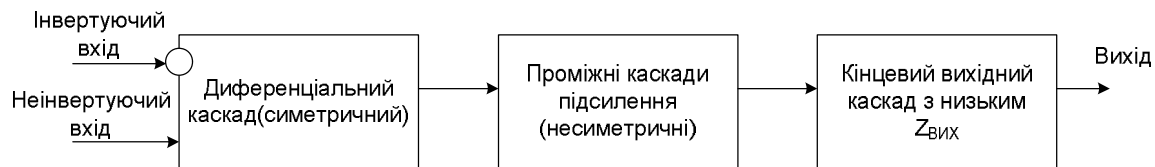


Рисунок 7.5 – Спрощена структура ІМС ОП

Найпростіший АК може бути виконано на ІМС ОП без зворотних зв'язків. Вигляд його схеми визначається полярністю порівнюваних напруг.

7.2.2 Аналогові компаратори для порівняння однополярних напруг на інтегральній мікросхемі операційного підсилювача

Схема АК, що виконує порівняння двох додатних напруг, наведена на рисунку 7.6, а. Одна напруга є еталонною ($U_{ET} = const$), а друга (U_X) – повільно змінюється за законом, представленим на рисунку 7.6, в.

До моменту часу $t = t_1$ напруга $U_X < U_{ET}$. Потенціал неінвертуючого входу, ІМС ОП більш додатний, ніж потенціал інвертуючого входу.

У цьому випадку різниця напруг між входами I та H

$$|-\Delta U'| = |(U_I - U_H)| > |-\Delta U_1|. \quad (7.7)$$

Відповідно до передатної характеристики ІМС ОП (рисунок 7.6, б) у цьому випадку $U_{ВИХ} = +U_{НАС}$. Після моменту часу $t = t_1$ вхідний сигнал $U_X \geq U_{ET}$. З'являється різниця напруг між входами

$$\Delta U'' = (U_I - U_H) > \Delta U_2. \quad (7.8)$$

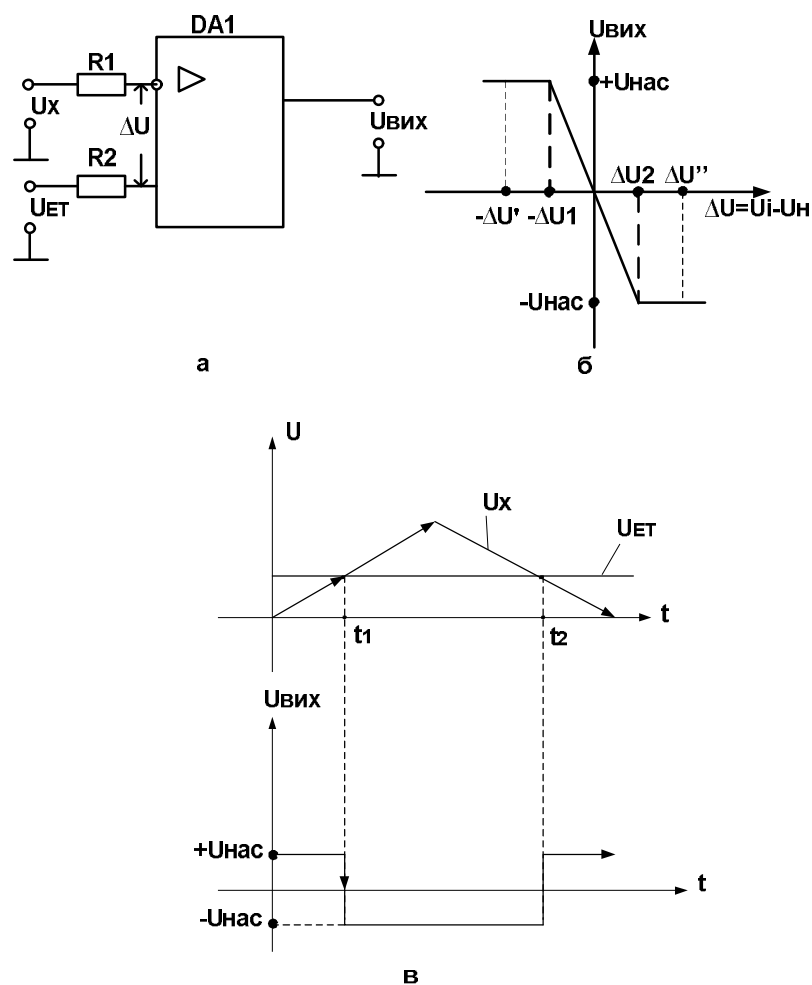


Рисунок 7.6 – АК для порівняння однополярних напруг: а – схема; б – передатна характеристика; в – часові діаграми

Відповідно до передатної характеристики ІМС ОП (рисунок 7.6, б) у цьому випадку $U_{\text{вих}} = -U_{\text{нас}}$.

На рисунку 7.6, в переключення АК відбувається миттєво, що є ідеальним випадком. Реально має місце невелика затримка між моментом досягнення рівності сигналів U_x та $U_{\text{ЕТ}}$ і моментом, коли вихідний сигнал $U_{\text{вих}}$ починає зменшуватися. Крім того, лінія, що відображає зменшення $U_{\text{вих}}$, йде не перпендикулярно осі часу, а під невеликим нахилом.

Після моменту часу $t = t_2$ напруга U_{ET} знову стає більшою ніж U_X , вихідна напруга змінюється і приймає значення: $+U_{НАС}$.

7.2.3 Аналогові компаратори для порівняння різнополярних напруг на інтегральній мікросхемі операційного підсилювача

Схема АК, що виконує порівняння різнополярних напруг, наведена на рисунку 7.7, а. Одна напруга ($U_{ET} = const$) є додатною, а друга (U_X – від’ємна) – повільно змінюється за законом, представленим на рисунку 7.7, б.

Значення резисторів схеми вибирається з умови

$$R1 = R2 = R; R3 = \frac{R}{2}. \quad (7.9)$$

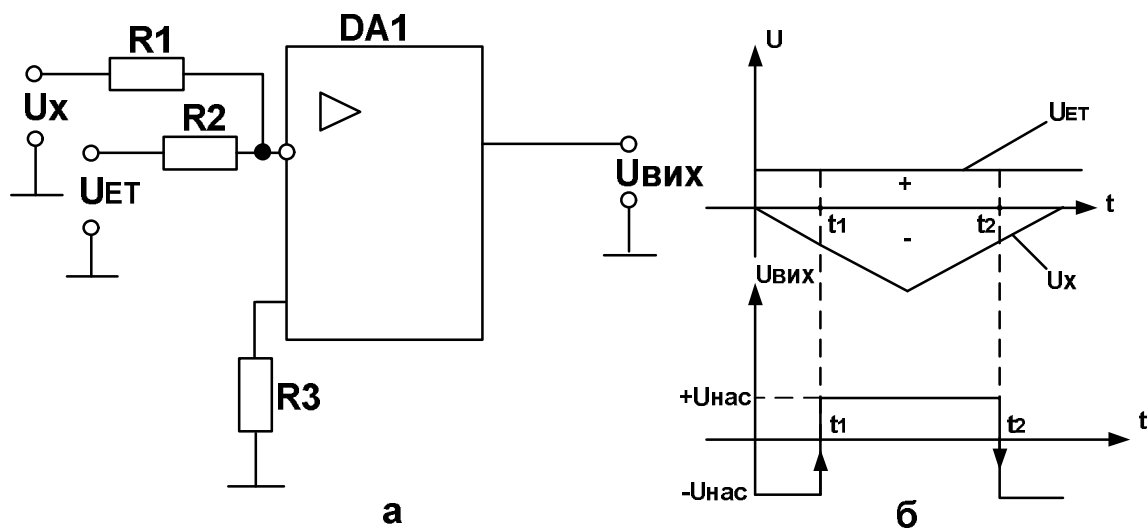


Рисунок 7.7 – АК для порівняння однополярних напруг:

а – схема; б – часові діаграми

До моменту часу $t = t_1$ (рисунок 7.7, б) $U_{\text{ЕТ}} > |-U_{\text{Х}}|$, тому потенціал інвертуючого входу, ІМС ОП ($\Delta U' > \Delta U_2$, рисунок 7.7, б) – додатний, а $U_{\text{ВИХ}} = -U_{\text{НАС}}$.

Коли $t \geq t_1$ (рисунок 7.7, б), то $|-U_{\text{Х}}| > U_{\text{ЕТ}}$. Потенціал інвертуючого входу ІМС ОП стає від'ємним ($|\Delta U'| > |\Delta U_1|$), а $U_{\text{ВИХ}} = +U_{\text{НАС}}$ (рисунок 7.7, б). Як і в попередньому випадку переключення АК (зміна $U_{\text{ВИХ}}$) відбувається з невеликою затримкою щодо моментів часу, коли $U_{\text{Х}}$ стає рівним $U_{\text{ЕТ}}$, а також зміна $U_{\text{ВИХ}}$ відбувається з невеликим нахилом до осі часу (на рисунках це не показано).

Якщо в схемах на рисунках 7.6, 7.7 $U_{\text{ЕТ}} = 0$, то переключення АК відбувається при $U_{\text{Х}} \approx 0$, а такий компаратор називають детектором нульового рівня.

7.3 АНАЛОГОВИЙ КОМПАРАТОР У СКЛАДІ AVR-МІКРОКОНТРОЛЕРІВ

7.3.1 Загальні відомості

Модуль аналогового компаратора входить до складу усіх без винятку мікроконтролерів сімейства Mega [1,2]. Будучи увімкненим, компаратор дає можливість порівнювати значення напруг, на відповідних двох виводах мікроконтролера. Результатом порівняння є логічне значення нуля або одиниця, що може бути прочитано з програми. За результатом порівняння може бути згенероване переривання, а також відбутися захоплення стану таймера/лічильника T1. Остання функція дозволяє, зокрема, вимірювати тривалості аналогових сигналів. Виводи, що використовуються компаратором, є контактами портів введення/виведення загального

призначення. В таблиці 7.1 наведені такі виводи окремих мікроконтролерів сімейства Mega.

Таблиця 7.1 – Виводи, що використовуються аналоговим компаратором

Назва	ATmega8x	ATmega8515x/8535x	ATmega16x/32x	ATmega64x/128x/165x	ATmega162x/163x	ATmega48x/88x/168x	ATmega164x/324x/644x	ATmega325x/3250x/645x/6450x	ATmega640x/1280x/1281x/2560x/2561x	Призначення
AIN0	PD6	PB2	PB2	PE2	PB2	PD6	PB2	PE2	PE2	Неінвертуючий вхід
AIN1	PD7	PB3	PB3	PB3	PB3	PD7	PB3	PB3	PE3	Інвертуючий вхід

Щоб зазначені виводи могли використовуватися аналоговим компаратором, вони повинні бути сконфігуровані як входи (відповідний розряд регістра DDRx скинутий у "0"). Крім того, необхідно відключити внутрішні підтягуючі резистори записом лог. 0 у відповідний розряд регістра PORTx.

В моделях ATmega48x/88x/168x, ATmega165x, Atmega325x/3250x/645x/6450x та ATmega640x/1280x/1281x/2560x/2561x є можливість відключення вхідних цифрових буферів у випадку, якщо контакти мікроконтролера, що відповідають виводам AIN0 та AIN1, використовується тільки для введення аналогових сигналів. При відключених цифрових буферах зменшується загальний струм споживання мікроконтролера, а відповідні розряди регістра PINx завжди читаються як 0.

Відключення цифрових буферів на входах AIN0 та AIN1 здійснюється записом лог. 1 відповідно в розряди AIN0D та AIN1D регістра DIDR1, розташованого за адресою (\$7F). Формат цього регістра наведено на рисунку 7.8.

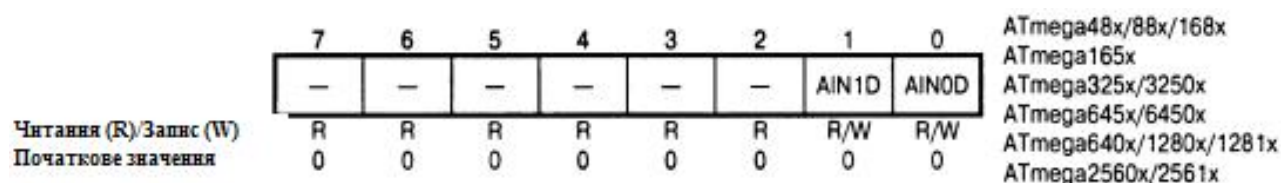


Рисунок 7.8 – Формат регістра DIDR1

7.3.2 Функціонування компаратора

Структурна схема аналогового компаратора приведена на рисунку 7.9. Керування компаратором і контроль його стану здійснюється за допомогою регістра ACSR, який в залежності від моделі розташований за адресою \$08 (\$28) або \$30(\$50) (таблиця 7.2). Формат цього регістра приведено на рисунку 7.10, а призначення його розрядів коротко описано в таблиці 7.3.

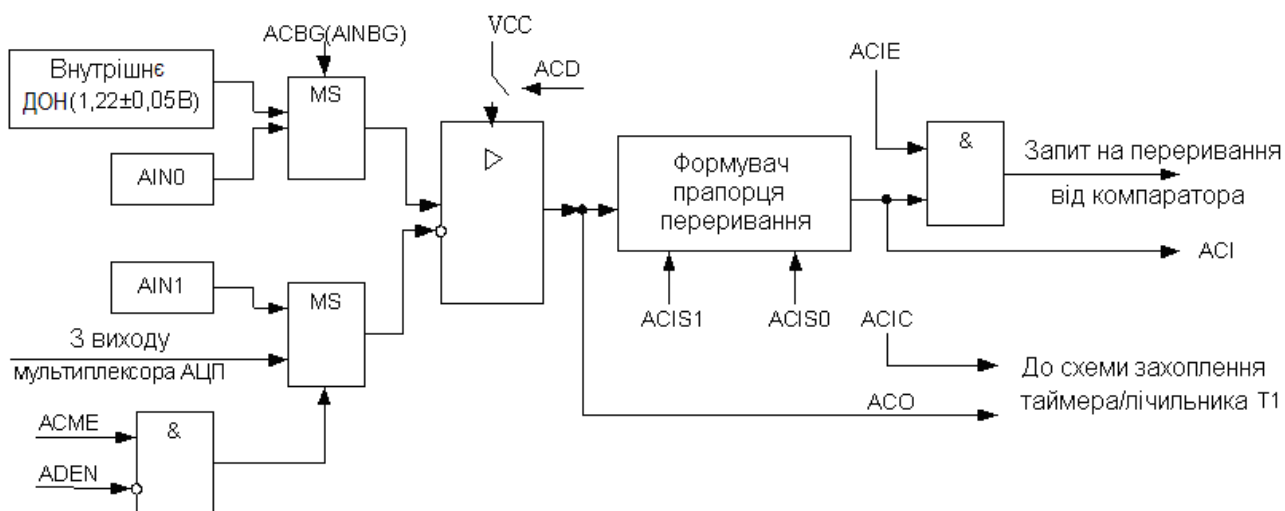


Рисунок 7.9 – Структурна схема аналогового компаратора

Таблиця 7.2 – Розміщення регістра ACSR

Модель	Адреса регістра
ATmega8515x/8535x	\$08 (\$28)
ATmega8x/16x/32x/64x/128x	
ATmega48x/88x/168x	\$30 (\$50)
ATmega162x	\$08 (\$28)
ATmega164x/324x/644x	\$30 (\$50)
ATmega165x	
ATmega325x/3250x/645x/6450x	
ATmega640x/1280x/1281x/2560x/2561x	

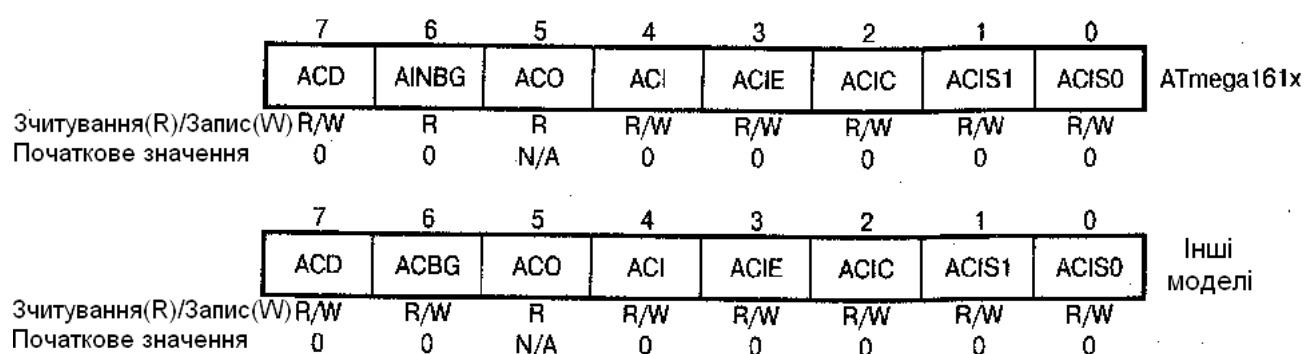


Рисунок 7.10 – Формат регістра ACSR

За своєю дією розглянутий вузол мікроконтролера є звичайним компаратором. Якщо напруга на виводі AIN0 (неінвертуючий вхід) більш додатна (менш від’ємна) напруги на виводі AIN1 (інвертуючий вхід), то результат порівняння буде дорівнювати "1". У іншому випадку результат порівняння буде дорівнювати "0". Цей результат (стан виходу компаратора) зберігається в розряді ACO регістра ACSR. Розряд ACD відповідає за вмикання і вимикання компаратора.

Таблиця 7.3 – Розряди регістра ACSR

Розряд	Назва	Опис
7	ACD	Вимикання компаратора ("0" - ввімкнено, "1" - вимкнено)
6	ACBG (AINBG*)	Підключення до неінвертуючого входу компаратора внутрішнього ДОН ("0" - не підключено, "1" - підключено)
5	ACO	Результат порівняння (вихід компаратора)
4	ACI	Прапорець переривання від компаратора
4	ACIE	Дозвіл переривання від компаратора
2	ACIC	Підключення компаратора до схеми захоплення таймера/лічильника T1 ("1" - підключено, "0"- не підключено)
1, 0	ACIS1:ACIS0	Умова виникнення переривання від компаратора
*В моделях ATmega161x		

Оскільки при подачі напруги живлення всі розряди регістра ACSR скидаються в "0", компаратор вмикається автоматично при ввімкненні мікроконтролера. Для вимикання компаратора розряд ACD слід встановити в "1". При зміні стану цього розряду переривання від компаратора слід заборонити. Як уже було сказано, за результатом порівняння схема компаратора може генерувати запит на переривання. Якщо стан виходу компаратора (розряд ACO) змінився заданим чином, встановлюється прапорець переривання ACIF регістра ACSR і генерується запит на переривання, якщо воно дозволено. Як і для інших переривань, цей прапорець скидається апаратно при запуску підпрограми обробки переривання або програмно, записом у нього лог. 1. Для дозволу переривання необхідно встановити в "1" розряд ACI регістра ACSR і, зрозуміло, прапорець I регістра SREG. Яка саме зміна стану виходу компаратора викликає переривання, визначається станом розрядів

ACIS1:ACIS0 регістра ACSR відповідно до таблиці 7.4. При зміні цих розрядів переривання від компаратора (як і для розряду ACD) повинно бути заборонено.

Таблиця 7.4 – Умови генерації запиту на переривання від компаратора

ACIS1	ACIS0	Умова
0	0	Будь-яка зміна стану виходу компаратора
0	1	Зарезервовано
1	0	Зміна стану виходу компаратора з "1" на "0"
1	1	Зміна стану виходу компаратора з "0" на "1"

Крім генерації переривання, компаратор також може керувати схемою захоплення таймера/лічильника T1. Для цього необхідно встановити в "1" розряд ACIC регістра ACSR. В результаті вихід компаратора підключиться до схеми захоплення замість виводу ICP1 мікроконтролера. Якщо ж розряд ACIC скинуто у "0", компаратор повністю відключений від блоку захоплення таймера/лічильника T1. Компаратор може порівнювати сигнали, не тільки на виводах AIN0 і AIN1. Так, замість виводу AIN0 мікроконтролера до неінвертуючого входу компаратора може бути підключене внутрішнє джерело опорної напруги (ДОН) величиною 1.22 (± 0.05) В. Для цього необхідно встановити в "1" розряд ACBG (AINBG у моделях ATmega161x) регістра ACSR.

Причому, якщо схема BOD вимкнена (BODEN = "1"), то між підключенням ДОН до компаратора і початком використання компаратора необхідно зачекати деякий час (не більше 100 мкс). На інвертуючий вхід компаратора може також надходити сигнал з виходу мультиплексора модуля АЦП. Іншими словами, замість виводу AIN1 мікроконтролера інвертуючий вхід компаратора може бути підключено до кожного з входів АЦП ADC0...ADC7 (ADC0...ADC15 у моделях

ATmega640x/1280x/2560x/2561x). Відповідно, у моделях ATmega8515x, ATmega161x і ATmega162x, що не мають модуля АЦП, ця можливість недоступна.

Підключення виходу мультиплексора АЦП до входу компаратора здійснюється встановленням в 1 розряду АСМЕ, який розташований, в залежності від моделі (таблиця 7.5), або в регістрі спеціальних функцій SFIOR (3-й розряд регістра), або в регістрі керування АЦП ADCSRB (6-й розряд). Зрозуміло, модуль АЦП при цьому має бути вимкнено (розряд ADEN регістра ADCSRA скинуто в 0). Який саме з входів АЦП буде використовуватись в якості інвертуючого входу компаратора, визначається розрядами MUX2...0 регістра ADMUX, як показано в таблиці 7.6 (в моделях ATmega640x/1280x/1281x/2560x/2561x використовується також розряд MUX5 регістра ADCSRB).

Таблиця 7.5 – Підключення мультиплексора АЦП до компаратора

Модель	Регістр	Адреса
ATmega8515x/8535x	SFIOR	\$30 (\$50)
ATmega8x/16x/32x/64x/128x		
ATmega48x/88x/168x		\$20 (\$40)
ATmega162x	ADCSRB	(\$7B)
ATmega164x/324x/644x		
ATmega165x		
ATmega325x/3250x/645x/6450x		
ATmega640x/1280x/1281x/2560x/2561x		

Таблиця 7.6 – Керування інвертуючим входом компаратора

ACME	ADEN	MUX5 ²⁾	MUX2..0	Інвертуючий вхід компаратора
0	x	x	xxx	AIN1
1	1	x	xxx	AIN1
1	0	0	001	ADC1
1	0	0	010	ADC2
1	0	0	011	ADC3
1	0	0	100	ADC4
1	0	0	101	ADC5
1	0	0	110	ADC6 ¹⁾
1	0	0	111	ADC7 ¹⁾
1	0	1	000	ADC8 ²⁾
1	0	1	001	ADC9 ²⁾
1	0	1	010	ADC10 ²⁾
1	0	1	011	ADC11 ²⁾
1	0	1	100	ADC12 ²⁾
1	0	1	101	ADC13 ²⁾
1	0	1	110	ADC14 ²⁾
1	0	1	111	ADC15 ²⁾
¹⁾ В моделях ATmega8x та ATmega48x/88x/168x – тільки в корпусах TQFP та MLF.				
²⁾ Тільки в моделях ATmega640x/1280x/2560x.				

Детально регістри ADCSRx та ADMUX розглянуто у лекції, яка присвячена модулю АЦП.

Основні параметри аналогового компаратора представлено в [2].

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Наведіть та поясніть структуру модуля аналогового компаратора (АК).
- 2) Наведіть та поясніть формати керуючих регістрів модуля (АК).
- 3) Яким чином можна підключити мультиплексор АЦП до компаратора.
- 4) Поясніть роботу АК, який виконано на ІМС ОП.

ЛІТЕРАТУРА [1...10]

8 АНАЛОГО-ЦИФРОВИЙ ПЕРЕТВОРЮВАЧ (АЦП)

8.1 Загальні особливості модуля АЦП

Модуль 10-розрядного АЦП послідовного наближення входить до складу моделей сімейства Mega, за винятком ATmega8515x/162x. Основні параметри модуля АЦП наступні:

- абсолютна похибка: ± 2 МЗР (молодшого значущого розряду);
- інтегральна нелінійність: ± 0.5 МЗР¹;
- швидкодія: до 15 тис. вибірок/с [2,20].

На вході модуля АЦП моделей є 8-канальний (в моделях ATmega640x/1280x/2560x – 16-канальний) аналоговий мультиплексор, що надає в розпорядження користувача 16 каналів з однополярними (несиметричними) входами. Окремо варто сказати про мікроконтролери ATmega8x. Мікроконтролери цієї групи, що випускають у корпусі DIP-32, мають тільки 6 каналів перетворення. Крім того, у всіх моделях ATmega8x два канали (ADC4 і ADC5) є 8-розрядними.

У більшості моделей входи АЦП можуть також поєднуватися попарно для формування різного числа каналів з диференціальним входом.

При цьому в деяких каналах є можливість 10- і 200-кратного попереднього підсилення вхідного сигналу. При коефіцієнтах підсилення 1x та 10x діюча роздільна здатність АЦП складає 8 розрядів, а при коефіцієнті підсилення 200x – 7 розрядів.

¹ МЗР в зарубіжній літературі позначається LSB (Least Significant Bit – молодший значущий розряд), аналогічно, MSB (Most Significant Bit – старший значущий розряд) відповідає позначенню СЗР.

Як джерело опорної напруги для АЦП може використовуватись як напруга живлення мікроконтролера, так і внутрішнє або зовнішнє джерело опорної напруги.

Модуль АЦП може функціонувати у двох режимах:

- режим одиночного перетворення, коли запуск кожного перетворення ініціюється користувачем;
- режим безперервного перетворення, коли запуск перетворень виконується безперервно через певні інтервали часу.

Модуль АЦП містить ПВЗ (пристрій вибірки-зберігання), що підтримує під час перетворення напругу безпосередньо на вході АЦП на постійному рівні.

8.2 Функціонування модуля АЦП

Функціональна схема модуля АЦП наведено на рисунку 8.1. У моделях ATmega8x і ATmega48x/88x/168x елементи і пов'язані з ними сигнали, які виділено на рисунку сірим кольором, відсутні, а неінвертуючий вхід компаратора з пристроєм вибірки-збереження підключений безпосередньо до виходу мультиплексора (показано пунктирною лінією).

Вхідний аналоговий сигнал зберігається в ПВЗ. За командою «Старт» за допомогою регістра зсуву (логіка перетворення) послідовно у часі кожен розряд ЦАП, починаючи зі старшого розряду, переводиться у положення 1. Компаратор зрівнює напругу на виході ЦАП з вхідною напругою: якщо напруга на вході більша, ніж напруга на виході ЦАП, то на виході компаратора високий рівень, а в тригер регістра зсуву записується 1, якщо напруга на вході менша, ніж напруга на виході ЦАП, то тригер переводиться у положення 0. В кінці перетворення у регістрі

даних АЦП (ADCH/ADCL) буде двійковий код, який еквівалентний вхідній напрузі.

АЦП працює у двох режимах: однополярному та диференціальному. При однополярному режимі вхідний сигнал поступає на один із входів ADC0-ADC15. При диференціальному режимі використовуються шість входів мультиплексора (ADC0, ADC1, ADC2, ADC8, ADC9, ADC10). Різниця сигналу між цими входами підсилюється за допомогою диференціального підсилювача і поступає на вхід компаратора.

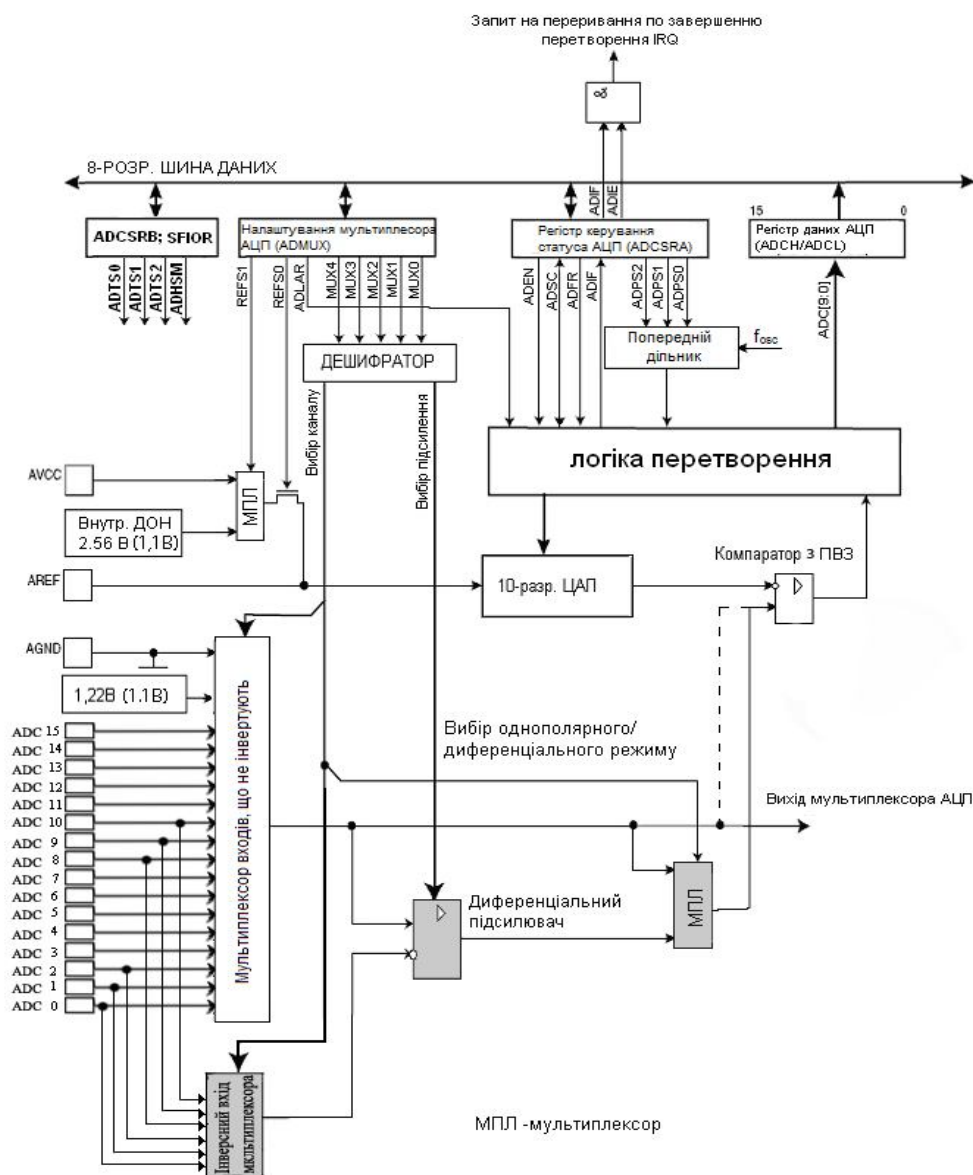


Рисунок 8.1 – Функціональна схема аналого-цифрового перетворювача

8.3 Програмування модуля АЦП

Як приклад для деяких моделей сімейства AVR у таблиці 8.1 наведено реєстри, що використовуються для керування модулем АЦП. Формат реєстрів ADCSRA (ADCSR) і ADMUX наведено на рисунках 8.2 і 8.3, а короткий опис функцій їх розрядів наведено у таблицях 8.2 і 8.3 відповідно.

Таблиця 8.1– Реєстри керування модулем АЦП

Регістр	Адреса	ATmega8535x	ATmega8x	ATmega16x\32x	ATmega163x	ATmega323x	ATmega48x\88x\168x	ATmega64x	ATmega164x\324x\644x	ATmega165x	ATmega325x/3250x, ATmega645x/6450x	ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x	ATmega128x	Опис
ADCSR	\$06(\$26)		♦		♦	♦								Регістр керування і стану
ADCSR A	\$06 (\$26)	♦		♦				♦					♦	Регістр керування і стану А
	(\$7A)						♦		♦	♦	♦	♦		
ADCSR B	(\$8E)							♦						Регістр керування і стану В
	(\$7B)						♦		♦	♦	♦	♦		
ADMUX	\$07 (\$27)	♦	♦	♦	♦	♦		♦					♦	Регістр керування мультиплексором
	(7C)						♦		♦	♦	♦	♦		
SFOR	\$30 (\$50)	♦	♦	♦										Регістр спеціальних функцій
	\$20 (\$40)												♦	

Таблиця 8.2 – Розряди регістра ADCSRA (ADCSR*)

Розряд	Назва	Опис
7	ADEN	Дозвіл АЦП (1 - увімкнено, 0 - вимкнено)
6	ADSC	Запуск перетворення (1 - почати перетворення)
5	ADATE (ADFR **)	Вибір режиму роботи АЦП (0 - одиночне, 1 - безперервне перетворення)
4	ADIF	Прапорець переривання завершення АЦП
3	ADIE	Дозвіл переривання від завершення АЦП
2..0	ADPS2:ADPS0	Вибір частоти перетворення (див. таблицю 8.6)
* В моделі ATmega8x		
** В моделях ATmega8x, ATmega128x		

Таблиця 8.3– Розряди регістра ADMUX

Розряд	Назва	Опис	Модель
7..6	REFS1:REFS0	Вибір джерела опорної напруги (див. таблицю 8.5)	Всі моделі
5	ADLAR	Ліве вирівнювання результату (див. таблицю 8.11)	Всі моделі
4	-	Зарезервовано	ATmega8x
	MUX4	Вибір входних каналів (див. таблицю 8.9)	Всі моделі крім ATmega8x
3..0	MUX3..MUX0	Вибір входних каналів і частоти перетворення (див. таблиці 8.8, 8.9)	Всі моделі

	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ATmega8x ATmega128x
Зчитування(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	
	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	Інші моделі
Зчитування(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	
ATmega 8x	– ADCSR								
Інші моделі	– ADCSRA								

Рисунок 8.2 – Формат регістра ADCSRA (ADCSR)

	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ATmega8x ATmega48x/88x/168x
Зчитування(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	
	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	Інші моделі
Зчитування(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 8.3 – Формат регістра ADMUX

Формат регістрів ADCSRB і SFIOR наведено рисунках 8.4 і 8.5 відповідно (не використовувані розряди регістра SFIOR позначені як «-»).

Для дозволу робити АЦП необхідно записати лог. 1 в розряд ADEN регістра ADCSRA (ADCSR), а для заборони – відповідно лог. 0. Якщо АЦП буде вимкнено під час циклу перетворення, то перетворення не буде завершено (в регістрі даних АЦП залишиться результат попереднього перетворення).

Більшість моделей має можливість використовувати модуль АЦП в якості мультиплексора аналогових сигналів для аналогового компаратора, для цього необхідно встановити розряд ACME (Analog Comparator Multiplexer Enable) в лог. 1 (див. розділ 7). Інверсний вхід компаратора може бути з'єднано, як з його входом AIN1 так і з будь-яким із входів АЦП в залежності від значення розряду ADEN (лог. 1 – AIN1, лог. 0 – один із входів АЦП).

У моделях ATmega8x та ATmega128x режим роботи АЦП визначається станом розряду ADFR.

Якщо його встановлено в "1", АЦП працює в режимі безперервного перетворення. У цьому режимі запуск кожного наступного перетворення здійснюється автоматично після закінчення поточного.

Якщо ж розряд ADFR скинуто в "0", АЦП працює в режимі одиночного перетворення й запуск кожного перетворення здійснюється командою користувача.

	7	6	5	4	3	2	1	0	
	-	-	-	-	-	ADTS2	ADTS1	ADTS0	ATmega64x
Зчитування(R)/Запис(W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega48x/88x/168x ATmega164x/324x/644x ATmega165x/325x/3250x ATmega645x/6450x ATmega1281x/2561x
	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0	
Зчитування(R)/Запис(W)	R	R/W	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	ATmega640x/1280x/2560x
	-	ACME	-	-	MUX5	ADTS2	ADTS1	ADTS0	
Зчитування(R)/Запис(W)	R	R/W	R	R	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

Рисунок 8.4 – Формат регістра ADCSRB

	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	-	X	X	X	X	ATmega8535x ATmega16x/32x
Зчитування(R)/Запис(W)	R	R	R	R	R	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

Рисунок 8.5 – Формат регістра SFIOR

У всіх моделях, окрім ATmega8x та ATmega128x, запуск АЦП можливий не тільки командою користувача, але й перериванням від деяких периферійних пристроїв, наявних у складі мікроконтролера. Для вибору режиму роботи в цих моделях використовується розряд ADATE регістра ADCSRA і розряди ADTS2...0 регістра SFIOR або ADCSRB.

Якщо розряд ADATE скинуто в "0", АЦП працює в режимі одиночного перетворення. Якщо ж розряд ADATE встановлено в "1",

функціонування АЦП визначається вмістом розрядів ADTS2...0 відповідно до таблиці 8.4.

Таблиця 8.4 – Джерело сигналу для запуску перетворення

ADTS2	ADTS1	ADTS0	Джерело стартового сигналу
0	0	0	Режим безперервного перетворення
0	0	1	Переривання від аналогового компаратора
0	1	0	Зовнішнє переривання INT0
0	1	1	Переривання за подією "Збіг А" таймера/лічильника T0
1	0	0	Переривання за переповненням таймера/лічильника T0
1	0	1	Переривання за подією "Збіг В" таймера/лічильника T1
1	1	0	Переривання за переповненням таймера/лічильника T1
1	1	1	Переривання за подією "Захоплення" таймера/лічильника T1

Запуск кожного перетворення в режимі одиночного перетворення, а також запуск першого перетворення в режимі безперервного перетворення здійснюється встановленням в "1" розряду ADSC регістра ADCSRA (ADCSR).

Запуск перетворення за перериванням здійснюється при установці в "1" прапорця обраного переривання. Розряд ADSC регістра ADCSRA при цьому апаратно встановлюється в "1". Запуск перетворення в цих режимах також може бути здійснений встановленням в "1" розряду ADSC регістра ADCSRA.

Модуль АЦП може використовувати різні джерела опорної напруги (ДОН). Вибір конкретного джерела опорної напруги здійснюється за допомогою розрядів REFS1:REFS0 регістра ADMUX (таблиця 8.5).

Як зазначено в таблиці 8.5, внутрішнє ДОН може бути підключене до виводу AREF мікроконтролера. Тому при його використанні для підвищення завадозахищеності до виводу AREF можна підключити зовнішній фільтруючий конденсатор.

Таблиця 8.5– Вибір джерела опорної напруги

REFS1	REFS0	Джерело опорної напруги ¹⁾	Модель
0	0	Зовнішнє ДОН, підключено до виводу AREF; внутрішнє ДОН відключено	Всі моделі
0	1	Напруга живлення AVcc ²⁾	Всі моделі
1	0	Внутрішнє ДОН напругою 1.1 В ²⁾	ATmega164x/324x/644x ATmega640x/1280x/1281x ATmega2560x/2561x
		Зарезервовано	Решта моделей
1	1	Внутрішнє ДОН напругою 2.56 В ²⁾	ATmega48x/88x/168x ATmega165/325x/3250x ATmega645x/6450x
		Внутрішнє ДОН напругою 1.1 В ²⁾	Решта моделей

¹⁾ При роботі з підсиленням 10х чи 200х в якості внутрішнього ДОН можна використовувати тільки ДОН яке, підключено при REFS1:0=11.

²⁾ Якщо до виводу AREF підключено зовнішнє джерело напруги, дані варіанти використовуватись не можуть.

АЦП перетворює вхідну аналогову напругу в 10-розрядний код методом послідовного наближення. Мінімальне значення відповідає рівню GND, а максимальне рівню AREF мінус 1 мол. розряду. До виводу AREF опціонально може бути підключено напругу AVCC або внутрішнє ДОН на 2,56 (1,1) В шляхом запису відповідних значень у розряди REFSn у регістр ADMUX. Незважаючи на те, що ДОН на 2,56 (1,1) В перебуває усередині мікроконтролера, для зниження чутливості до шумів до його виходу може бути підключено фільтруючий конденсатор, тому що він пов'язаний з виводом AREF.

Канал однополярного або диференціального аналогового введення та каскад диференціального підсилення обираються шляхом програмування розрядів MUXn ($n=0,1\dots5$) у регістрі ADMUX (таблиці 8.9, 8.10). У якості однополярного аналогового входу АЦП в залежності від моделі мікроконтролера може бути обраний один із входів ADC0...ADC15, а також GND і вихід фіксованого внутрішнього джерела опорної напруги 1,22 (1,1) В. У режимі диференціального введення передбачена можливість вибору входів, що інвертують і не інвертують диференціального підсилювача.

Якщо обрано диференціальний режим аналогового введення, то диференціальний підсилювач буде помножувати різницю напруг між обраною парою входів на заданий коефіцієнт підсилення. Підсиленне в такий спосіб значення надходить на аналоговий вхід АЦП. Якщо обирається однополярний режим аналогового введення, то каскад підсилення пропускається.

Робота АЦП дозволяється шляхом встановлення розряду ADEN у регістрі ADCSRA. Вибір опорного джерела та каналу перетворення неможливо виконати до встановлення ADEN. Якщо ADEN = 0, то АЦП не споживає струм, тому при переході в економічні режими сну рекомендується попередньо відключити АЦП.

АЦП генерує 10-розрядний результат, що міститься в парі регістрів даних АЦП: ADCH і ADCL. У початковому стані результат перетворення розміщується в молодших 10-ти розрядах 16-розрядного слова (вирівнювання вправо), але може бути розміщений у старших 10-ти розрядах (вирівнювання вліво) шляхом встановлення розряду ADLAR у регістрі ADMUX (таблиці 8.3, 8.11).

Практична корисність подання результату з вирівнюванням вліво існує, коли досить точності 8-розрядного значення. В цьому випадку необхідно читати тільки регістр ADCH. В іншому ж випадку необхідно

першим читати вміст регістра ADCL, а потім ADCH, чим гарантується, що обидва байти є результатом того самого перетворення. Як тільки виконано читання ADCL блокується доступ до регістрів даних з боку АЦП. Це означає, що якщо зчитаний ADCL і перетворення завершується перед читанням регістра ADCH, то жоден з регістрів не може модифікуватися й результат перетворення губиться. Після читання ADCH доступ до регістрів ADCH і ADCL з боку АЦП знову дозволяється.

АЦП генерує власний запит на переривання за завершенням перетворення. Якщо між читанням регістрів ADCH і ADCL доступ до даних для АЦП заборонено, то переривання виникне, навіть якщо результат перетворення буде загублено.

Одиночне перетворення запускається шляхом запису лог. 1 у розряд запуску перетворення АЦП ADSC. Даний розряд залишається у високому стані в процесі перетворення й скидається за завершенням перетворення. Якщо в процесі перетворення перемикається канал аналогового введення, то АЦП автоматично завершить поточне перетворення, перш ніж перемкне канал.

У режимі автоматичного перезапуску АЦП безупинно оцифровує аналоговий сигнал і оновлює регістр даних АЦП. Даний режим задається шляхом запису лог. 1 у розряд ADFR (ADATE) регістра ADCSR (ADCSRA). Перше перетворення ініціюється шляхом запису лог. 1 у розряд ADSC регістра ADCSR (ADCSRA). У даному режимі АЦП виконує послідовні перетворення, незалежно від того скидається прапорець переривання АЦП ADIF чи ні.

8.4 Формування тактової частоти АЦП

На рисунку 8.5 наведено схему попереднього дільника, що формує тактовий сигнал для АЦП. Попередній дільник формує похідні частоти

відносно частоти синхронізації мікроконтролера. Коефіцієнт ділення встановлюється за допомогою розрядів ADPSn у регістрі ADCSRA (таблиця 8.6).

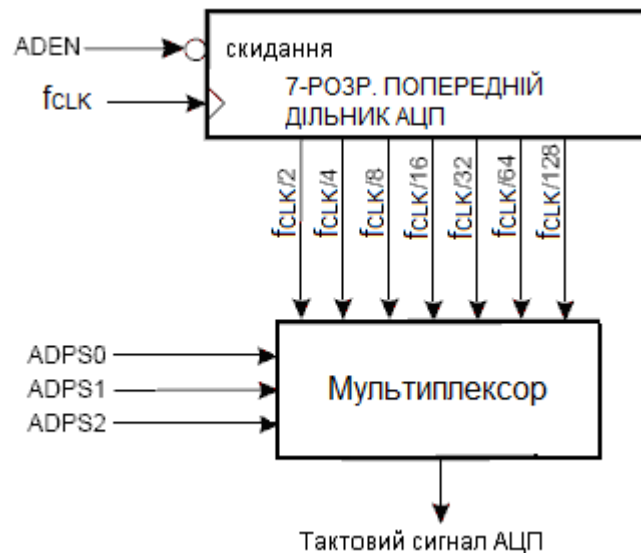


Рисунок 8.5 – Схема попереднього дільника АЦП

Попередній дільник починає лічбу з моменту включення АЦП встановленням розряду ADEN у регістрі ADCSRA. Попередній дільник працює доки розряд ADEN = 1 і скинутий, коли ADEN = 0.

Якщо потрібно забезпечити максимальну роздільну здатність (10 розрядів), то частота на вході схеми послідовного наближення повинна бути в діапазоні 50...200 кГц [2]. Якщо достатньо точності менше 10 розрядів, але потрібна більш висока частота перетворення, то частота на вході АЦП може бути встановлена понад 200 кГц.

Таблиця 8.6 - Задання коефіцієнта ділення попереднього дільника АЦП

ADPS2	ADPS1	ADPS0	Коефіцієнт ділення
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

8.5 Часові діаграми роботи АЦП

На рисунках 8.6...8.9 наведено часові діаграми роботи АЦП у різних режимах.

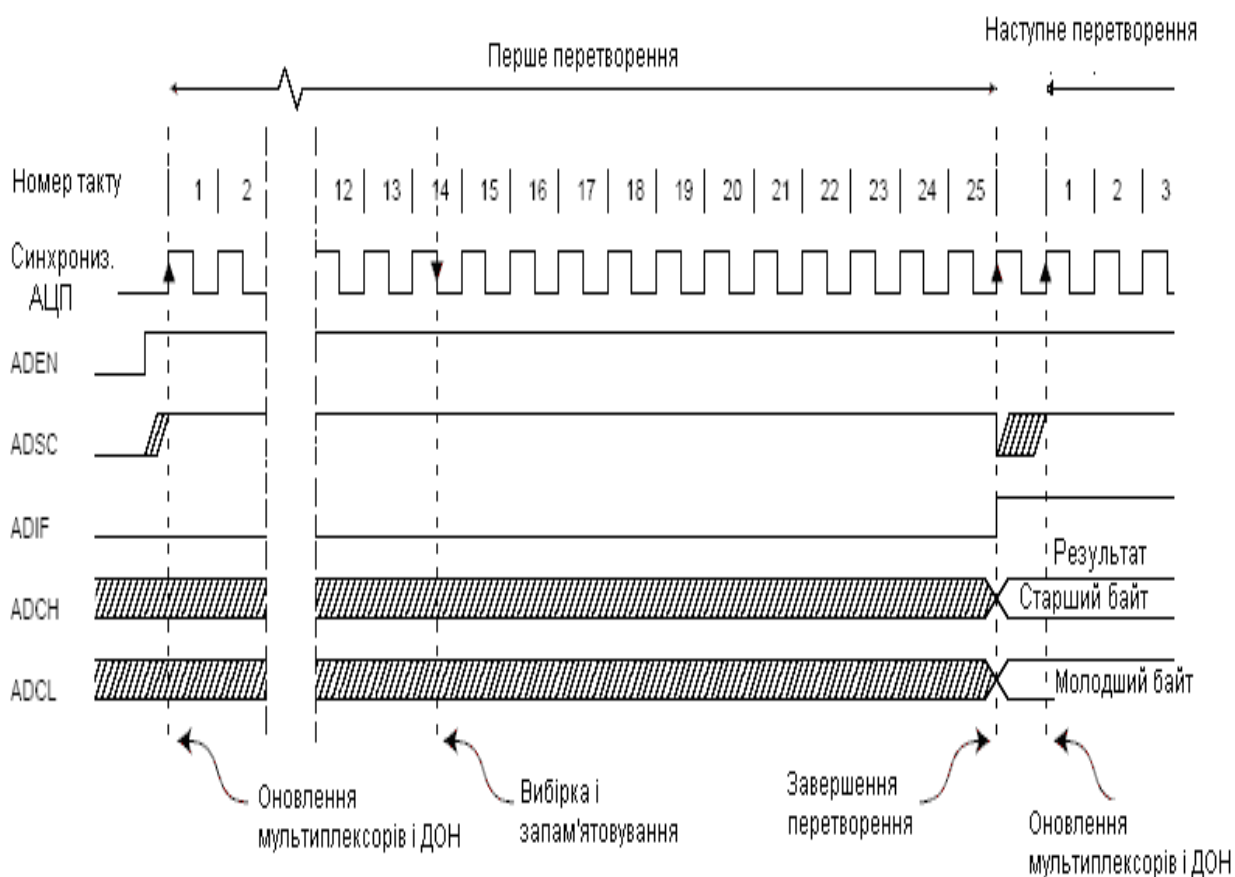


Рисунок 8.6 – Часові діаграми роботи АЦП при першому перетворенні в режимі одиночного перетворення

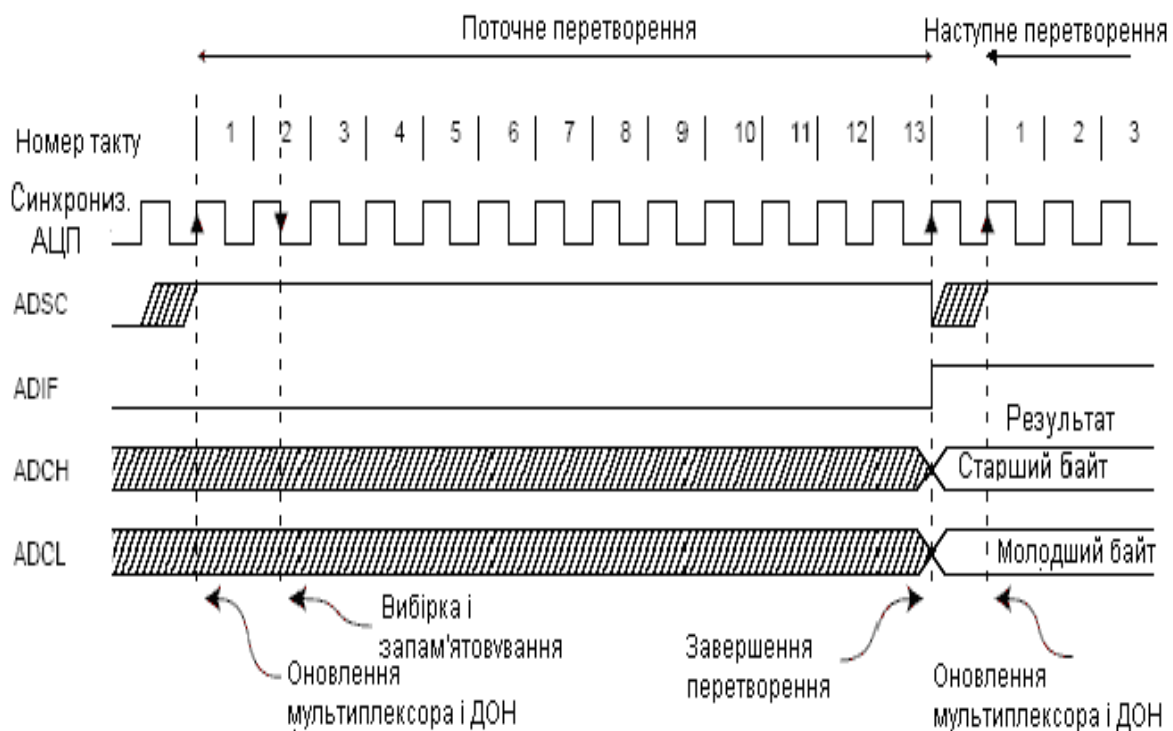


Рисунок 8.7 – Часові діаграми роботи АЦП у режимі одиночного перетворення

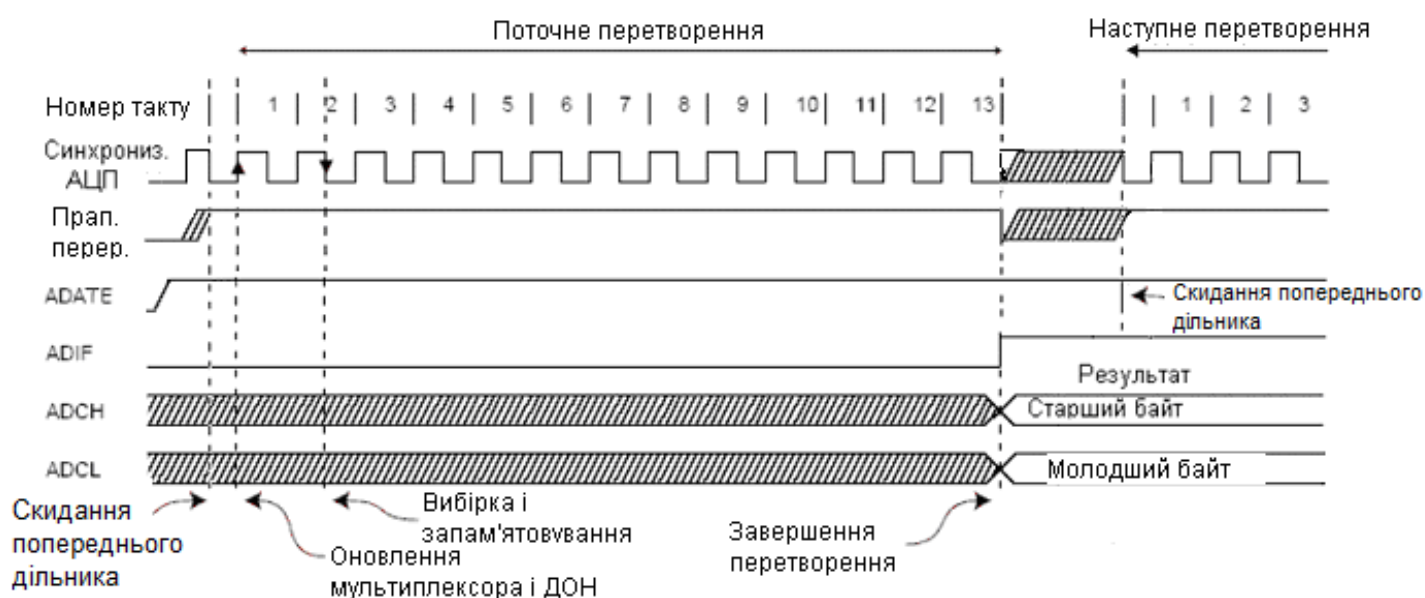


Рисунок 8.8 – Часові діаграми роботи АЦП у режимі запуску за перериванням

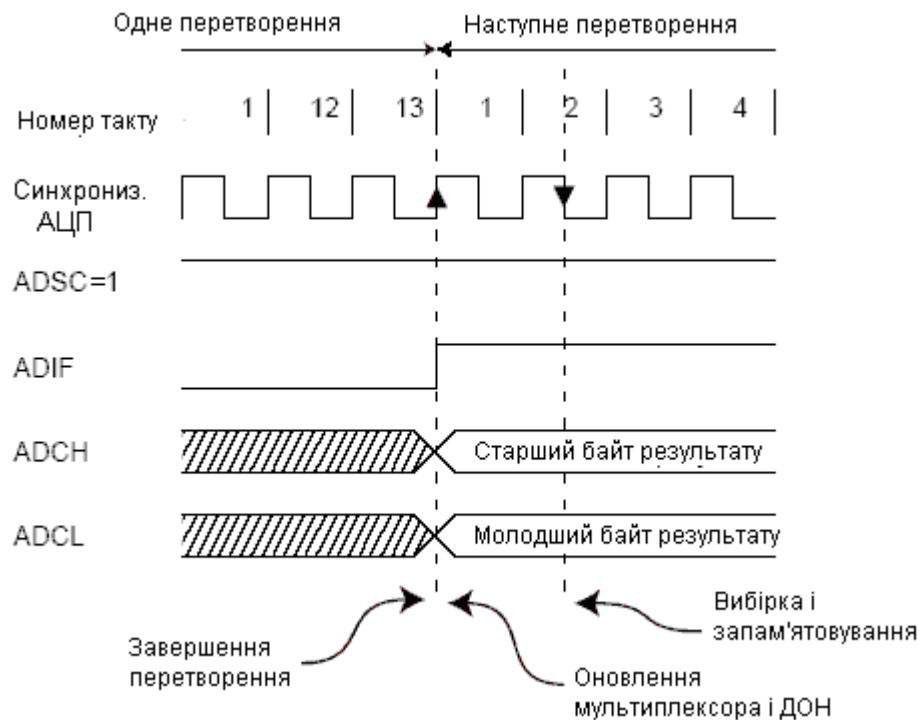


Рисунок 8.9 – Часові діаграми роботи АЦП у режимі автоматичного перезавпуску для однополярного перетворення

Якщо ініціюється одиночне однополярне перетворення встановленням розряду ADSC у регістрі ADCSRA, то перетворення починається з наступного наростаючого фронту тактового (синхро) сигналу АЦП.

У режимі одиночного перетворення нове перетворення може бути запущено відразу ж після скидання розряду ADSC (до збереження результату поточного перетворення). Однак реально цикл перетворення почнеться не раніше ніж через один такт після закінчення поточного перетворення.

Нормальне перетворення вимагає 13 тактів синхронізації АЦП. Перше перетворення після включення АЦП (встановлення ADEN в регістрі ADCSRA) вимагає 25 тактів синхронізації АЦП за рахунок необхідності ініціалізації модуля.

Після початку нормального перетворення на вибірку-зберігання затрачується 1.5 такти синхронізації АЦП, а після початку першого перетворення - 13,5 тактів. По завершенні перетворення результат зберігається в регістрах даних АЦП і встановлюється прапорець ADIF. У режимі одиночного перетворення одночасно скидається розряд ADSC. Програмно розряд ADSC може бути знову встановлено і нове перетворення буде ініційовано першим наростаючим фронтом тактового сигналу АЦП.

У режимі безперервного перетворення (автоматичного перезапуску) нове перетворення починається відразу по завершенні попереднього, при цьому ADSC залишається у високому стані. Час перетворення для різних режимів перетворення представлено в таблиці 8.7.

Таблиця 8.7 –Час перетворення АЦП

Тип перетворення	Тривалість вибірки-зберігання (у тактах з моменту початку перетворення)	Час перетворення (у тактах)
Перше перетворення	14.5	25
Нормальне однополярне перетворення	1.5	13
Нормальне диференціальне перетворення	1.5/2.5	13/14

Якщо використовується запуск за перериванням, то цикл перетворення починається за першим наростаючим фронтом тактового сигналу після встановлення прапорця обраного переривання. Причому при встановленні цього прапорця здійснюється скидання попереднього дільника модуля АЦП. Тим самим забезпечується фіксована затримка між генерацією запиту на переривання й початком циклу перетворення.

Перетворення запускається при встановленні відповідного прапорця, тобто навіть тоді, коли саме перетворення встановленням прапорця ADSCA (ADSC) не ініціалізувалося.

Як і прапорці інших переривань, прапорець ADIF скидається апаратно при запуску підпрограми обробки переривання від АЦП або програмно, записом у нього логічної 1. Дозвіл переривання здійснюється встановленням в "1" розряду ADIE регістра ADCSR при встановленому прапорці I регістра SREG.

Диференціальне перетворення синхронізується з внутрішнім генератором тактових імпульсів SKADC2, частота якого дорівнює половині частоти синхронізуючого пристрою АЦП. Ця синхронізація виконується автоматично інтерфейсом АЦП таким чином, щоб вибірка-зберігання відбувалась у певній фазі SKADC2.

Диференціальне перетворення, що ініційовано користувачем (тобто, всі одиночні перетворення та перше перетворення), коли SKADC2 має низький рівень, виконується за той же час, що й одиночне перетворення – 13 тактів. Диференціальне перетворення, що ініційовано користувачем в той час, коли SKADC2 мало високий рівень, через специфіку механізму синхронізації буде виконано за 14 тактів.

Ці процеси, в свою чергу, також впливатимуть на тривалість вибірки-зберігання – 1,5 або 2,5 такти відповідно.

Якщо використовуються диференціальні вхідні канали, то необхідно взяти до уваги деякі особливості.

У режимі автоматичного перезапуску нове перетворення ініціюється відразу по завершенні попереднього, і так як в цей момент SKADC2 дорівнює лог. 1, всі перетворення, що були автоматично перезапущені (тобто всі, крім першого), будуть тривати 14 тактів синхронізації АЦП.

Окремо варто сказати про використання режиму запуску заперериванням одночасно з диференціальними каналами. В такому

випадку АЦП необхідно вимикати між перетвореннями, щоб уникнути некоректних вимірювань, пов'язаних із невизначеністю моменту скидання попереднього дільника АЦП. У результаті вимикання і вмикання АЦП між перетвореннями виконуватимуться лише «довгі» перетворення, результати яких завжди будуть коректними.

Підсилювальний каскад оптимізований під частотний діапазон до 4 кГц для будь-яких коефіцієнтів підсилення. Підсилення сигналів більш високих частот буде нелінійним.

Тому, якщо вхідний сигнал містить частотні складові вище частотного діапазону підсилювального каскаду, необхідно встановити зовнішній фільтр низьких частот. Зверніть увагу, що частота синхронізації АЦП не пов'язана з обмеженням за частотним діапазоном підсилювального каскаду.

8.6 Керування вхідним мультиплексором

Виводи мікроконтролера, які підключені до входу АЦП, визначаються станом розрядів MUX3...MUX0; MUX4...MUX0; MUX5...MUX0, регістра ADMUX, та регістра ADCSRB відповідно до таблиць 8.8...8.10. Для каналів з диференціальним входом зазначені розряди визначають також коефіцієнт попереднього підсилення вхідного сигналу.

Слід зазначити, що попередній підсилювач, що використовується каналами з диференціальним входом, має вбудовану схему корекції зсуву. Величина зсуву, що залишилася після корекції, може бути врахована програмним способом.

Для цього треба обидва входи диференціального підсилювача підключити до того самого вивода мікроконтролера (таблиці 8.9, 8.10), а потім віднімати отримане значення з результату наступних перетворень.

Таким чином, помилка зсуву може бути знижена до величини, меншої 1МЗР.

Таблиця 8.8 – Керування вхідним мультиплексором у моделях
ATmega8x, ATmega48x/88x/168x

MUX3...MUX0	Однополярний (несиметричний) вхід
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6 ¹
0111	ADC7 ¹
1000...1101	Зарезервовано
1110	1.22 В (1,1 В ²)
1111	0В (GND)

¹ Є тільки в корпусах TQFP-32 і MLF-32.
² В моделях ATmega48x/88x/168x

Програмування розрядів MUXn (n=0,1,...,5) і REFS1:0 у регістрах ADMUX та ADCSRB підтримується буферизацією через тимчасовий регістр. Цим гарантується, що нові налаштування каналу перетворення й опорного джерела набудуть чинності в безпечний момент для перетворення.

До початку перетворення будь-які зміни каналу та опорного джерела набувають чинності відразу після їхньої модифікації. Як тільки починається процес перетворення доступ до зміни каналу та опорного джерела блокується, чим гарантується достатність часу на перетворення для АЦП. Безперервність модифікації повертається на останньому такті АЦП перед завершенням перетворення (перед встановленням прапорця ADIF у регістрі ADCSRA). Зверніть увагу, що перетворення починається наступним наростаючим фронтом тактового сигналу АЦП після встановлення ADSC. Таким чином, користувачеві не рекомендовано

записувати нове значення каналу або опорного джерела в ADMUX до 1-го такту синхронізації АЦП після встановлення ADSC.

Таблиця 8.9 – Керування вхідним мультиплексором у моделях ATmega16x/164x/32x/8535x/64x/128x/164x/165x/325x/3250x/645x/6450x/ 1281x/2561x

MUX4...MUX0	Однополярний вхід	Диференціальний вхід		Попереднє підсилення
		(додатний)	(від'ємний)	
00000	ADC0	Не застосовується		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000 ¹	Не застосовується	ADC0	ADC0	10x
01001 ¹		ADC1	ADC0	10x
01010 ¹		ADC0	ADC0	200x
01011 ¹		ADC1	ADC0	200x
01100 ¹		ADC2	ADC2	10x
01101 ¹		ADC3	ADC2	10x
01110 ¹		ADC2	ADC2	200x
01111 ¹		ADC3	ADC2	200x
10000 ¹		ADC0	ADC1	1x
10001 ¹		ADC1	ADC1	1x
10010 ¹		ADC2	ADC1	1x
10011 ¹		ADC3	ADC1	1x
10100 ¹		ADC4	ADC1	1x
10101 ¹		ADC5	ADC1	1x
10110 ¹		ADC6	ADC1	1x
10111 ¹		ADC7	ADC1	1x
11000 ¹	Не застосовується	ADC0	ADC2	1x
11001 ¹		ADC1	ADC2	1x
11010 ¹		ADC2	ADC2	1x
11011 ¹		ADC3	ADC2	1x
11100 ¹		ADC4	ADC2	1x
11101 ¹		ADC5	ADC2	1x
11110	1.22 В (1,1 В ¹)	Не застосовується		
11111	0В (GND)			

¹ У моделях ATmega165x/325x/3250x/645x/6450x/1251x/2561x

¹ У моделях ATmega165x/325x/3250x/645x/6450x/1251x/2561x

Таблиця 8.10 – Керування вхідним мультиплексором в моделях
 ATmega640x/1280x/2560x

MUX5...MUX0	Несиметричний Вхід	Диференціальний вхід		Попереднє підсилення
		додатний	від’ємний	
000000	ADC0	—		
000001	ADC1			
000010	ADC2			
000011	ADC3			
000100	ADC4			
000101	ADC5			
000110	ADC6			
000111	ADC7			
001000	—	ADC0	ADC0	10x
001001		ADC1	ADC0	10x
001010		ADC0	ADC0	200x
001011		ADC1	ADC0	200x
001100		ADC2	ADC2	10x
001101		ADC3	ADC2	10x
001110		ADC2	ADC2	200x
001111		ADC3	ADC2	200x
010000		ADC0	ADC1	1x
010001		ADC1	ADC1	1x
010010		ADC2	ADC1	1x
010011		ADC3	ADC1	1x
010100		ADC4	ADC1	1x

Продовження таблиці 8.10

MUX5...MUX0	Несиметричний Вхід	Диференціальний вхід		Попереднє підсилення
		додатний	від'ємний	
010101		ADC5	ADC1	1x
010110		ADC6	ADC1	1x
010111		ADC7	ADC1	1x
011000		ADC0	ADC2	1x
011001		ADC1	ADC2	1x
011010		ADC2	ADC2	1x
011011		ADC3	ADC2	1x
011100		ADC4	ADC2	1x
011101		ADC5	ADC2	1x
011110	1.1 В	—		
011111	0 В (GND)			
100000	ADC8			
100001	ADC9			
100010	ADC10			
100011	ADC11			
100100	ADC12			
100101	ADC13			
100110	ADC14			
100111	ADC15			
101000	—	ADC8	ADC8	10x
101001		ADC9	ADC8	10x
101010		ADC8	ADC8	200x
101011		ADC9	ADC8	200x
101100		ADC10	ADC10	10x
101101		ADC11	ADC10	10x
101110		ADC10	ADC10	200x
101111		ADC11	ADC10	200x
110000		ADC8	ADC9	1x
110001		ADC9	ADC9	1x
110010		ADC10	ADC9	1x
110011		ADC11	ADC9	1x
110100		ADC12	ADC9	1x
110101		ADC13	ADC9	1x
110110		ADC14	ADC9	1x

Продовження таблиці 8.10

MUX5...MUX0	Несиметричний Вхід	Диференціальний вхід		Попереднє підсилення
		додатний	від'ємний	
110111	-	ADC15	ADC9	1x
111000		ADC8	ADC10	1x
111001		ADC9	ADC10	1x
111010		ADC10	ADC10	1x
111011		ADC11	ADC10	1x
111100		ADC12	ADC10	1x
111101		ADC13	ADC10	1x
111110	Зарезервовано	—		
111111	Зарезервовано			

Особливі заходи необхідно вжити при зміні диференціального каналу. Як тільки здійснено вибір диференціального каналу підсилювальному каскаду потрібно 125 мкс для стабілізації нового значення. Отже, протягом перших 125 мкс після перемикання диференціального каналу перетворення не повинно стартувати. Якщо ж у цей період перетворення все-таки виконувалися, то їхній результат необхідно ігнорувати.

Таку ж затримку на встановлення необхідно ввести при першому диференціальному перетворенні після зміни опорного джерела АЦП (за рахунок зміни розрядів REFS1:0 в ADMUX).

Якщо дозволено роботу інтерфейсу JTAG, то функції каналів АЦП на виводах портів, що використовуються для модуля JTAG, відміняються [2].

У нових мікроконтролерах, а саме в ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega165x, ATmega325x/3250x/645x/6450x і ATmega640x/1280x/1281x/2560x/2561x, є можливість відключення вхідних цифрових буферів на виводах ADC0...ADC15 у випадку, якщо відповідні виводи використовуються тільки для зчитування аналогових сигналів. При відключених цифрових буферах зменшується загальний струм споживання

мікроконтролера, а відповідні розряди регістрів PINX завжди читаються як 0.

Відключення цифрових буферів на входах ADC0...ADC7 здійснюється записом лог. 1 відповідно в розряди ADC0D...ADC7D регістра DIDR0, розташованого за адресою (\$7E). А відключення буферів на входах ADC8...ADC15 (моделі ATmega640x/1280x/2560x) здійснюється записом лог. 1 у розряди ADC8D...ADC15D регістра DIDR2, розташованого за адресою (\$7D). Формат цих регістрів наведено на рисунку 8.10.

	7	6	5	4	3	2	1	0	ATmega48x/88x/168x ATmega165x ATmega325x/3250x ATmega645x/6450x ATmega640x/1280x/1281x ATmega2560x/2561x
	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	
Зчитування(R)/Запис(W)	R/W	R/W	R/W		R/W	R/W	R/W	R/W	
Початкове значення	0	0	0		0	0	0	0	

	7	6	5	4	3	2	1	0	ATmega640x/1280x/2560x
	ADC15D	ADC14D	ADC13D	ADC12D	ADC11D	ADC10D	ADC09D	ADC08D	
Зчитування(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 8.10 – Формат регістрів DIR0 (а) и DIR2 (б)

8.7 Збереження результату перетворення

Після завершення перетворення (при встановленні в "1" прапорця ADIF регістра ADCSR) його результат зберігається в регістрі даних АЦП. Оскільки АЦП має 10 розрядів, цей регістр фізично розміщено у двох регістрах введення/виведення ADCH:ADCL, доступних тільки для читання. Ці регістри розташовані за адресами \$05:\$04 і при включенні мікроконтролера містять значення "\$0000".

У початковому стані результат перетворення вирівнюється вправо (старші 6 розрядів регістра ADCH – не є значущими).

Однак він може вирівнюватися також вліво (молодші 6 розрядів регістра ADCL – не є значущими). Для керування вирівнюванням

результату перетворення призначено розряд ADLAR регістра ADMUX. Якщо цей розряд встановлено в "1", результат перетворення вирівнюється за лівою границею 16-розрядного слова, якщо скинутий в "0" - за правою границею.

При використанні диференціального режиму перетворення результат представляється в коді двійкового доповнення до двох (в додатковому коді).

У таблиці 8.11 наведено приклади вирівнювання результату вліво та вправо.

Таблиця 8.11 – Вирівнювання результату АЦП

ADLAR	Розряд	15	14	13	12	11	10	9	8	
0		-	-	-	-	-	-	ADC9	ADC8	ADCH
		ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	Розряд	7	6	5	4	3	2	1	0	
	R/ $\overline{W}$	R	R	R	R	R	R	R	R	ADCH
		R	R	R	R	R	R	R	R	ADCL
	Поч. зн.	0	0	0	0	0	0	0	0	ADCH
1		0	0	0	0	0	0	0	0	ADCL
	Розряд	15	14	13	12	11	10	9	8	
		ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
		ADC1	ADC0	-	-	-	-	-	-	ADCL
	Розряд	7	6	5	4	3	2	1	0	
	R/ $\overline{W}$	R	R	R	R	R	R	R	R	ADCH
		R	R	R	R	R	R	R	R	ADCL
	Поч. зн.	0	0	0	0	0	0	0	0	ADCH
		0	0	0	0	0	0	0	0	ADCL

Звернення до регістрів ADCH і ADCL для отримання результату перетворення повинно виконуватися в певній послідовності: спочатку необхідно прочитати регістр ADCL, а потім ADCH. Ця вимога пов'язана з тим, що після звернення до регістра ADCL процесор блокує доступ до регістрів даних з боку АЦП доти, поки не буде прочитано регістр ADCH. Завдяки цьому можна бути впевненим, що при читанні регістрів ADCH, ADCL у них будуть перебувати складові того самого результату.

Відповідно, якщо чергове перетворення завершиться до звернення до регістра ADCH, результат перетворення буде загублено. З іншого боку, якщо результат перетворення вирівнюється вліво й досить точності 8-розрядного значення, для отримання результату можна прочитати тільки вміст регістра ADCH.

8.8 Рекомендації з перемикання вхідних каналів

При перемиканні вхідних каналів необхідно врахувати деякі рекомендації, які виключають некоректність перемикання.

У режимі одиночного (несиметричного) перетворення програмування перемикання каналів необхідно виконувати перед початком перетворення. Перемикання каналів відбувається протягом одного такту синхронізації АЦП після запису лог. 1 в ADSC. Тому найпростішим способом є очікування завершення перетворення перед вибором нового каналу.

У режимі автоматичного перезапуску режим необхідно вибирати перед початком першого перетворення. Перемикання каналів відбувається аналогічно - протягом одного такту синхронізації АЦП після запису лог. 1 в ADSC. Найпростішим способом є очікування завершення першого перетворення, а потім перемикання каналів.

Оскільки наступне перетворення вже запущено автоматично, то новий результат буде відповідати попередньому каналу. Наступні перетворення відображають результат для нового каналу.

При перемиканні на диференціальний канал перше перетворення буде характеризуватися поганою точністю через перехідний процес у схемі автоматичного регулювання зсуву. Отже, перший результат такого перетворення рекомендується ігнорувати.

8.9 Особливості підключення джерела опорної напруги

Джерело опорної напруги (ДОН) для АЦП ($U_{\text{ДОН}}$) визначає діапазон перетворення АЦП. Якщо рівень однополярного сигналу понад $U_{\text{ДОН}}$, то результатом перетворення буде 0x3FF. У якості $U_{\text{ДОН}}$ можуть виступати AVCC, внутрішнє ДОН 2,56 В (1,1 В) або зовнішнє ДОН, що підключено до виводу AREF. AVCC підключається до АЦП через пасивний ключ. Внутрішня опорна напруга 2,56 В (1,1 В) генерується внутрішнім еталонним джерелом VBG, що буферизовано внутрішнім підсилювачем. У кожному разі зовнішній вивід AREF зв'язаний безпосередньо з АЦП і, тому, можна знизити вплив шумів на опорне джерело за рахунок підключення конденсатора між виводом AREF і спільним виводом. Напруга $U_{\text{ДОН}}$ також може бути виміряна на виводі AREF вольтметром з високим входним опором. Зверніть увагу, що $U_{\text{ДОН}}$ є високоомним джерелом і, тому, зовні до нього може бути підключене тільки ємнісне навантаження.

Якщо користувач використовує зовнішнє опорне джерело, що підключено до виводу AREF, то не допускається використання іншої опції опорного джерела, тому що це приведе до шунтування зовнішньої опорної напруги. Якщо до виводу AREF не прикладена напруга, то користувач може обрати AVCC і 2,56 В (1,1 В) як опорне джерело. Результат першого перетворення після перемикання опорного джерела може характеризуватися низькою точністю, тому користувачеві рекомендується його ігнорувати.

8.10 Особливості схеми аналогового входу

Схема аналогового входу для однополярних каналів представлено на рисунку 8.11.

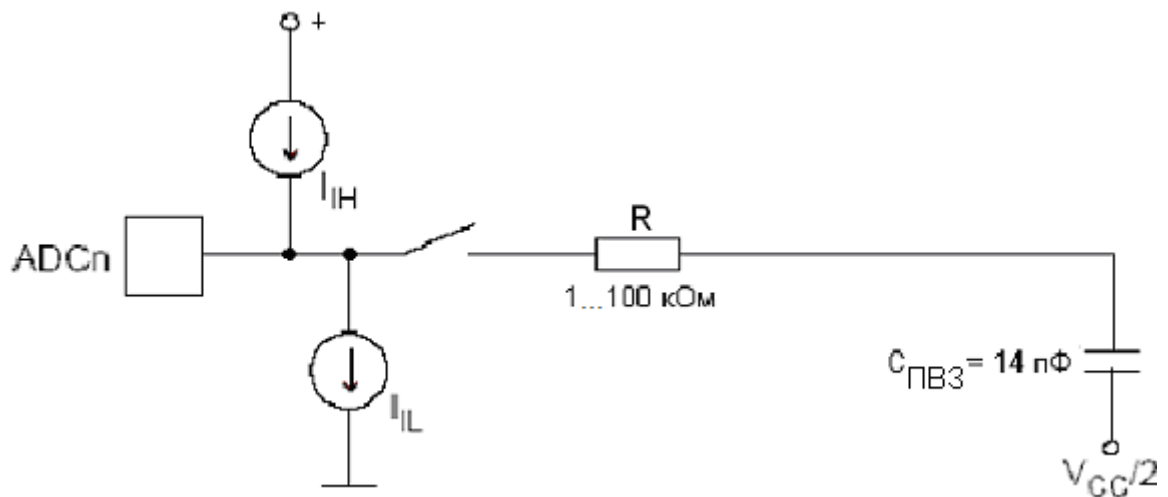


Рисунок 8.11 – Схема аналогового входу

Незалежно від того, який канал підключено до АЦП, аналоговий сигнал, підключено до виводу ADCn, навантажується ємністю виводу й вхідним опором витoku. Після підключення каналу до АЦП аналоговий сигнал буде пов'язано з конденсатором вибірки-зберігання через послідовний резистор, опір якого еквівалентний всьому вхідному ланцюгу.

АЦП оптимізований під аналогові сигнали з вихідним опором не більше 10 кОм. Якщо використовується таке джерело сигналу, то час вибірки незначний. Якщо ж використовується джерело з більш високим вихідним опором, то час вибірки буде визначатися часом, що потрібний для зарядки конденсатора вибірки-зберігання джерелом аналогового сигналу. Рекомендовано використовувати джерела тільки з малим вихідним опором і сигналами, що повільно змінюються, тому що в цьому випадку буде досить швидким зарядження конденсатора вибірки-зберігання.

Стосовно каналів з диференціальним підсиленням рекомендовано використовувати сигнали із внутрішнім опором до декількох сотень кОм. Варто передбачити, щоб у попередніх каскадах формування аналогового сигналу до входу АЦП не вносилися частоти вище $f_{\text{ацп}}/2$, в іншому випадку

результат перетворення може бути некоректним. Якщо ймовірність проникнення високих частот існує, то рекомендується перед АЦП встановити фільтр низьких частот.

8.11 Результат перетворення АЦП

Для каналів з однополярним (несиметричним) входом результат перетворення визначається виразом: $ADC = \frac{1024U_{IN}}{U_{REF}}$, де U_{IN} – значення вхідної напруги в мілівольтах, а U_{REF} - величина опорної напруги. На рисунку 8.12 представлено функцію перетворення АЦП в однополярному режимі. Код 0x000 відповідає рівню аналогової землі, а 0x3FF – рівню напруги ДОН мінус 1 крок квантування за напругою.

Таблиця 8.12 відображає зв'язок між вхідним сигналом й вихідними кодами для однополярного режиму.

Приклад: Нехай $ADMUX = 0x00...0x07$ (будь-який однополярний вхід), напруга на одному з входів 1000 мВ, напруга ДОН рівна 2,56 В, тоді:

$$\text{КодАЦП} = 1024 * 1000 / 2560 = 400 = 0x190.$$

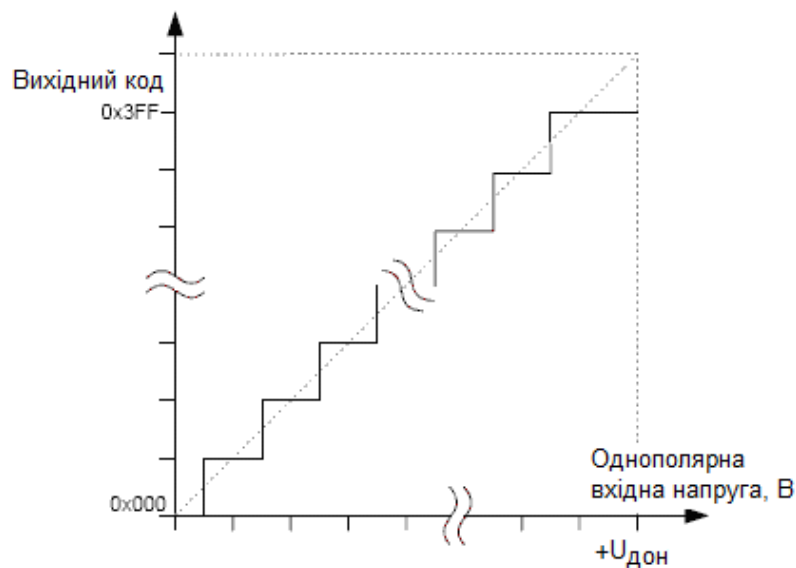


Рисунок 8.12 – Функція перетворення АЦП при зміні однополярного сигналу

Таблиця 8.12 – Зв'язок між вхідним и вихідним кодами

$U_{\text{АЦПн}}^*$	Зчитаний код	Відповідне десяткове значення
$U_{\text{АЦПн}} + U_{\text{ДОН}}$	0x3FF	1023
$U_{\text{АЦПн}} + 0.999 U_{\text{ДОН}}$	0x3FF	1023
$U_{\text{АЦПн}} + 0.998 U_{\text{ДОН}}$	0x3FE	1022
...	...	...
$U_{\text{АЦПн}} + 0.001 U_{\text{ДОН}}$	0x001	1
$U_{\text{АЦПн}}$	0x000	0

$U_{\text{АЦПн}}$ – вхідна напруга, яка дорівнює нулю, $U_{\text{АЦПн}}$ – поточне значення вхідної напруги.

Для каналів з диференціальним входом результат перетворення визначається виразом: $ADC = 512 \cdot K_{\text{п}} \cdot \frac{U_{\text{POS}} - U_{\text{NEG}}}{U_{\text{REF}}}$, де U_{POS} – величина напруги на додатному (неінвертуючому) вході, U_{NEG} – величина напруги на від'ємному (інвертую чому) вході, а $K_{\text{п}}$ – коефіцієнт підсилення. Результат перетворення представляється в цьому випадку в додатковому коді, а його значення лежить у діапазоні: \$200 (-512)...\$1FF (+511).

По завершенні перетворення ($\text{ADIF} = 1$) результат може бути зчитаний з пари регістрів результату перетворення АЦП (ADCL , ADCH).

Для каналів з диференціальним входом результат представляється в коді двійкового доповнення, починаючи з 0x200 (-512d) до 0x1FF (+511d). Зверніть увагу, що при необхідності швидко визначити полярність результату досить опитати старший розряд результату перетворення (ADC9 в ADCH). Якщо даний розряд дорівнює лог. 1, то результат від'ємний, якщо ж лог. 0, то додатний. На рисунку 8.13 представлено функцію перетворення АЦП у диференціальному режимі.

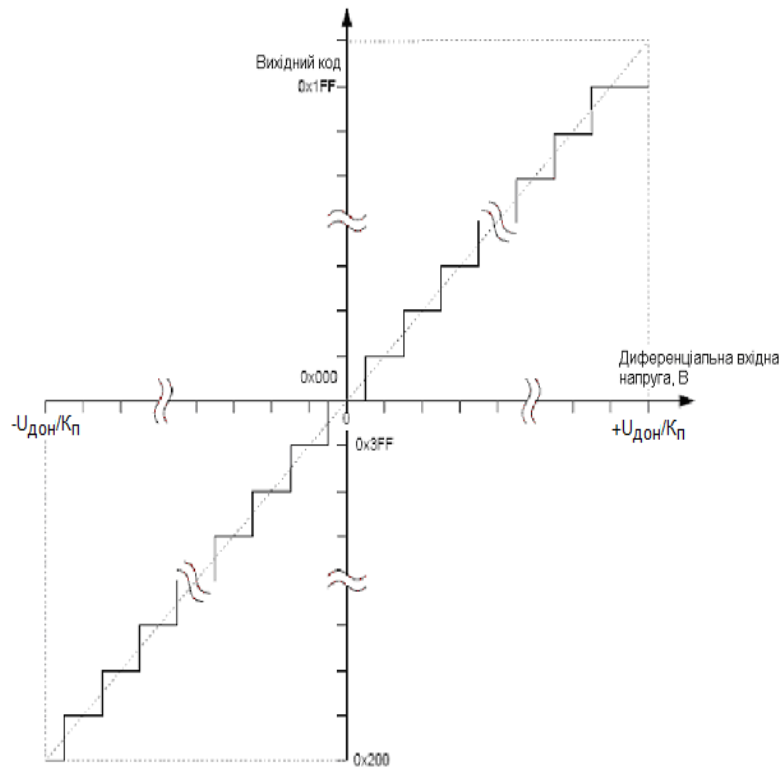


Рисунок 8.13 – Функція перетворення АЦП при зміні диференціального сигналу

У таблиці 8.13 представлено результуючі вихідні коди для диференціальної пари каналів (ADC_n - ADC_m) з коефіцієнтом підсилення K_p й опорною напругою $U_{дон}$.

Приклад: Нехай ADMUX = 0xED (пари входів ADC3 – ADC2, $K_p=10$, $U_{дон}=2.56V$, результат з лівостороннім вирівнюванням), напруга на вході ADC3 = 300 мВ, а на вході ADC2 = 500 мВ, тоді:

$$КОД_{АЦП} = 512 \cdot 10 \cdot \frac{300 - 500}{2560} = -400 = 0x270.$$

З урахуванням обраного формату розміщення результату (лівосторонній) ADCL = 0x00, а ADCH = 0x9C. Якщо ж обраний правосторонній формат (ADLAR=0), то ADCL = 0x70, ADCH = 0x02.

Таблиця 8.13 – Зв’язок між вхідними сигналами і вихідними кодами для каналів з диференціальним входом

$U_{\text{АЦПm}} *$	Зчитаний код	Відповідне десяткове значення
$U_{\text{АЦПm}} + U_{\text{ДОН}} / K_{\text{п}}$	0x1FF	511
$U_{\text{АЦПm}} + 0.999 U_{\text{ДОН}} / K_{\text{п}}$	0x1FF	511
$U_{\text{АЦПm}} + 0.998 U_{\text{ДОН}} / K_{\text{п}}$	0x1FE	510
...	...	...
$U_{\text{АЦПm}} + 0.001 U_{\text{ДОН}} / K_{\text{п}}$	0x001	1
$U_{\text{АЦПm}}$	0x000	0
$U_{\text{АЦПm}} - 0.001 U_{\text{ДОН}} / K_{\text{п}}$	0x3FF	-1
...	...	...
$U_{\text{АЦПm}} - 0.999 U_{\text{ДОН}} / K_{\text{п}}$	0x201	-511
$U_{\text{АЦПm}} - U_{\text{ДОН}} / K_{\text{п}}$	0x200	-512

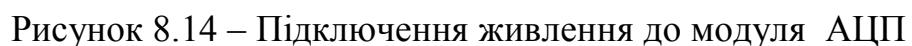
$U_{\text{АЦПm}}$ – вхідна напруга, яка дорівнює нулю, $U_{\text{АЦПn}}$ – поточне значення вхідної напруги.

8.12 Підвищення точності перетворення

8.12.1 Загальні рекомендації

У цьому розділі наведені деякі рекомендації, що дозволяють найбільшою мірою використати можливості АЦП. Насамперед, для мінімізації похибки самого АЦП необхідно правильно обрати тактову частоту перетворення. Із цією метою на вході АЦП рекомендується встановлювати фільтр низьких частот. Крім того, при розробці конструкції та топології друкованої плати варто дотримуватися загальних правил проектування цифро-аналогових пристроїв.

4. Якщо який-небудь із виводів АЦП використовується як цифровий вихід, то надзвичайно важливо не допустити перемикання стану цього виходу в процесі перетворення.



8.12.2 Придушувач шумів

Модуль АЦП має можливість придушення шумів, які викликані роботою ядра ЦПП та периферійних пристроїв введення/виведення (один з режимів зменшеного енергоспоживання МК-ра) [2]. Придушувач шумів може бути використано у режимі зниження шумів АЦП (ADC Noise Reduction) і в режимі холостого ходу. При використанні даної функції необхідно дотримуватися наступної процедури:

1. Переконайтесь, що роботу АЦП дозволено та він не виконує перетворення. Виберіть режим одиночного перетворення й дозвольте переривання за завершенням перетворення.

2. Запрограмуйте режим зменшення шумів АЦП або режим холостого ходу [2]. АЦП запустить перетворення як тільки зупиниться ЦПП.

3. Якщо до завершення перетворення не виникає інших переривань, то АЦП викличе переривання ЦПП і програма перейде на вектор обробки переривання за завершенням перетворення АЦП. Якщо до завершення перетворення інше переривання будить мікроконтролер, то це переривання обробляється, а по завершенні перетворення генерується відповідний запит на переривання. АЦП залишається в активному режимі доки не буде виконано чергову команда Sleep.

Зверніть увагу, що АЦП не відключається автоматично при переведенні в усі режими сну, крім режиму холостого ходу й зниження шумів АЦП. Тому, користувач повинен передбачити запис лог. 0 у розряд ADEN перед переведенням у такі режими сну щоб уникнути надмірного енергоспоживання.

Якщо роботу АЦП було дозволено в режимах сну та користувач бажає виконати диференціальне перетворення, то після пробудження необхідно включити, а потім виключити АЦП для ініціалізації

розширеного перетворення, чим буде гарантовано отримання дійсного результату.

8.12.3 КОМПЕНСАЦІЯ ЗСУВУ

Підсилювальний каскад має вбудовану схему компенсації зсуву, яка прагне максимально наблизити до нуля зсув диференціального виміру. Зсув, що залишився, можна виміряти, якщо як диференціальні входи АЦП обрати той самий вивід мікроконтролера. Виміряний в такий спосіб залишковий зсув можна програмно відняти з результату перетворення. Використання програмного алгоритму корекції зсуву дозволяє зменшити зсув нижче одного молодшого розряду.

8.12.4 ВИЗНАЧЕННЯ ПОХИБОК АНАЛОГО-ЦИФРОВОГО ПЕРЕТВОРЕННЯ

N-розрядний однополярний АЦП перетворює напругу лінійно між GND і $U_{\text{дон}}$ з кількістю кроків 2^N мол. розрядів. Мінімальний код = 0, максимальний = $2^N - 1$. Основні похибки перетворення є відхиленням реальної функції перетворення від ідеальної. До них відносяться:

1. Похибка зсуву - відхилення першого переходу (з 0x000 на 0x001) у порівнянні з ідеальним переходом (тобто при 0.5 мол. розр.). Ідеальне значення : 0 мол. розр. (рисунок 8.15).

2. Похибка підсилення (рисунок 8.16). Після корегування зсуву похибка підсилення являє собою відхилення останнього переходу (з 0x3FE на 0x3FF) від ідеального переходу (тобто відхилення при максимальному значенні мінус 1,5 мол. розр.). Ідеальне значення: 0 мол. розр.

3. Інтегральна нелінійність (ІНЛ) (рисунок 8.17). Після корегування зсуву й похибки підсилення ІНЛ являє собою максимальне відхилення реальної функції перетворення від ідеальної для будь-якого коду. Ідеальне значення ІНЛ = 0 мол. розр.

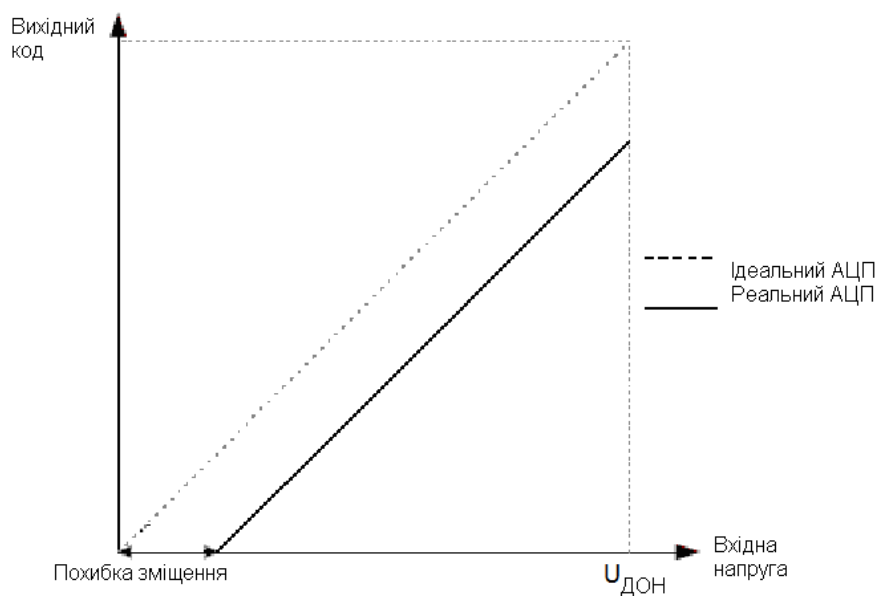


Рисунок 8.15 – Похибка зсуву

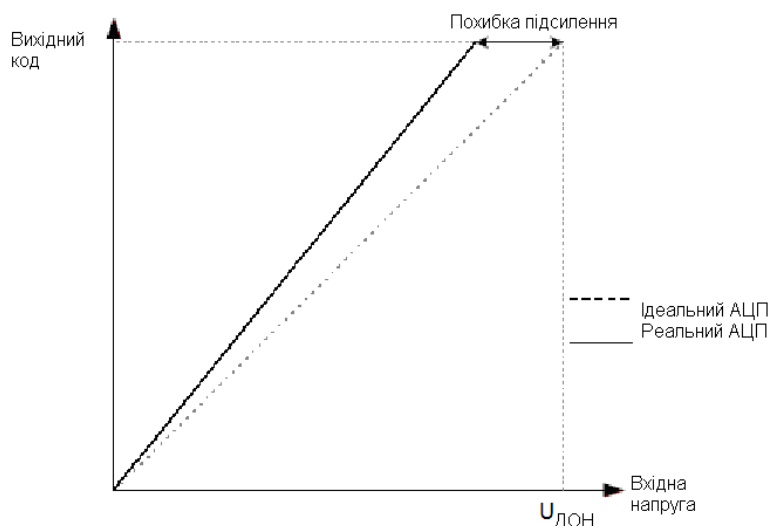


Рисунок 8.16 – Похибка підсилення

4. Диференціальна нелінійність (ДНЛ) (рисунок 8.18). Максимальне відхилення ширини фактичного коду (інтервал між двома суміжними переходами) від ширини ідеального коду (1 мол. розр.). Ідеальне значення: 0 мол. розр.

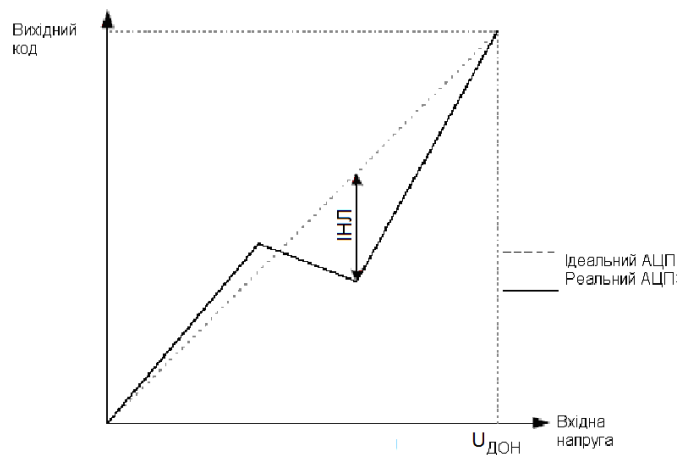


Рисунок 8.17 – Інтегральна нелінійність

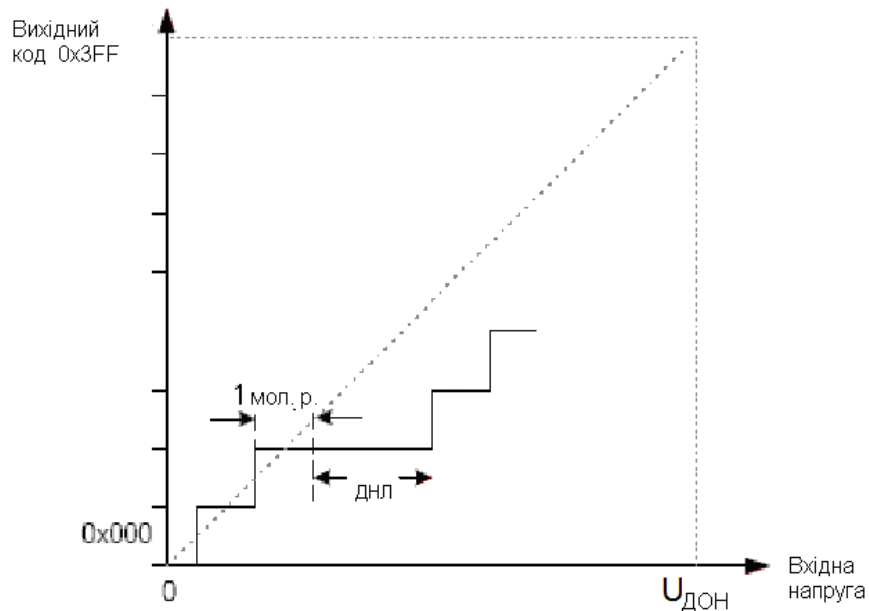


Рисунок 8.18 – Диференціальна нелінійність (ДНЛ)

5. Похибка квантування за рівнем. Виникає через ділення (квантування) вхідної напруги на постійні інтервали ΔU , границям яких відповідає постійне значення двійкового коду.

Зміні вхідної напруги на $\pm \Delta U$ відповідає зміна вихідного двійкового коду на ± 1 молодшого значущого розряду (МЗР). Абсолютна похибка квантування за рівнем дорівнює $\pm \Delta U/2$ або $\pm 0,5$ МЗР.

Відносна похибка квантування за рівнем

$$\delta_{\text{вдн}} = \frac{50}{N_{\text{д}} - 1} \%,$$

де $N_{\text{д}}$ - число дискретних значень вихідного коду, включаючи нульове.

У свою чергу

$$N_{\text{д}} = 2^{N_{\text{РДК}}},$$

де $N_{\text{РДК}}$ - число розрядів вихідного двійкового коду.

Для нашого АЦП $N_{\text{РДК}} = 10$, $N_{\text{д}} = 2^{10} = 1024$. Тоді,

$$\delta_{\text{вдн}} = \frac{50}{1024 - 1} \% = \frac{50}{1023} \% = 0,05 \, \%.$$

6. Абсолютна похибка. Максимальне відхилення реальної (без підстроювання) функції перетворення від ідеальної при будь-якому коді. Є результатом дії декількох ефектів: зсув, похибка підсилення, диференціальна похибка, нелінійність і похибка квантування. Ідеальне значення: ± 0.5 мол. розр.

8.13 ОСНОВНІ ПАРАМЕТРИ РОБОТИ АЦП

Основні параметри АЦП наведено в таблиці 8.13. Всі значення зазначені для діапазону температур навколишнього середовища $-40..+80^{\circ}\text{C}$ [2].

Таблиця 8.13 – Основні параметри АЦП

Позначення	Параметр	Умова	Min	typ	max
	Роздільна здатність [біт]	Однополярний вхід	-	10	-
		Диференціальний вхід, $K_U=1 \times 120 \times$	-	8	-
		Диференціальний вхід, $K_U = 200 \times$	-	7	-
	Абсолютна похибка [МЗР]	Несиметричний вхід, $U_{REF} = 4V$ $f_{ADC} = 200 \text{кГц}$	-	2	2.5
		Однополярний вхід, $U_{REF} = 4V$ $f_{ADC} = 1 \text{МГц}$	-	4	-
ІНЛ (INL)	Інтегральна нелінійність [МЗР]	$U_{REF} = 4V$	-	0.5	-
ДНЛ (DNL)	Диференціальна нелінійність [МЗР]	$U_{REF} = 4V$	-	0.25	-
-	Помилка зсуву [МЗР]	$V_{REF} = 4V$	-	0.75	-
-	Час перетворення [мкс]	Режим безперервного перетворення	13	-	260
f_{ADC}	Тактова частота [кГц]	-	50	-	1000
AV_{CC}	Напруга живлення [В]	-	$U_{CC}-0.3$	-	$U_{CC}+0.3$
U_{REF}	Опорна напруга [В]	Однополярний вхід	2.0	-	U_{CC}
		Диференціальний вхід	2.0	-	$U_{CC}-0.2$
U_{INT}	Напруга внутрішнього ДОН [В]	-	2.4	2.56	2.7
		-	1.0	1.1	1.2
R_{REF}	Вхідний опір каналу опорної напруги [кОм]	-	-	32	-
R_{AIN}	Вхідний опір аналогового входу [МОм]	-	-	100	-

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) До складу яких моделей AVR-мікроконтролерів входить модуль АЦП?
- 2) Назвіть основні параметри АЦП.
- 3) Назвіть основні джерела опорної напруги для АЦП.
- 4) Назвіть режими роботи АЦП.
- 5) Яку роль виконує дешифратор?
- 6) У якому регістрі зберігається результат перетворення?
- 7) Поясніть як саме відбувається перетворення вхідного сигналу у двійковий код?
- 8) Яку роль виконує компаратор?
- 9) Який принцип перетворення використовується в АЦП?
- 10) Який регістр відповідає за налаштування мультиплексора АЦП?
- 11) Наведіть та поясніть формати регістрів стану і керування АЦП.
- 12) Який розряд регістра ADCSRA відповідає за дозвіл роботи АЦП?
- 13) Як обрати джерело опорної напруги?
- 14) Як вибрати режим роботи АЦП?
- 15) За якими перериваннями може відбуватися запуск АЦП?
- 16) Яку функцію виконує попередній дільник?
- 17) Як формується тактовий сигнал АЦП?
- 18) Який регістр використовується для налаштування вхідного мультиплексора?
- 19) Як можна програмно врахувати величину зсуву?
- 20) Чому після перемикання диференціального каналу підсилювача перетворення не повинно стартувати не раніше, ніж через 125 мкс?
- 21) Назвіть способи вирівнювання результату АЦП.
- 22) Яке вирівнювання слід використовувати якщо досить точності 8-розрядного значення?

- 23) Які особливості перемикання каналів при одиночному перетворенні?
- 24) Які особливості перемикання каналів при перетворенні у режимі автоматичного перезапуску?
- 25) Що може виступати у ролі джерела опорної напруги?
- 26) Який вихідний опір повинно мати джерело вхідного сигналу?
- 27) Яким буде результат перетворення для каналів з однополярним входом?
- 28) Яким буде результат перетворення для каналів з диференціальним входом?
- 29) Назвіть методи підвищення точності перетворення АЦП.
- 30) Чому дорівнює похибка зсуву та як її можна програмно врахувати?
- 31) Що таке інтегральна нелінійність?
- 32) Що таке диференціальна нелінійність?

ЛІТЕРАТУРА [1...10]

9 СПЕЦІАЛЬНІ РЕЖИМИ РОБОТИ МІКРОКОНТРОЛЕРА

9.1 ТАКТУВАННЯ

9.1.1 Загальні відомості

Розглянемо питання тактування AVR-мікроконтролерів на прикладі сімейства Mega [1,2]. Для тактування можуть використовуватися самі різні джерела тактового сигналу. Перш за все, це вбудований кварцовий генератор з під'єднуваним зовнішнім резонатором. Також у якості тактового може використовуватися найпростіший RC-генератор – як з внутрішнім (каліброваним), так і з зовнішнім RC-ланцюгом. Крім того, в якості тактового може використовуватися сигнал від зовнішнього джерела.

9.1.2 Тактовий генератор

Спрощену структуру пристрою синхронізації мікроконтролерів сімейства Mega представлено на рисунку 9.1 (блок попереднього дільника і можливість підключення зовнішнього RC-ланцюжка є не у всіх моделях).

Як видно з рисунка, на базі системного (задаючого) тактового сигналу формуються додаткові сигнали, що використовуються для тактування різних модулів і блоків мікроконтролера:

– f_{clkCPU} – тактовий сигнал центрального процесора. Використовується для тактування блоків мікроконтролера, що відповідають за роботу з ядром мікроконтролера (регістровий файл, пам'ять даних і т. ін.). При відключенні цього сигналу центральний

процесорний пристрій (ЦПП) зупиняється і відповідно припиняються всі обчислення;

– $f_{clkI/O}$ – тактовий сигнал підсистеми введення/виведення. Використовується більшістю периферійних пристроїв, таких як таймери/лічильники та інтерфейсні модулі. Цей сигнал використовується також підсистемою зовнішніх переривань, проте ряд зовнішніх переривань може генеруватися і за його відсутності;

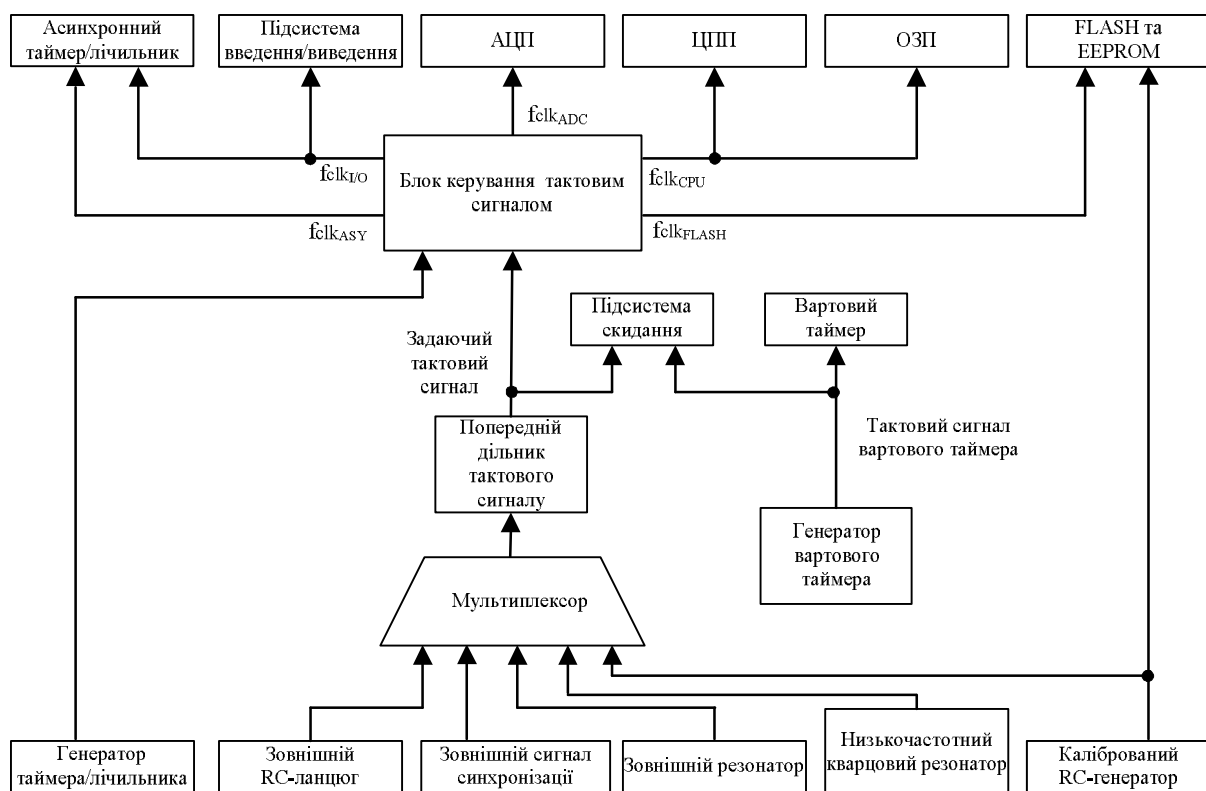


Рисунок 9.1 – Спрощена структура пристрою синхронізації

– $f_{clkFLASH}$ – тактовий сигнал для FLASH-пам'яті програм. Як правило, цей сигнал активується й деактивується одночасно з тактовим сигналом центрального процесора f_{clkCPU} ;

– f_{clkASY} – тактовий сигнал асинхронного таймера/лічильника. Тактування здійснюється безпосередньо від зовнішнього кварцового

резонатора (32768 Гц). Наявність цього сигналу дозволяє використовувати відповідний таймер/лічильник в якості годинника реального часу, навіть при знаходженні мікроконтролера у «Сплячому» режимі;

– f_{clkADC} – тактовий сигнал для модуля АЦП. Наявність цього тактового сигналу дозволяє здійснювати перетворення при зупинених ЦПП і підсистемі введення/виведення. При цьому значно зменшується рівень завад, що генеруються мікроконтролером, і відповідно збільшується точність перетворення.

Тактовий генератор мікроконтролерів сімейства Mega може працювати із зовнішнім кварцовим/керамічним резонатором, зовнішнім або внутрішнім RC-ланцюжком, а також із зовнішнім сигналом синхронізації. Можливість використання того чи іншого джерела тактового сигналу залежить від моделі мікроконтролера і наведено в таблиці 9.1.

Таблиця 9.1 – Джерела тактового сигналу

Джерело тактового сигналу	ATmega8x/16x/32x/ 64x/128x	ATmega8515x	ATmega8535x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/ 644x	ATmega165x	ATmega325x/3250x, ATmega645x/6450x	ATmega640x, ATmega1280x/1281x
Кварцовий генератор	•	•	•	•	•	•	•	•	•
Генератор з зовнішнім RC-ланцюгом	•	•	•						
Внутрішній RC-генератор	•	•	•	•	•	•	•	•	•
Зовнішній сигнал синхронізації	•	•	•	•	•	•	•	•	•

Оскільки архітектура мікроконтролерів повністю статична, мінімально допустима частота нічим не обмежена (аж до покрокового режиму роботи), а максимальна робоча частота визначається конкретною моделлю мікроконтролера.

У всіх моделях вибір режиму роботи тактового генератора здійснюється програмуванням конфігураційних комірок (FUSE Bits) CKSEL3...0. Необхідні значення для кожного режиму роботи наведено в таблиці 9.2.

Від значень, занесених в ці комірки, залежить також тривалість затримки скидання t_{out} .

9.1.2.1 Генератор з зовнішнім резонатором

Резонатор підключається до виводів XTAL1 і XTAL2 мікроконтролерів, як показано на рисунку 9.2. Ці виводи є відповідно входом і виходом інвертуючого підсилювача тактового генератора.

Ємності конденсаторів C1 і C2, що підключаються між выводами резонатора й спільним проводом, залежать від типу резонатора. Для кварцових резонаторів ємності цих конденсаторів зазвичай знаходяться в межах 12...22 пФ, а для керамічних повинні вибиратися відповідно до рекомендацій виробників резонаторів. Тепер розглянемо, яким чином здійснюється конфігурування тактового генератора із зовнішнім резонатором (далі, для простоти, кварцового генератора) в різних моделях.

ATmega8515x/8535x і ATmega8x/16x/32x/64x/128x. Підсилювач тактового генератора цих моделей може працювати в одному з двох режимів, що визначається станом конфігураційної комірки СКОРТ. Якщо ця комірка запрограмована (0), то розмах коливань на виході підсилювача (вивід XTAL2) практично дорівнює напрузі живлення. Даний режим корисний при роботі приладу в умовах сильних електромагнітних завад, а також при використанні сигналу тактового генератора для керування зовнішніми пристроями. В останньому випадку між виводом і зовнішньою схемою обов'язково повинен бути буфер.

Таблиця 9.2 – Вибір режиму роботи тактового генератора

Режим роботи	CKSEL3...0			
	ATmega8515x/8535x, ATmega8x/16x/32x/ 64x/128x	ATmega162x	ATmega165x, ATmega325x/3250x, ATmega645x/6450x	ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega640x, ATmega1280x/1281x
Економічний кварцовий генератор	1111...1010	-	-	1111...1000
Кварцовий генератор		1111...1000	1111...1000	0111...0110
Низькочастотний кварцовий генератор	1001	0111...0100	0111...0110	0101...0100
Генератор з зовнішнім RC-ланцюгом	1000...0101	-	-	-
Внутрішній RC-генератор на 128 кГц	-	-	-	0011
Внутрішній калібрований RC-генератор ¹⁾	0100...0001	0010	0010	0010
Зовнішній сигнал синхронізації	0000	0000	0000	0000
Зарезервовано	-	0011, 0001	0011, 0001 0101, 0100	0001
¹⁾ Режим за замовчування.				

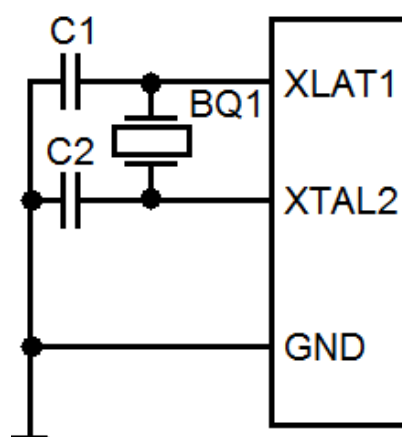


Рисунок 9.2 – Підключення кварцового чи керамічного резонатора

Якщо комірка СКОРТ не запрограмована (1), розмах коливань на виході підсилювача буде значно менше. Відповідно, струм споживання

мікроконтролера зменшується, однак при цьому звужується і діапазон можливих частот тактового сигналу. Крім того, в цьому режимі сигнал тактового генератора не можна використовувати для керування зовнішніми пристроями.

Власне генератор може працювати в чотирьох різних режимах, кожен з яких призначено для певного діапазону частот. Ці режими визначаються конфігураційними комірками CKSEL3...1 і SKOPT та зазначено в таблиці 9.3.

Таблиця 9.3 – Режими роботи кварцового генератора

SKOPT	CKSEL3...1	Приблизний діапазон частот (МГц)
1	101 ¹⁾	0.4...0.9
1	110	0.9...3.0
1	111	3.0...8.0
0	101,110,111	>8.0
¹⁾ В цьому режимі має використовуватися лише керамічний резонатор		

ATmega162x, ATmega165x, ATmega325x/3250x/645x/6450x.

Кварцовий генератор цих моделей також може працювати в чотирьох різних режимах, що визначаються станом конфігураційних комірок CKSEL3...1. Діапазони частот, для яких призначений той чи інший режим, наведено в таблиці 9.4.

ATmega48x/88x/168x, ATmega164x/324x/644x, Tmega640x/1280x/1281x/2560x/2561x. Кварцовий генератор цих моделей може працювати в п'яти різних режимах. Причому в першому режимі розмах коливань на виході генератора практично дорівнює напрузі живлення, а інші чотири режими є «економічними», тобто розмах коливань на виході генератора буде набагато менше. Як вже згадувалося, в такому разі зменшується струм споживання мікроконтролера, однак при цьому мікроконтролер стає

більш чутливим до зовнішніх завад. Використовуваний режим задається конфігураційними комірками CKSEL3...1 (таблиця 9.5).

Таблиця 9.4 – Режими роботи кварцового генератора

CKSEL3...1	Приблизний діапазон частот (МГц)
100 ¹⁾	0.4...0.9
101	0.9...3.0
110	3.0...8.0
111	>8.0
¹⁾ В цьому режимі має використовуватися лише керамічний резонатор	

Таблиця 9.5 – Режими роботи кварцового генератора

CKSEL3...1	Приблизний діапазон частот (МГц)
011 ¹⁾	0.4...16
100 ²⁾	0.4...0.9
101	0.9...3.0
110	3.0...8.0
111	>8.0
¹⁾ Звичайний (неекономічний) режим	
²⁾ В цьому режимі має використовуватися лише керамічний резонатор	

9.1.2.2 Низькочастотний кварцовий генератор

Цей режим призначено для використання низькочастотного кварцового резонатора на частоту 32 768 Гц, так званого «годинникового кварцу». Як і інші зовнішні резонатори, він підключається до виводів TOSC1 та TOSC2 мікроконтролерів.

У деяких моделях є внутрішні конденсатори, які можна підключити між виводами резонатора й спільним проводом. При цьому потреба в зовнішніх конденсаторах, природно, відпадає. У моделях ATmega8515x/8535x і ATmega8x/16x/32x/64x128x конденсатори ємністю 36 пФ підключаються при записі 0 в конфігураційну комірку SKOPT, а в моделях ATmega162x конденсатори підключаються, якщо в конфігураційних комірках CKSEL3...0 записано значення 0110 або 0111. Ємність кожного з конденсаторів становить 10 пФ. В інших моделях необхідно використовувати зовнішні конденсатори.

9.1.2.3 Зовнішній сигнал синхронізації

Сигнал від зовнішнього джерела подається на вивід XTAL1, як показано на рисунку 9.3. Зрозуміло, що цей сигнал повинен задовольняти вимогам мікроконтролера за частотою, шпаруватістю та рівнями напруги. Вивід XTAL2 в цьому режимі залишають непідключеним.

У моделях ATmega8x/16x/32x/64x/128x і ATmega8515x/8535x між виводом XTAL1 і спільним проводом можна включити внутрішній конденсатор ємністю 36 пФ. Це здійснюється записом 0 в конфігураційну комірку SKOPT.

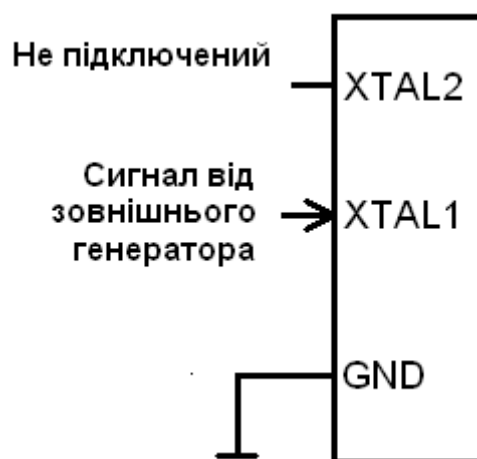


Рисунок 9.3 – Підключення зовнішнього джерела тактового сигналу

9.1.2.4 Генератор з зовнішнім RC-ланцюгом

При реалізації додатків, які не потребують високої часової точності, можна використовувати найпростіший RC-генератор. При цьому зовнішній RC-ланцюг підключається до виводу XTAL1, як показано на рисунку 9.4.

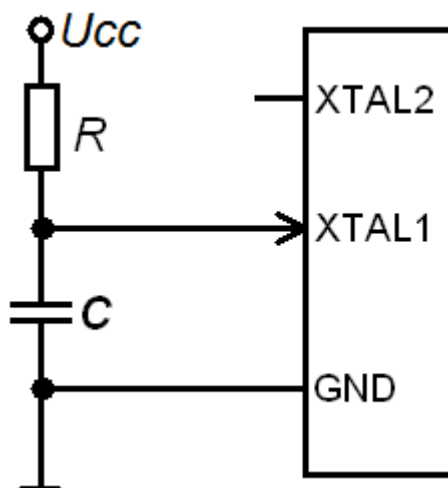


Рисунок 9.4 – Підключення зовнішнього RC-ланцюга

Ємність конденсатора ланцюжка повинна бути не менше 22 пФ, а опір резистора рекомендується вибирати з діапазону 3.3...100 кОм. Орієнтовно частоту сигналу генератора можна розрахувати за формулою:

$$f = \frac{1}{3 \cdot RC}.$$

Зовнішній конденсатор в цих моделях можна виключити, задіявши внутрішній ємністю 36 пФ. Як вже було зазначено, внутрішній конденсатор підключається при записі 0 в конфігураційну комірку СКОРТ.

Як і у випадку кварцового генератора, при використанні зовнішнього RC-ланцюжка тактовий генератор може працювати в чотирьох різних режимах, кожен з яких оптимізовано для певного діапазону частот. Ці режими визначаються вмістом конфігураційних комірок CKSEL3...0 згідно таблиці 9.6.

Таблиця 9.6 – Режими роботи генератора з зовнішнім RC-ланцюгом

CKSEL3...0	Приблизний діапазон частот (МГц)
0101 ¹⁾	0...0.9
0110	0.9...3.0
0111	3.0...8.0
1000	8.0...12.0
¹⁾ Режим за замовчуванням.	

9.1.2.5 Внутрішній калібрований RC-генератор

Використання вбудованого RC-генератора з внутрішнім часозадаючим RC-ланцюгом (внутрішнього RC-генератора) є найбільш

економічним рішенням, так як при цьому не потрібні ніякі зовнішні компоненти.

Номинальні частоти внутрішнього RC-генератора для різних моделей наведено в таблиці 9.7.

Таблиця 9.7 – Номинальні частоти внутрішнього RC-генератора

Модель	Частота [МГц]	Зауваження
ATmega8515х/8535х	1.0,2.0,4.0,8.0	При U _{cc} = 5.0 В, T=+25°C
ATmega8х/16х/32х/64х/128х	1.0,2.0,4.0,8.0	
ATmega48х/88х/168х	8.0	При U _{cc} = 5.0 В, T=+25°C
ATmega162х	8.0	
ATmega 164х/324х/644х	8.0	
ATmega 165х/325х/3250х/645х/6450х	8.0	
ATmega640х/1280х/1281х/2560х/2561х	8.0	

Як видно з таблиці, в деяких моделях внутрішній RC-генератор може працювати на декількох фіксованих частотах. Робоча частота генератора цих моделей визначається вмістом конфігураційних комірок CKSEL3...0 згідно таблиці 9.8.

Таблиця 9.8 – Режими роботи внутрішнього RC-генератора

CKSEL3...0	Приблизний діапазон частот(МГц)
0001 ¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0
¹⁾ Режим за замовчуванням.	

Слід зазначити, що при роботі з внутрішнім RC-генератором в конфігураційну комірку СКОРТ (якщо вона є в мікроконтролері) повинно бути записано 1. У всіх мікроконтролерах сімейства передбачена

можливість підстроювання частоти внутрішнього генератора (так зване калібрування). Для цієї мети використовується регістр OSCCAL, розташування якого в просторі регістрів введення/виведення для різних моделей наведено в таблиці 9.9. Чим більше значення, яке записано в регістрі OSCCAL, тим більша частота генератора. Діапазон можливої зміни частоти RC-генератора в залежності від вмісту цього регістра зазначений у таблиці 9.10. Значення, необхідне для налаштування генератора на номінальну частоту з точністю не гірше $\pm 1\%$ при температурі $+25^{\circ}\text{C}$ (для всього діапазону робочих температур це значення відповідає точності не гірше $\pm 3\%$), записується при виготовленні мікроконтролера в спеціальні калібрувальні комірки, кількість яких дорівнює кількості номінальних частот внутрішнього RC-генератора даної моделі [2].

Таблиця 9.9 – Адреса регістра OSCCAL

Модель	Адреса регістра OSCCAL
ATmega8515x	\$04(\$24)
ATmega8535x	\$31 (\$51)
ATmega8x	\$31 (\$51)
ATmega16x/32x	\$31(\$51) ¹⁾
ATmega64x/128x	(\$6F)
ATmega48x/88x/168x	(\$66)
ATmega162x	\$04(\$24) ¹⁾
ATmega164x/324x/644x	(\$66)
ATmega165x/325x/3250x/645x/6450x	(\$66)
ATmega640x/1280x/1281x/2560x/2561x	(\$66)
1) Регістр OSCCAL доступний з програми за вказаною адресою тільки в тому випадку, якщо в конфігураційну комірку OCDEN записано 1.	

Таблиця 9.10 – Вплив вмісту регістра OSCCAL на частоту внутрішнього RC-генератора

Вміст регістра OSCCAL	fmin (у відсотках від номінальної)	fmax (у відсотках від номінальної)
\$00	50	100
\$7F	75	150
\$FF	100	200

Ці комірки доступні тільки в режимі програмування мікроконтролерів. Завантаження калібрувальної константи для частоти 1МГц здійснюється апаратно при кожному включенні живлення. Якщо ж потрібна інша частота внутрішнього RC-генератора, відповідне значення має завантажуватися в регістр OSCCAL програмним шляхом. Для цього треба обчислити вміст калібрувальних комірок і записати його в заздалегідь зазначене місце у FLASH-пам'яті програм або EEPROM-пам'яті даних. А на самому початку програми слід прочитати вміст за цією адресою і завантажити його в регістр OSCCAL.

Слід пам'ятати, що внутрішній генератор призначений для роботи на номінальних частотах. Тому налаштування на інші частоти хоч і можливе, але не гарантується. Більше того, внутрішній RC-генератор визначає часові параметри доступу до FLASH- і EEPROM-пам'яті, тому збільшення частоти генератора більше ніж на 10% може призвести до неможливості запису в ці області пам'яті.

9.1.2.6 Внутрішній RC-генератор на 128 кГц

У моделях ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega640x і ATmega1280x/1281x/2560x/2561x в якості тактового можна використовувати ще один внутрішній RC-генератор, що має фіксовану

частоту 128 кГц. Для вибору цього джерела тактового сигналу в конфігураційних комірках CKSEL3...0 повинно бути записано значення 0011.

9.1.2.7 Керування тактовою частотою

У деяких моделях сімейства є можливість програмного зменшення частоти сигналу, що надходить від тактового генератора. Зрозуміло, що одночасно зі зменшенням тактової частоти зменшуються частоти сигналів f_{clkCPU} , $f_{\text{clkI/O}}$, f_{clkFLASH} , f_{clkADC} , тобто сповільнюється робота всіх периферійних пристроїв мікроконтролера. Якщо асинхронний таймер/лічильник працює в синхронному режимі, то відповідним чином змінюється і частота сигналу f_{clkASY} .

Для керування попереднім дільником тактового сигналу використовується один з регістрів введення/виведення. Назва цього регістра і його адреса для різних моделей наведено в таблиці 9.11.

У мікроконтролерах ATmega64x і ATmega128x для керування попереднім дільником тактового сигналу призначено регістр введення/виведення XDIV, який розташовано за адресою \$3C(\$5C). Формат цього регістра наведено на рисунку 9.5.

Старший розряд регістра (XDIVEN) служить для увімкнення/вимкнення попереднього дільника тактового сигналу, а інші розряди (XDIV6...0) визначають тактову частоту мікроконтролера. Якщо позначити десятковий еквівалент вмісту розрядів XDIV6...0 як d , залежність тактової частоти від стану цих розрядів буде визначатися виразом [2] :

$$f_{\text{CLK}} = \frac{\text{Частота джерела}}{129 - d}.$$

Таблиця 9.11 – Регістри керування попереднім дільником тактового сигналу

Модель	Регістр	Адрес
ATmega64x/128x	XDIV	\$3C (\$5C)
ATmega48x/88x/168x	CLKPR	(\$61)
ATmega162x	CLKPR	(\$61)
ATmega 164x/324x/644x	CLKPR	(\$61)
ATmega 165x/325x/3250x/645x/6450x	CLKPR	(\$61)
ATmega640x/1280x/1281x/2560x/2561x	CLKPR	(\$61)

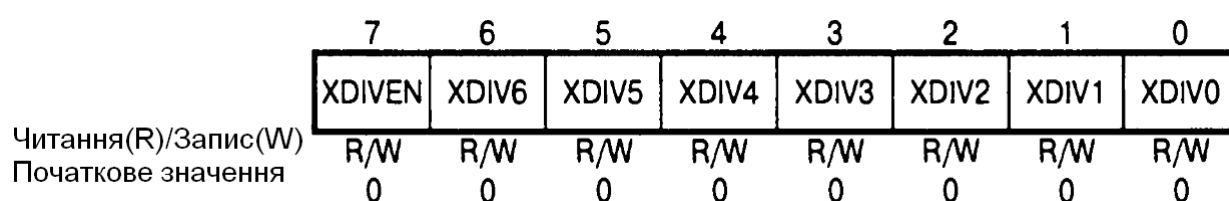


Рисунок 9.5 – Формат регістра XDIV

Зміна розрядів XDIV6...0 можлива тільки при скинутому розряді XDIVEN. При установці його в 1 тактова частота мікроконтролера буде визначатися виразом, наведеними вище. При скинутому в 0 розряді XDIVEN вміст розрядів XDIV6...0 ігнорується.

В інших моделях для керування попереднім дільником тактового сигналу призначений регістр CLKPR, розташований за адресою (\$61) в просторі додаткових регістрів введення/виведення. Формат цього регістра наведено на рисунку 9.6.

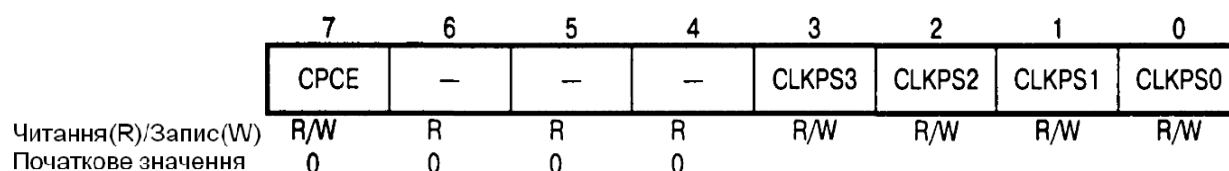


Рисунок 9.6 – Формат регістра CLKPR

Старший розряд (CPCE) служить для дозволу зміни частоти тактового сигналу, а розряди CLKPS3...CLKPS0 задають коефіцієнт ділення попереднього дільника (таблиця 9.12).

Таблиця 9.12 – Вибір коефіцієнта ділення попереднього дільника тактового сигналу

CLKPS3	CLKPS2	CLKPS1	CLKPS0	Коефіцієнт ділення
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	Зарезервовано
...				
1	1	1	1	

Для зміни вмісту розрядів CLKPS3...0 слід виконати наступні дії:

1. Записати в розряд CPCE лог. 1, а в розряди CLKPS3...0 - лог. 0.
2. Протягом наступних чотирьох тактів занести необхідні значення в розряди CLKPS3...0, при цьому розряд CPCE буде скинуто в 0. Інакше розряд CPCE буде скинуто апаратно після закінчення чотирьох тактів, забороняючи подальшу зміну розрядів CLKPS3...0.

Початковий стан розрядів CLKPS3...0 визначається конфігураційною коміркою CKDIV8. Якщо вона не запрограмована (1), то при запуску мікроконтролера в розрядах CLKPS3...0 буде знаходитися значення 0000.

Якщо ж комірка CKDIV8 запрограмована (0), стартовим значенням розрядів CLKPS3...0 є 0011 (коефіцієнт ділення - 8).

9.2 Режим зниженого енергоспоживання

9.2.1 Загальні відомості

Всі мікроконтролери сімейства Mega мають декілька (до шести) режимів зниженого енергоспоживання, що називаються одним терміном – «сплячий» режим. Кожен з цих режимів дозволяє знизити енергоспоживання мікроконтролера в періоди його бездіяльності. Вхід в будь-який з цих режимів виконується командою SLEEP. При виході мікроконтролера з «сплячого» режиму виконання програми продовжується з місця зупинки [1,2].

Режим зниженого енергоживлення надають користувачеві широкі можливості щодо зменшення струму, який споживається кристалом. Для цього у будь-якої моделі сімейства є декілька (від 3 до 6) режимів зниженого споживання, при переході в які виконання програми припиняється, а поновлюється при настанні певних подій. У нових моделях, крім того, передбачено менш радикальний спосіб зниження енергоспоживання кристала - відключення тактових сигналів незадіяних периферійних модулів.

9.2.2 Керування режимами зниженого енергоспоживання (сплячими режимами)

Різні моделі мікроконтролерів сімейства підтримують від 3 до 6 режимів зниженого енергоспоживання (таблиця 9.13). Режими відрізняються числом периферійних пристроїв мікроконтролера,

функціонуючих під час «сну» мікроконтролера, і відповідно ступенем зменшення енергоспоживання.

Залежно від моделі для керування «сплячим» режимом використовується різне число регістрів введення/виведення, які наведено в таблиці 9.14. Формати цих регістрів наведено на рисунках 9.7...9.10 (розряди, які не мають відношення до керування енергоспоживанням, виділені сірим кольором).

Таблиця 9.13 – Режими зниженого енергоспоживання

Режими зниженого енергоспоживання	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x/64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x, ATmega325x/3250x, ATmega645x/6450x	ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x
Idle	•	•	•	•	•	•	•	•	•
ADC Noise Reduction		•	•	•	•		•	•	•
Power Down	•	•	•	•	•	•	•	•	•
Power Save		•	•	•	•	•	•	•	•
Standby	•	•	•	•	•	•	•	•	•
Extended Standby		•		•		•	•		•

Для керування «сплячим» режимом в мікроконтролерах сімейства, в залежності від моделі, використовується два або чотири розряди регістрів введення/виведення. Призначення цих розрядів наведено в таблиці 9.15.

Таблиця 9.14 – Регістри для керування «сплячим» режимом

Назва	Опис	Адреса	Рис.	Модель
MCUCR	Регістр керування мікроконтролера	\$35(\$55)	7	ATmega8x/16x/32x/64x/128x, ATmega8535x
MCUCR	Регістр керування мікроконтролера	\$35(\$55)	7	ATmega8515x, ATmega162x
MCUCSR	Регістр керування і стану мікроконтролера	\$34(\$54)	8	
EMCUCR	Додатковий регістр керування мікроконтролера	\$36(\$56)	9	
SMCR	Регістр керування сплячим режимом	\$33(\$53)	10	ATmega48x/88x/168x, ATmega 164x/324x/644x/165x, ATmega325x/3250x/645x/6450x, ATmega640x, ATmega1280x/1281x/2560x/2561x

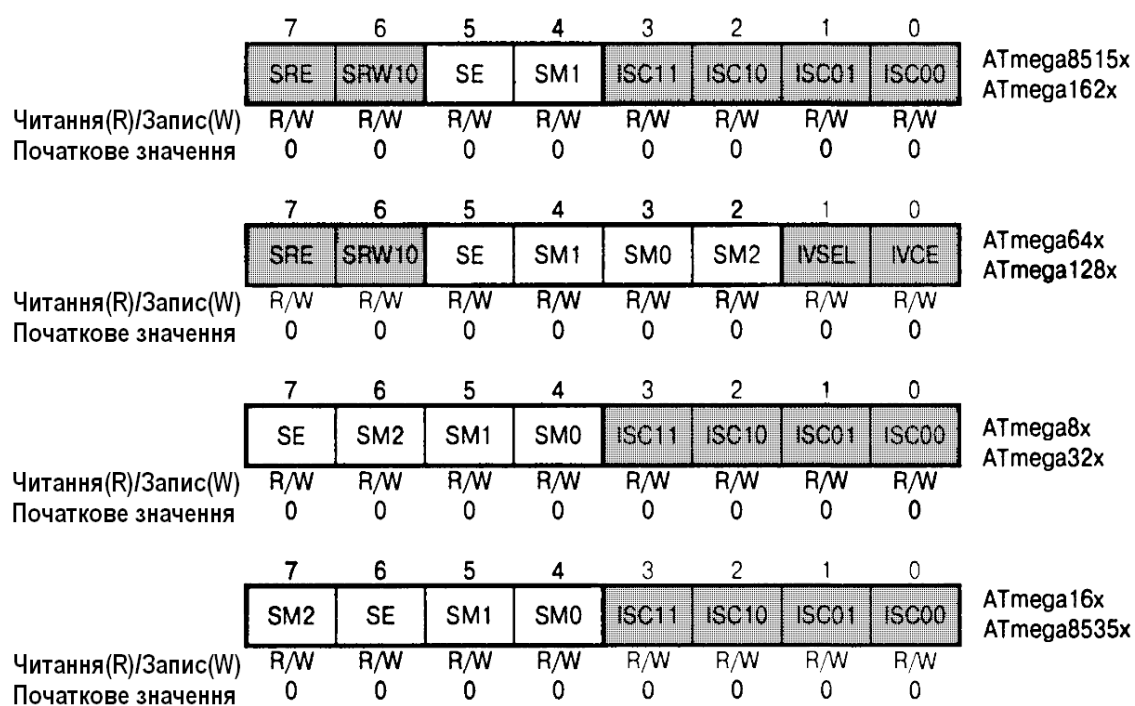


Рисунок 9.7 – Формат регістра MCUCR

	7	6	5	4	3	2	1	0	
	---	---	SM2	---	WDRF	BORF	EXTRF	PORF	ATmega8515x
Читання(R)/Запис(W)	R	R	R/W	R	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	JTD	—	SM2	JTRF	WDRF	BORF	EXTRF	PORF	ATmega162x
Читання(R)/Запис(W)	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

Рисунок 9.8 – Формат регістра MCUCSR моделей ATmega8515x і ATmega162x

	7	6	5	4	3	2	1	0	
	SM0	SRL2	SRL1	SRL0	SRW01	SRW00	SRW11	ISC2	ATmega8515x ATmega162x
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 9.9 – Формат регістра EMCUCR моделей ATmega8515x і ATmega162x

	7	6	5	4	3	2	1	0	
	—	—	—	—	SM2	SM1	SM0	SE	
Читання(R)/Запис(W)	R	R	R	R	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

Рисунок 9.10 – Формат регістра SMCR

Перехід в будь-який з режимів зниженого споживання здійснюється командою SLEEP. При цьому прапорець SE повинно бути встановлено в 1. Щоб уникнути ненавмисного перемикання мікроконтролера у «сплячий» режим рекомендується встановлювати цей прапорець безпосередньо перед виконанням команди SLEEP. Режим, в який перейде мікроконтролер після виконання команди SLEEP, визначається станом розрядів SM2...SM0. Відповідність між вмістом цих розрядів і режимом зниженого енергоспоживання наведено в таблиці 9.16.

Наявність того чи іншого режиму в конкретній моделі можна визначити за таблиці 9.13, що знаходиться на початку цього підрозділу. При відсутності в конкретній моделі мікроконтролера будь-якого з режимів значення розрядів SM2...SM0, відповідні цьому режиму, є зарезервовано.

Таблиця 9.15 – Призначення розрядів регістрів для керування «сплячим» режимом

Розряд	Опис
SE	Дозвіл переходу в режим зниженого енергоспоживання. Встановлення цього розряду в 1 дозволяє переведення мікроконтролера в режим зниженого енергоспоживання. Перемикання здійснюється за командою SLEEP. При скинутому розряді SE виконання команди не робить ніяких дій.
SM2...SM0	Вибір режиму зниженого енергоспоживання. Стан цих розрядів визначає в який режим перейде мікроконтролер після виконання команди SLEEP (таблиця 9.16).

Вихід з «сплячого» режиму може бути здійснено:

1. В результаті переривання. При генерації переривання мікроконтролер переходить у робочий режим, зупиняється на 4 такти, виконує підпрограму обробки переривання і відновлює виконання програми з інструкції, наступної за командою SLEEP. Вміст РЗП, ОЗП і РВВ при цьому не змінюється.
2. У результаті скидання. Після переходу мікроконтролера в робочий режим керування передається за адресою вектора скидання.

Таблиця 9.16 – Вибір режиму зниженого енергоспоживання

SM2	SM1	SM0	Режими
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power Down
0	1	1	Power Save
1	0	0	Зарезервовано
1	0	1	Зарезервовано
1	1	0	Standby ¹⁾
1	1	1	Extended Standby ¹⁾
¹⁾ Ці режими можна використовувати тільки при роботі із зовнішнім резонатором (три перші режими роботи тактового генератора згідно таблиці 2)			

Нижче розглядаються особливості окремих режимів зниженого енергозбереження.

Idle (режим холостого ходу). У цьому режимі припиняється формування тактових сигналів f_{clkCPU} і f_{clkFLASH} . При цьому ЦПП мікроконтролера зупиняється, а всі інші периферійні пристрої (інтерфейсні модулі, таймери/лічильники, аналоговий компаратор, АЦП, вартовий таймер), а також підсистема переривань продовжують функціонувати. Тому вихід з режиму Idle можливий як за зовнішнім, так і за внутрішнім перериванням. Якщо дозволено роботу АЦП, то перетворення почне виконуватися відразу ж після переходу в «сплячий» режим.

Основною перевагою режиму Idle є швидка реакція на події, що призводять до «пробудження» мікроконтролера. Іншими словами, виконання програми починається відразу ж після переходу з режиму Idle в робочий режим.

ADC Noise Reduction (режим зниження шумів АЦП). Даний режим є тільки в моделях, що містять у своєму складі модуль АЦП. У цьому режимі припиняє роботу ЦПП мікроконтролера і підсистема введення/виведення (відключаються тактові сигнали f_{clkCPU} , f_{clkFLASH} і $f_{\text{clkI/O}}$),

а АЦП, підсистема обробки зовнішніх переривань, вартовий таймер, асинхронний 8-розрядний таймер/лічильник, блок порівняння адреси модуля TWI і схема виявлення стану СТАРТ модуля USI продовжують функціонувати. За рахунок цього зменшуються завади на входах АЦП, що викликаються роботою ЦПП та підсистеми введення/виведення мікроконтролера, що, у свою чергу, дозволяє підвищити точність перетворення. Якщо АЦП включено, перетворення почне виконуватися відразу ж після переходу у «сплячий» режим.

Оскільки тактовий сигнал підсистеми введення/виведення $f_{\text{clkI/O}}$ у цьому режимі не генерується, повернення мікроконтролера в робочий режим може відбутися тільки в результаті скидання (апаратного, від вартового таймера, від схеми BOD) або в результаті генерації наступних переривань:

- переривання від вартового таймера;
- переривання від асинхронного таймера/лічильника;
- переривання за збігом адреси від інтерфейсу TWI;
- переривання за виявленням модулем USI стану СТАРТ;
- зовнішнього переривання, що виявляється асинхронно;
- переривання за зміною стану відповідних виводів;
- переривання від EEPROM-пам'яті і SPM-переривання;
- переривання від модуля АЦП.

Power Down (режим мікроспоживання). У режимі Power Down відключаються всі внутрішні тактові сигнали, відповідно припиняється функціонування всіх підсистем мікроконтролера, що працюють у синхронному режимі. Єдиними вузлами, що продовжують працювати в цьому режимі, є асинхронні модулі мікроконтролера: вартовий таймер (якщо він включений), підсистема обробки зовнішніх переривань, блок порівняння адреси модуля TWI і схема виявлення стану СТАРТ модуля USI. Відповідно, вихід з режиму Power Down можливий або в результаті

скидання (апаратного, від вартового таймера, від схеми BOD) або в результаті генерації наступних переривань:

- переривання за збігом адреси від інтерфейсу TWI;
- переривання за виявленням модулем USI стану СТАРТ;
- зовнішнього переривання (виявляється асинхронно).

Оскільки тактовий генератор мікроконтролера в режимі Power Down зупиняється, між настанням події, що приводить до «пробудження» мікроконтролера, і початком його роботи проходить деякий час, протягом якого тактовий генератор мікроконтролера виходить на робочий режим. Ця затримка визначається тими ж конфігураційними комірками, які задають затримку скидання мікроконтролера [2].

Також слід пам'ятати, що для «пробудження» мікроконтролера за зовнішнім перериванням та низьким рівнем, тривалість активного сигналу на вході мікроконтролера повинна бути не менше часу запуску мікроконтролера t_s [2]. Якщо сигнал, що викликав «пробудження» мікроконтролера, буде знято раніше, ніж мікроконтролер перейде в робочий режим, то обробник відповідного переривання викликано не буде.

Power Save (економічний режим). Цей режим практично ідентичний режиму Power Down, за винятком поведінки 8-розрядного таймера/лічильника, що підтримує роботу в асинхронному режимі. Крім того, цей сплячий режим у різних моделях сімейства реалізовано трохи по-різному. У моделях ATmega8535x, ATmega8x/16x/32x/64x/128x і ATmega162x асинхронний таймер/лічильник може працювати в асинхронному режимі під час «сну» мікроконтролера. А в інших моделях цей таймер/лічильник може працювати під час «сну» як в асинхронному, так і в синхронному режимі, причому в останньому випадку синхронний тактовий сигнал подається тільки на цей модуль мікроконтролера. Якщо синхронний режим роботи таймера/лічильника не задіяний, то при вході

мікроконтролера в режим Power Save основне джерело тактового сигналу зупиняється.

Таким чином, вихід з режиму Power Save можливий не тільки в наслідок подій, перерахованих при розгляді режиму Power Down, але і за перериванням від асинхронного таймера/лічильника. Зрозуміло, ці переривання повинно бути дозволено.

Standby (режим очікування). Цей режим рекомендується задіяти тільки при використанні в якості джерела тактового сигналу вбудованого генератора із зовнішнім резонатором. Режим Standby повністю ідентичний режиму Power Down, за винятком того, що тактовий генератор продовжує функціонувати. Завдяки цьому, перехід мікроконтролера в робочий режим відбувається набагато швидше – за 6 тактів.

Extended Standby (розширений режим очікування). Як і режим Standby, цей режим рекомендується задіяти тільки при використанні в якості джерела тактового сигналу вбудованого генератора із зовнішнім резонатором. Режим Extended Standby повністю ідентичний режиму Power Save, за винятком того, що тактовий генератор продовжує функціонувати. Тому інтервал між «пробудженням» мікроконтролера і виходом його в робочий режим складає всього 6 тактів. Основні відмінності різних режимів зниженого енергоспоживання наведено в таблиці 9.17.

9.2.3 Керування тактовими сигналами окремих модулів

У моделях ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega165x/325x/3250x/645x/6450x і ATmega640x/1280x/1281x/2560x/2561x крім енергозберігаючих режимів передбачено ще один метод зниження сумарного енергоспоживання кристала. Цей метод полягає у відключенні тактових сигналів тих периферійних модулів, які не задіяно в конкретному додатку.

При цьому стан периферійного модуля «заморожується», а його регістри стають недоступними для операцій читання/запису. Перед

відключенням тактового сигналу периферійного модуля бажано забороняти функціонування даного модуля. Це пов'язано з тим, що при відключенні тактового сигналу апаратні ресурси кристала, що використовуються периферійним модулем, залишаються в його розпорядженні.

Для керування локальними тактовими сигналами використовується один або два (залежно від моделі мікроконтролера) регістри введення/виведення, що наведено у таблиці 9.18. Формати цих регістрів наведено на рисунках 9.11...9.13.

Таблиця 9.17 – Відмінності різних режимів зниженого енергоспоживання

Режим	Активні внутрішні тактові сигнали					Генератори		Джерела «пробудження» мікроконтролера							
	f_{clkCPU}	$f_{clkFLASH}$	$f_{clkI/O}$	f_{clkADC}	f_{clkASY}	Основне джерело тактового сигналу	Генератор таймера/лічильника	Зовнішні переривання	Блок збігу адреси модуля TWI	Схема детектування стану START	Асинхронний 8-бітний таймер/лічильник	Готовність SPM/EEPROM	АЦП	Вартовий таймер	Інші модулі введення/виведення
Idle			•	•	•	•	• ²⁾	•	•	•	•	•	•	•	•
ADC Noise				•	•	•	• ²⁾	• ⁴⁾	•	•	•	•	•	•	
Power Down							• ²⁾	• ⁴⁾	•	•				•	
Power Save					• ³⁾		• ²⁾	• ⁴⁾	•	•	• ³⁾			•	
Standby ¹⁾						•		• ⁴⁾	•	•				•	
Extended Standby ¹⁾					• ³⁾	•	• ²⁾	• ⁴⁾	•		• ³⁾			•	
¹⁾ Джерелом тактового сигналу повинен бути кварцовий генератор. ²⁾ Якщо таймер/лічильник працює в асинхронному режимі. ³⁾ Якщо таймер/лічильник працює в асинхронному режимі (тільки моделі ATmega8535x, ATmega8x/16x/32x/64x/128x і ATmega162x) і при будь-якому режимі роботи таймера/лічильника в інших моделях. ⁴⁾ Тільки переривання, які виявляються асинхронно.															

Таблиця 9.18 – Регістри для керування тактовими сигналами модулів

Назва	Опис	Адреса	Рисунок	Модель
PRR	Регістр зниження споживаної потужності	(\$64)	11	ATmega48x/88x/168x, ATmega65x/325x/3250x/645x/6450x
PRR0	0-й регістр зниження споживаної потужності	(\$64)	12	ATmega 164x/324x/644x
PRR0	0-й регістр зниження споживаної потужності	(\$64)	12	ATmega640x, ATmega1280x/1281x/2560x/2561x
PRR1	1-й регістр зниження споживаної потужності	(\$65)	13	

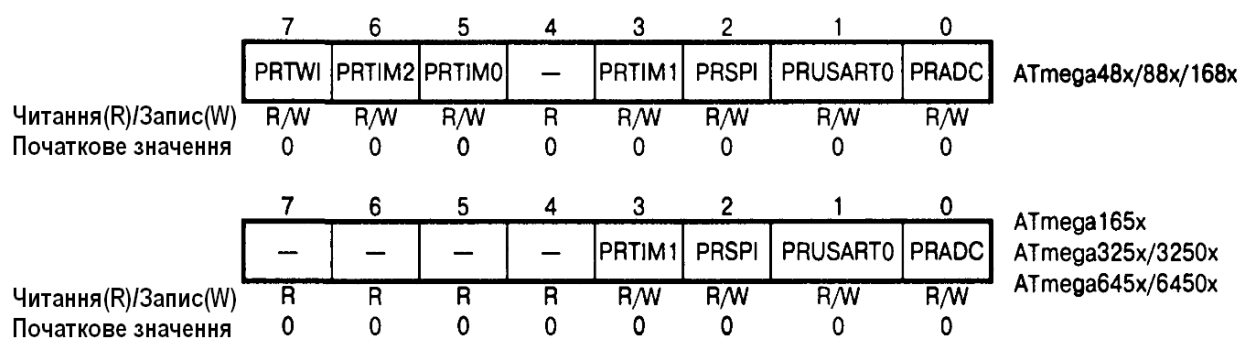


Рисунок 9.11 – Формат регістру PRR

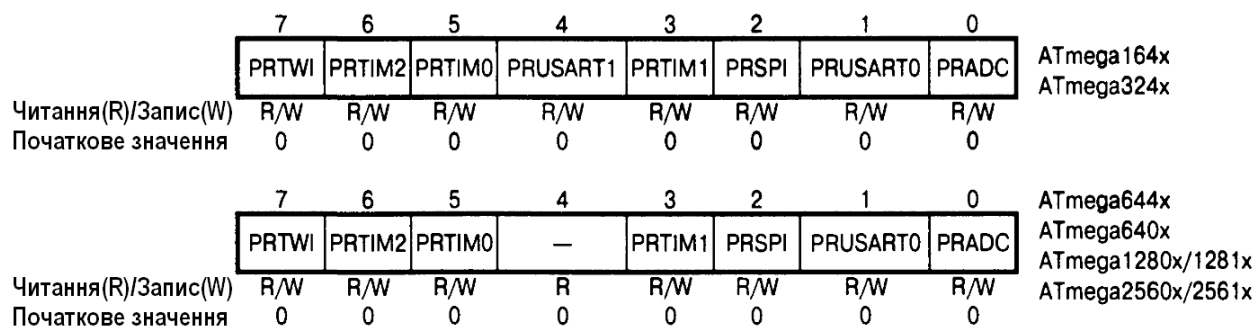


Рисунок 9.12 – Формат регістру PRR0

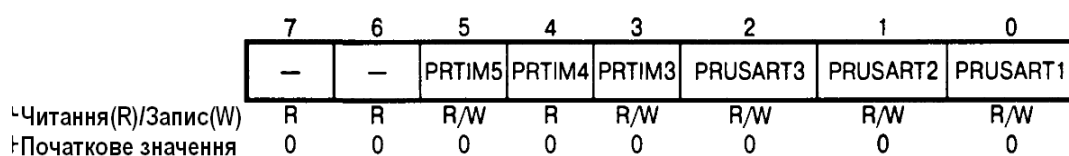


Рисунок 9.13 – Формат регістру PRR1

Призначення розрядів регістрів зниження споживаної потужності

ізних моделей наведено в таблицях 9.19 та 9.20. Використовувати відключення тактових сигналів модулів можна як при нормальному функціонуванні мікроконтролера, так і спільно з енергозберігаючим режимом Idle.

Таблиця 9.19 – Розряди регістра PRR і PRR0

Біт	Назва	Опис	Модель
7	PRTWI	Відключення модуля TWI. При встановленні цього розряду в 1 припиняється формування тактового сигналу модуля TWI, в результаті чого він вимикається. Після повторного включення модуль необхідно реініціалізувати	ATmega48x/88x/168x, ATmega 164x/324x/644x, ATmega640x, ATmega1280x/1281x, ATmega2560x//2561x
6	PRTIM2	Відключення таймера/лічильника T2. При встановленні цього розряду в 1 відключається таймер/ лічильник T2, який працює в синхронному режимі (розряд AS2 регістра ASSR скинутий в 0). Після повторного включення таймер/лічильник продовжить функціонування в колишньому режимі	
5	PRTIM0	Відключення таймера/лічильника T0. При встановленні цього розряду в 1 відключається таймер/ лічильник T0. Після повторного включення таймер/лічильник продовжить функціонування в колишньому режимі	
4	PRUSART1	Відключення модуля USART1. При встановленні цього розряду в 1 припиняється формування тактового сигналу 1-го модуля USART, в результаті чого він відключається. Після повторного включення модуль необхідно реініціалізувати	ATmega 164x/324x
	-	Зарезервований, читається як 0	Решта моделей
3	PRTIM1	Відключення таймера/лічильника T1. При встановленні цього розряду в 1 відключається таймер/ лічильник T1. Після повторного включення таймер/лічильник продовжить функціонування в колишньому режимі	Всі моделі
2	PRSPI	Відключення модуля SPI. При встановленні цього розряду в 1 припиняється формування тактового сигналу модуля SPI, в результаті чого він вимикається. Після повторного включення модуль необхідно реініціалізувати	Всі моделі
1	PRUSART0	Відключення модуля USART0. При встановленні цього розряду в 1 припиняється формування тактового сигналу 0-го модуля USART, в результаті чого він відключається. Після повторного включення модуль необхідно реініціалізувати	Всі моделі
0	PRADC	Відключення модуля АЦП. При встановленні цього розряду в 1 відключається АЦП. Перед відключенням функціонування АЦП необхідно заборонити. <i>Примітка.</i> При вимкненому АЦП аналоговий компаратор не може використовувати вхідний мультиплексор модуля АЦП	Всі моделі

Таблиця 9.20 – Розряди регістра PRR1

Розр.	Назва	Опис	Модель
7,6	-	Зарезервовані, читаються як 0	ATmega640x, ATmega1280x/1281x, ATmega2560x//2561x
5	PRTIM5	Відключення таймера/лічильника T5. При встановленні цього розряду в 1 відключається таймер/лічильник T5. Після повторного включення таймер/лічильник продовжить функціонування в колишньому режимі	
4	PRTIM4	Відключення таймера/лічильника T4. При встановленні цього розряду в 1 відключається таймер/лічильник T4. Після повторного включення таймер/лічильник продовжить функціонування в колишньому режимі	
3	PRTIM3	Відключення таймера/лічильника T3. При встановленні цього розряду в 1 відключається таймер/лічильник T3. Після повторного включення таймер/лічильник продовжить функціонування в колишньому режимі	
2	PRUSART3	Відключення модуля USART3. При встановленні цього розряду в 1 припиняється формування тактового сигналу 3-го модуля USART, в результаті чого він відключається. Після повторного включення модуль необхідно реініціалізувати	
1	PRUSART2	Відключення модуля USART2. При встановленні цього розряду в 1 припиняється формування тактового сигналу 2-го модуля USART, в результаті чого він відключається. Після повторного включення модуль необхідно реініціалізувати	
0	PRUSART1	Відключення модуля USART1. При встановленні цього розряду в 1 припиняється формування тактового сигналу 1-го модуля USART, в результаті чого він відключається. Після повторного включення модуль необхідно реініціалізувати	

9.2.4 ЗАГАЛЬНІ РЕКОМЕНДАЦІЇ ПО ЗМЕНШЕННЮ ЕНЕРГОСПОЖИВАННЯ

Якщо потрібно звести енергоспоживання мікроконтролера до мінімуму, то при розробці програми необхідно враховувати два моменти. По-перше, необхідно якомога активніше використовувати

енергозберігаючі режими, причому режим слід вибирати так, щоб у ньому функціонувало мінімальна необхідна кількість вузлів мікроконтролера. По-друге, повинно бути заборонено роботу всіх невикористовуваних модулів. Нижче перераховані модулі, на які рекомендується звернути більш пильну увагу під час вирішення завдання мінімізації сумарного енергоспоживання кристала [1,2].

Аналого-цифровий перетворювач. Якщо функціонування АЦП дозволено, то він буде працювати у всіх «Сплячих» режимах. Відповідно, для зниження споживаного струму модуль АЦП необхідно відключати перед переведенням мікроконтролера в будь-який з режимів енергозбереження.

Аналоговий компаратор. Якщо компаратор не використовується, то при переведенні мікроконтролера в режим Idle або ADC Noise Reduction його необхідно відключити. В інших «Сплячих» режимах модуль аналогового компаратора відключається автоматично. Тим не менше, якщо в якості одного з вхідних сигналів у компараторі використовується сигнал від внутрішнього ДОН, модуль аналогового компаратора необхідно відключати вручну перед переведенням мікроконтролера в будь-який з «сплячих» режимів. В іншому випадку ДОН залишиться працювати незалежно від «сплячого» режиму, збільшуючи сумарне споживання кристала.

Детектор зниженої напруги живлення. Якщо детектор зниженої напруги живлення (Brown-Out Detector - BOD) у пристрої не потрібен, цей вузол необхідно відключити. Якщо ж роботу цього вузла дозволено, то він буде працювати в будь-якому з «Сплячих» режимів, збільшуючи сумарне споживання кристала.

Внутрішнє джерело опорної напруги. Внутрішнє ДОН буде працювати, якщо включений хоча б один з наступних вузлів мікроконтролера: детектор зниження напруги живлення, аналоговий

компаратор, АЦП. При відключенні всіх цих модулів ДОН вимкнеться автоматично. Не забудьте, що при повторному включенні ДОН його вихідний сигнал можна буде використовувати тільки через деякий проміжок часу.

Вартовий таймер. Якщо вартовий таймер не потрібен в пристрої, його необхідно відключити. В іншому випадку він буде працювати в будь-якому з «сплячих» режимів, збільшуючи загальне споживання кристала.

Порти введення/виведення. Перед перемиканням в «сплячий» режим всі виводи портів введення/виведення необхідно сконфігурувати так, щоб вони споживали найменший струм. Головне – звести до мінімуму кількість ліній, які керують резистивним навантаженням. У всіх «сплячих» режимах, де припиняється формування тактових сигналів $f_{clkI/O}$ і f_{clkADC} , вхідні буфери портів введення/виведення відключаються. Виняток становлять тільки вхідні вузли тих ліній введення/виведення, які використовуються для виведення мікроконтролера з «сплячого» режиму. Якщо ж при включеному вхідному буфері вивід мікроконтролера залишиться непідключеним або на ньому буде бути присутнім аналоговий сигнал з напругою, близьким до $U_{cc}/2$, то буфер буде споживати надмірний струм.

У моделях, які дозволяють відключати цифрові вхідні буфери на аналогових входах, вони повинні завжди бути вимкнені.

Модуль внутрішньосхемного налагодження. Якщо внутрішньосхемне налагодження дозволено, то при переході мікроконтролера в «сплячий» режим основне джерело тактового сигналу не зупиняється, продовжуючи споживати струм. Вимкнути модуль внутрішньосхемного налагодження можна трьома способами:

- записати 1 в конфігураційну комірку OCDEN;
- записати 1 в конфігураційну комірку JTAGEN;
- записати 1 в розряд JTD відповідного регістра введення/виведення.

9.3 Скидання

9.3.1 Загальні відомості

Реініціалізація, або так зване «скидання», переводить мікроконтролер в початковий стан. Скидання може бути викликано наступними подіями:

- включення напруги живлення мікроконтролера;
- подача сигналу НИЗЬКОГО рівня на вивід RESET (апаратне скидання);
- тайм-аут вартового таймера;
- падіння напруги живлення нижче заданої величини;
- скидання по інтерфейсу JTAG.

При настанні будь-якої з перерахованих вище подій у всі регістри введення/виведення заносяться їх початкові значення, а в лічильник команд завантажуються значення адреси вектора скидання. За цією адресою має знаходитись команда безумовного переходу (RJMP – для моделей ATmega8515x/8535x, ATmega8x і ATmega48x/88x, JMP – для решти моделей) на початок програми. Якщо ж переривання в програмі не використовуються, то вона може починатися безпосередньо з адреси вектора скидання. Сказане справедливо і для випадку, коли вектор скидання розташовується в області основної програми, а таблиця векторів переривань – в області завантажувача.

У всіх моделях, крім ATmega48x, значення адреси вектора скидання визначається станом конфігураційної комірки BOOTRST. Якщо BOOTRST=1 (до програмування), то вектор скидання розташовується з самого початку пам'яті програм за адресою \$ 0000 (в моделях ATmega48x вектор скидання завжди розташовується за цією адресою). Після

програмування комірки вектор скидання розташовується на початку області завантажувача. Конкретне значення цієї адреси залежить від конфігураційних комірок BOOTSZ1 і BOOTSZ0, що визначають в тому числі і розмір області завантажувача. Залежність значення адреси вектора скидання від значень конфігураційних комірок BOOTSZ1 і BOOTSZ0 для всіх моделей сімейства наведено в таблиці 9.21.

Узагальнену структурну схему підсистеми скидання наведено на рисунку 9.14. Елементи, виділені пунктиром, у ряді моделей відсутні.

Логіка схеми скидання всіх мікроконтролерів сімейства наступна. При настанні події, що приводить до скидання мікроконтролера, формується внутрішній сигнал скидання. Одночасно запускається таймер формування затримки скидання. Після закінчення певного проміжку часу внутрішній сигнал скидання знімається і починається виконання програми. Всі мікроконтролери сімейства дозволяють визначити подію, в результаті якої відбулося скидання пристрою. У залежності від моделі (таблиця 9.22) для цього використовується або регістр керування і стану мікроконтролера MCUCSR, розташований за адресою \$34 (\$54), або регістр стану мікроконтролера MCUSR, розташований за тією ж адресою.

Таблиця 9.21 – Значення адреси вектору скидання в області завантажувача

BOOTSZ1	BOOTSZ0	ATmega8515x/8535x, ATmega8x, ATmega88x	ATmega6x, ATmega62x, ATmega64x, ATmega65x, ATmega68x	ATmega32x, ATmega324x, ATmega325x/3250x	ATmega64x, ATmega640x, ATmega644x, ATmega645x/6450x	ATmega28x, ATmega280x/1281x	ATmega2560x/2561x
1	1	\$F80	\$1F80	\$3F80	\$7E00	\$FE00	\$1FE01
1	0	\$F00	\$1F00	\$3E00	\$7C00	\$FC00	\$1FC00
0	1	\$E00	\$1E00	\$3C00	\$7800	\$F800	\$1F800
0	0	\$C00	\$1C00	\$3800	\$7000	\$F000	\$1F000

Наведені в таблиці 9.22 регістри містять набір прапорів, стан яких залежить від події, що викликали скидання пристрою. Формат регістрів MCUSR і MCUCSR наведено на рисунку 9.15 (розряди, не пов'язані з підсистемою скидання, виділені сірим кольором). Опис прапорців, які використовуються для визначення джерела скидання, наведено в таблиці 9.23.

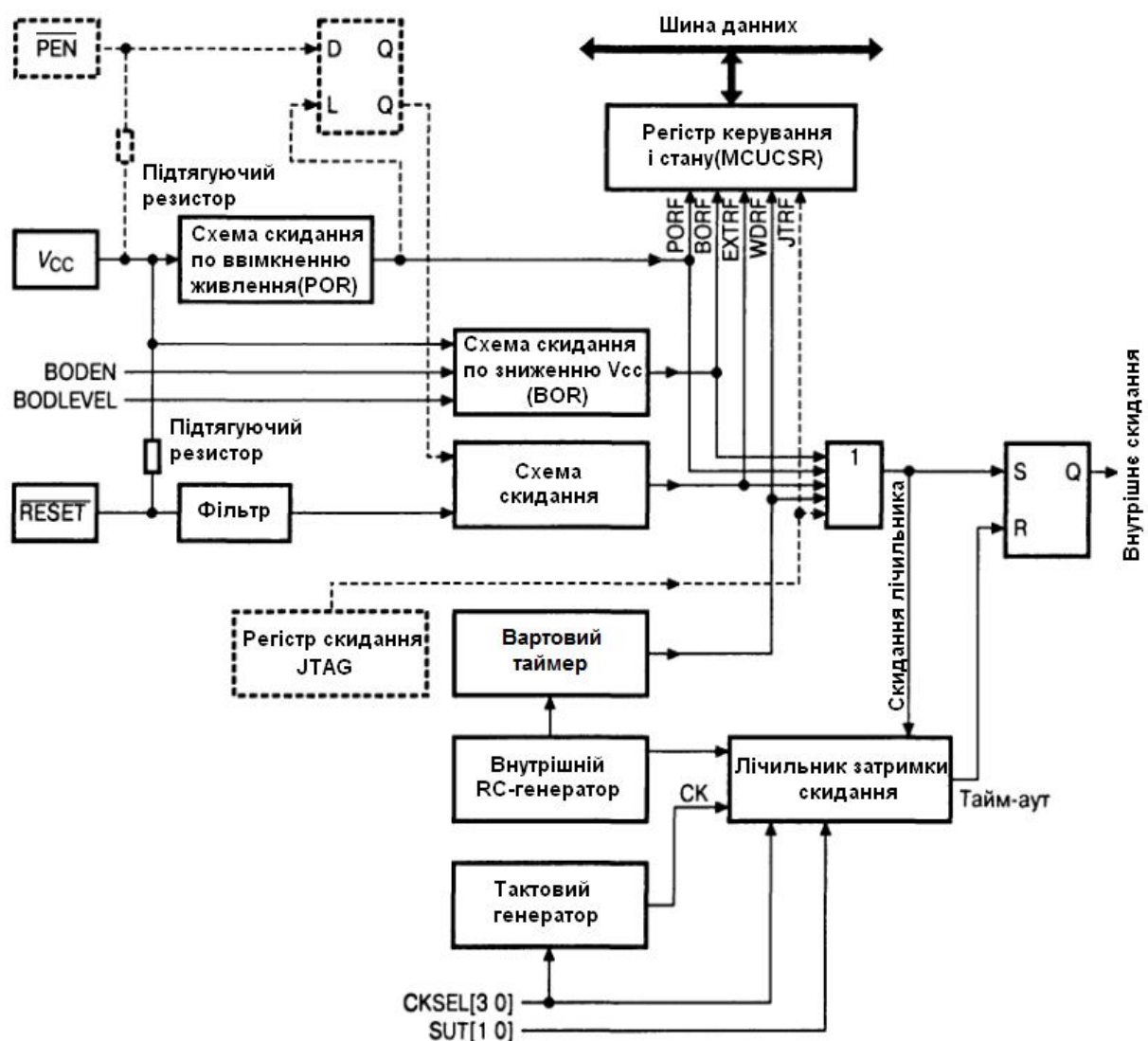


Рисунок 9.14 – Структурна схема підсистеми скидання

Таблиця 9.22 – Регістри для визначення джерела скидання

Назва	Опис	Адрес	Модель
MCUCSR	Регістр керування і стану мікроконтролера	\$34(\$54)	ATmega8515x/8535x, ATmega8x/16x/32x/64x/128x, ATmega162x
MCUSR	Регістр стану мікроконтролера	\$34 (\$54)	ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega165x/325x/3250x/645x/6450x, ATmega640x/1280x/1281x/2560x/2561x

	7	6	5	4	3	2	1	0	
	X	X	X	—	WDRF	BORF	EXTRF	PORF	ATmega8x ATmega8515x ATmega8535x
Читання(R)/Запис(W)	X	X	X	R	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	X	X	X	JTRF	WDRF	BORF	EXTRF	PORF	ATmega162x ATmega16x/32x ATmega64x/128x
Читання(R)/Запис(W)	X	X	X	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

a)

	7	6	5	4	3	2	1	0	
	—	—	—	—	WDRF	BORF	EXTRF	PORF	ATmega88x/48x/168x
Читання(R)/Запис(W)	R	R	R	R	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
	—	—	—	JTRF	WDRF	BORF	EXTRF	PORF	ATmega164x/324x/644x ATmega165x ATmega325x/3250x ATmega645x/6450x ATmega640x ATmega1280x/1281x ATmega2560x/2561x
Читання(R)/Запис(W)	R	R	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	

б)

Рисунок 9.15 – Формат регістрів MCUCSR (a) і MCUSR (б)

Таблиця 9.23 – Прапорці джерел скидання регістрів MCUCSR і MCUSR

Назва прапорця	Опис
JTRF	Прапорець JTAG-скидання. Встановлюється в 1, якщо скидання сталося в результаті команди JTAG AVR_RESET. Розряд скидається в результаті скидання по живленню або безпосереднім записом у нього лог. 0
WDRF	Прапорець скидання від вартового таймера. Встановлюється в 1, якщо джерелом скидання був вартовий таймер. Розряд скидається в результаті скидання по живленню або безпосереднім записом у нього лог. 0
BORF	Прапорець скидання щодо зниження живлення. Встановлюється в 1, якщо джерелом скидання була підсистема BOD. Розряд скидається в результаті скидання по живленню або безпосереднім записом у нього лог. 0
EXTRF	Прапорець апаратного скидання. Встановлюється в 1, якщо скидання сталося в результаті подачі на вивід скидання сигналу НИЗЬКОГО рівня. Розряд скидається в результаті скидання по живленню або безпосереднім записом у нього лог. 0
PORF	Прапорець скидання по включенню живлення. Встановлюється в 1 після подачі напруги живлення на мікроконтролер. Розряд скидається тільки безпосереднім записом у нього лог. 0

9.3.2 ОПИС ОКРЕМИХ ВИДІВ СКИДАННЯ

9.3.2.1 СКИДАННЯ ПО ВВІМКНЕННЮ ЖИВЛЕННЯ

До складу всіх мікроконтролерів сімейства Mega входить система скидання по ввімкненню живлення (схема POR – Power-on Reset). Ця схема утримує мікроконтролер у стані скидання до тих пір, поки напруга живлення не перевищить деякого порогового значення U_{POT} . При досягненні напругою живлення значення U_{POT} схема POR запускає таймер затримки скидання. Після закінчення лічби (після формування затримки t_{TOUT} , рисунок 9.16) внутрішній сигнал скидання знімається і відбувається запуск мікроконтролера.

Керування виводом $\overline{RESET}$ мікроконтролера при включенні живлення може здійснюватися двома способами. Якщо час наростання напруги джерела живлення відомий і не перевищує величини t_{TOUT} можна використовувати спосіб, при якому напруга на виводі $\overline{RESET}$ «повторює»

напругу живлення. Відповідні даному способу часові діаграми показані на рисунку 9.16. Для реалізації цього способу вивід $\overline{\text{RESET}}$ можна підключити до джерела живлення або залишити непідключеним, оскільки він вже підтягнутий до джерела живлення внутрішнім резистором (рисунок 9.14).

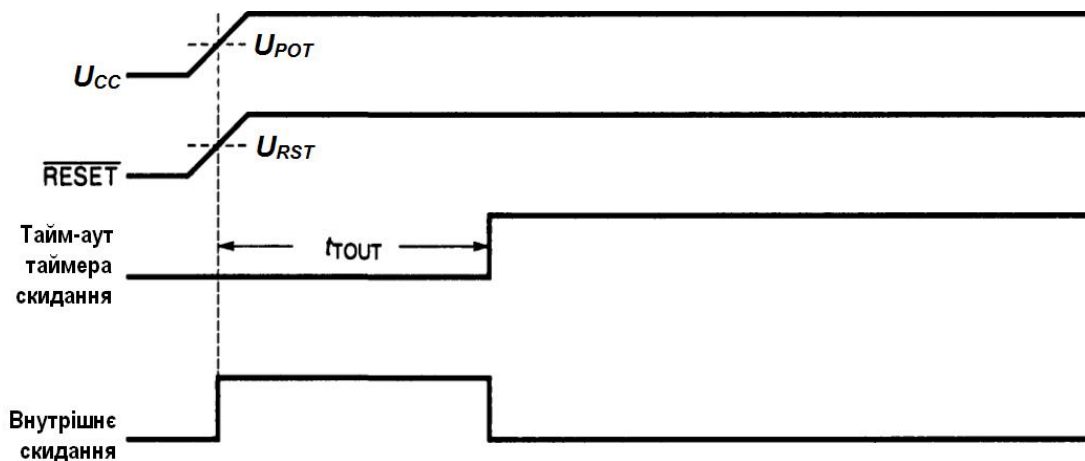


Рисунок 9.16 – Часові діаграми роботи при скиданні по ввімкненню живлення; вивід $\overline{\text{RESET}}$ підключений до V_{DD}

При іншому способі вивід $\overline{\text{RESET}}$ керується зовнішньою схемою, і сигнал ВИСОКОГО рівня подається на нього лише після встановлення напруги живлення. Часові діаграми, що відповідають цьому способу, показано на рисунку 9.17.

У цьому випадку роботою таймера затримки скидання буде керувати схема апаратного скидання, і він почне працювати при досягненні напругою на виводі $\overline{\text{RESET}}$ порогового значення U_{RST} .

Дане рішення є більш дорогим, оскільки вимагає застосування зовнішніх компонентів. Однак цей спосіб дозволяє «підганяти» час запуску мікроконтролера під час наростання напруги використовуваного джерела живлення.

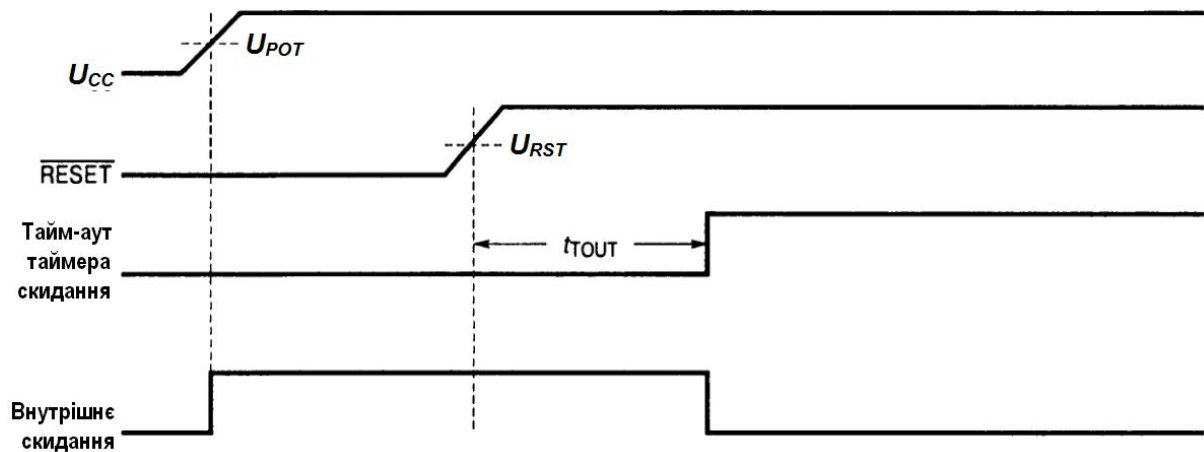


Рисунок 9.17 – Часові діаграми роботи при скиданні по ввімкненню живлення; вивід $\overline{\text{RESET}}$ керується зовнішньою схемою

9.3.2.2 Апаратне скидання

Апаратне (або зовнішнє) скидання мікроконтролера здійснюється подачею на вивід $\overline{\text{RESET}}$ сигналу НИЗЬКОГО рівня. Мікроконтролер залишається в стані скидання до тих пір, поки на виводі $\overline{\text{RESET}}$ буде присутній сигнал НИЗЬКОГО рівня. Мінімальна тривалість імпульсу, при якій гарантується скидання мікроконтролера, для різних моделей різна, але не перевищує 3.0 мкс (ATmega162x). При досягненні напругою на виводі $\overline{\text{RESET}}$ порогового значення U_{RST} запускається таймер затримки скидання. Після формування затримки t_{TOUT} внутрішній сигнал скидання знімається і відбувається запуск мікроконтролера. Часові діаграми сигналів при апаратному скиданні показано на рисунку 9.18.

9.3.2.3 Скидання від ВАРТОВОГО ТАЙМЕРА

За тайм-аутом вартового таймера (якщо він включений) формується короткий додатний імпульс і генерується внутрішній сигнал скидання (рисунок 9.19).

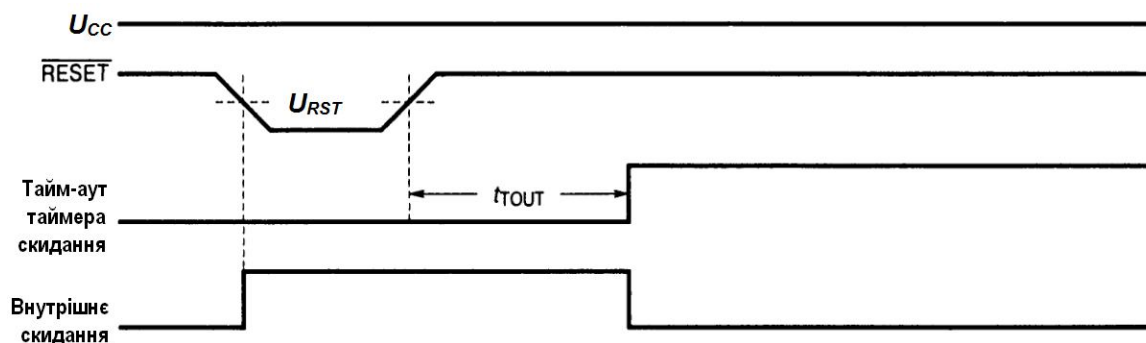


Рисунок 9.18 – Часові діаграми роботи при апаратному скиданні

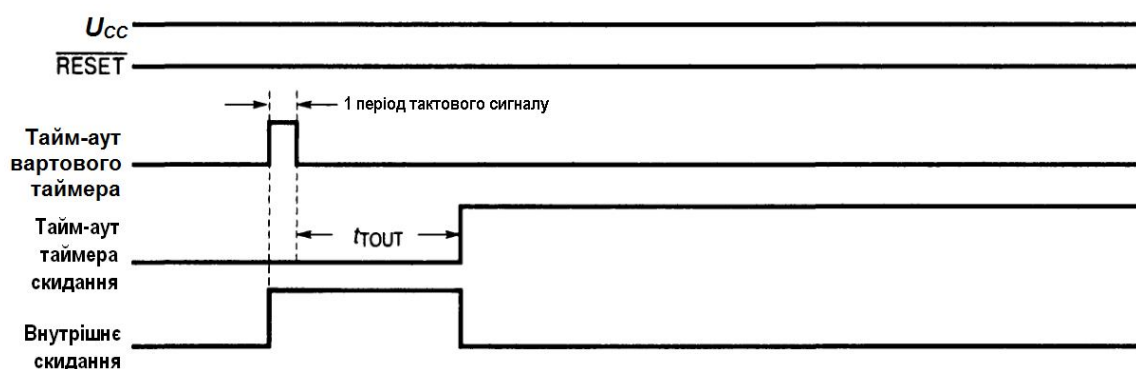


Рисунок 9.19 – Часові діаграми роботи при скиданні від вартового таймера

Тривалість імпульса дорівнює одному періоду тактового сигналу мікроконтролера. За спадаючим фронтом цього імпульсу запускається таймер затримки скидання. Після формування затримки t_{OUT} внутрішній сигнал скидання знімається і відбувається запуск мікроконтролера.

9.3.2.4 Скидання при зниженні напруги живлення

Усі моделі мікроконтролерів сімейства Mega мають у своєму складі схему BOD (Brown-Out Detection), яка відстежує рівень напруги джерела живлення (рисунок 9.20).

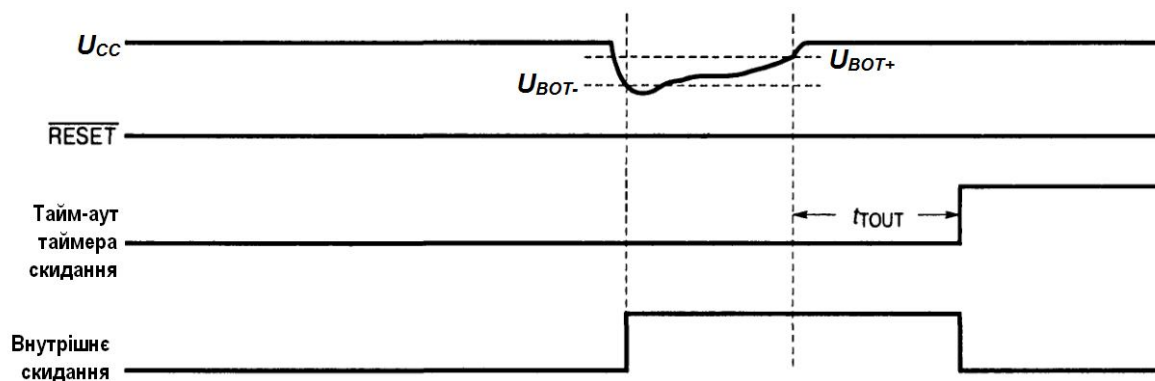


Рисунок 9.20 – Часові діаграми роботи при скиданні по зниженню напруги живлення

Якщо роботу цієї схеми дозволено, то при зниженні напруги живлення нижче деякого значення U_{BOD-} вона переводить мікроконтролер в стан скидання. Коли напруга живлення знов збільшиться до порогового значення U_{BOD+} , запускається таймер затримки скидання. Після формування затримки t_{TOUT} внутрішній сигнал скидання знімається і відбувається запуск мікроконтролера. Часові діаграми, відповідні скиданню від схеми BOD, показано на рисунку 9.20. Для зменшення ймовірності помилкових спрацьовувань поріг перемикання схеми має гістерезис щодо порогу спрацьовування U_{BOD} , який дорівнює 50мВ ($U_{BOD+} = U_{BOD} + 25\text{мВ}$, $U_{BOD-} = U_{BOD} - 25\text{мВ}$). Крім того, спрацьовування схеми BOD відбудеться тільки в тому випадку, якщо тривалість провалу напруги живлення буде більше 2мкс.

У всіх моделях включенням/виключенням схеми BOD керує конфігураційна комірка BODEN. Для дозволу роботи схеми ця комірка повинна бути запрограмована (0). Для задання порогу спрацьовування схеми BOD в різних моделях використовується різна кількість конфігураційних комірок [2].

ATmega8515x/8535x, ATmega8x/16x/32x/64x/128x. У цих моделях поріг спрацьовування визначається станом конфігураційної комірки BODLEVEL. Якщо в цій комірці записано 1, поріг спрацьовування

дорівнює 2.7 В. Якщо ж у ній записано 0 (після її програмування), поріг спрацьовування дорівнює 4.0 В.

ATmega325x/3250x/645x/6450x. У цих моделях для керування схемою BOD використовуються дві конфігураційні комірки BODLEVEL1...0. Вплив вмісту цих комірок на роботу схеми BOD показано в таблиці 9.24.

Таблиця 9.24 – Задання порога спрацьовування схеми BOD в моделях
ATmega325x/3250x/645x/6450x

BODLEVEL1...0	Поріг спрацьовування U_{BOT}
11	Схема BOD вимкнена
10	1.8
01	2.7
00	4.3

ATmega162x. У моделях ATmega162(V) для керування схемою BOD використовуються вже три конфігураційні комірки BODLEVEL2...0. Вплив вмісту цих комірок на роботу схеми BOD показано в таблиці 9.25.

ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega165x, ATmega640x/1280x/1281x/2560x/2561x. У цих моделях для керування схемою BOD використовуються теж три конфігураційні комірки BODLEVEL2...0. Вплив вмісту цих комірок на роботу схеми BOD показано в таблиці 9.26.

Таблиця 9.25 – Задання порога спрацьовування схеми BOD в моделях
ATmega162

BODLEVEL2...0	Поріг спрацьовування U _{вот}
111	Схема BOD вимкнена
110 ¹⁾	1.8
101	2.7
100	4.3
011 ¹⁾	2.3
010	Зарезервовано
001	
000	
¹⁾ Тільки для моделі ATmega162(V). У моделі ATmega162 ці значення зарезервовані.	

Таблиця 9.26 – Задання порога спрацьовування схеми BOD в моделях
ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega165x,
ATmega640x/ 1280x/1281x/2560x/2561x

BODLEVEL2...0	Поріг спрацьовування $U_{\text{вот}}$
111	Схема BOD вимкнена
110	1.8
101	2.7
100	4.3
011	Зарезервовано
010	
001	
000	

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Які джерела тактового сигналу можуть використовуватись мікроконтролерами сім'ї Mega?
- 2) Як здійснюється підключення кварцового резонатора?
- 3) Як здійснюється підключення зовнішнього RC-ланцюга?
- 4) Як врахувати необхідний опір резистора при використанні генератора з зовнішнім RC-ланцюгом?
- 5) Чим відрізняються між собою режими зниженого енергоспоживання?
- 6) Як здійснюється вибір режиму зниженого енергоспоживання?
- 7) Які види скидання можна використовувати у мікроконтролерах сімейства?
- 8) Наведіть структуру схеми скидання та поясніть її роботу.
- 9) Наведіть часові діаграми для різних видів скидання та поясніть їх.

ЛІТЕРАТУРА [1...6]

10 ЦИФРО-АНАЛОГОВИЙ ПЕРЕТВОРЮВАЧ

10.1 ХАРАКТЕРИСТИКА МОДУЛЯ ЦАП У СКЛАДІ МІКРОКОНТРОЛЕРІВ XMEGA

10.1.1 Загальні відомості

Нижче наведено основні характеристики модуля ЦАП AVR-мікроконтролерів [2,12,19]:

- 12-розрядна роздільна здатність;
- частота перетворення до 1 МГц;
- гнучкий діапазон перетворення;
- кілька джерел запуску;
- в якості виходу ЦАП може бути один неперервний вихід для одноканального режиму роботи, або два окремих виходи з схемою вибірки- зберігання для двоканального режим роботи.
- вбудоване калібрування зміщення і коефіцієнта передачі;
- внутрішнє або зовнішнє джерело опорної напруги (ДОН);
- можливість використання в якості входу для аналогового компаратора та АЦП;
- наявність енергозберігаючого режиму роботи;
- висока навантажувальна здатність – внутрішній опір становить 300 Ом для ATxmegaA3, ATxmegaA4 та 850 Ом для ATxmegaA1.

10.1.2 Принцип роботи

Модуль ЦАП (DAS) призначений для перетворення цифрового коду в аналогову напругу. Він має 12-розрядну роздільну здатність і може виконувати перетворення з частотою до 1 МГц. В якості виходу ЦАП може

мати один неперервний вихід, який використовується для одноканального режиму роботи, або два неперервних окремих виходи з схемою вибірки-зберігання для двоканального режиму роботи.

Модуль ЦАП підтримує економічний режим роботи та калібрування коефіцієнта передачі і зміщення.

Розмах вихідного сигналу залежить від опорної напруги (U_{REF}). В якості ДОН можуть бути вибрані наступні джерела:

- AU_{cc} ;
- внутрішня напруга 1.00 В;
- зовнішня напруга, що подається на вивід AREF порту А або В.

Вихідна напруга та коефіцієнт передачі каналу ЦАП можуть бути описані наступними виразами:

$$U_{цап} = \frac{CHnDATA}{4096} \cdot U_{ref},$$

$$K_{цап} = \frac{V_{ref}}{4096}$$

де $CHnDATA$ – десятковий еквівалент числа, яке зберігається в регістрі даних (де $n = 0/1$ – номер регістра), U_{ref} – опорна напруга.

На рисунку 10.1 зображена функціональна схема модуля, яка працює наступним чином. Вхідні дані для перетворення записуються у вхідні регістри даних $CH0DATA$ (в одноканальному режимі роботи) або $CH0DATA$ та $CH1DATA$ (у двоканальному режимі роботи). Після початку перетворення, дані з регістрів крізь мультиплексор потрапляють безпосередньо на схему ЦАП, де за допомогою джерела опорної напруги ДОН (U_{ref}), обраного відповідними бітами регістра керування, виконується перетворення. Регістри керування також задають режим живлення, вибір каналу, джерело запуску, дані калібрування тощо. Аналоговий сигнал, отриманий під час перетворення, подається через блок

керування виходами в канал 0 та канал 1, в залежності від обраного режиму роботи, та/або на вхід аналогового компаратора та АЦП, в залежності від значення розряду IDOEN регістра CTRLA.

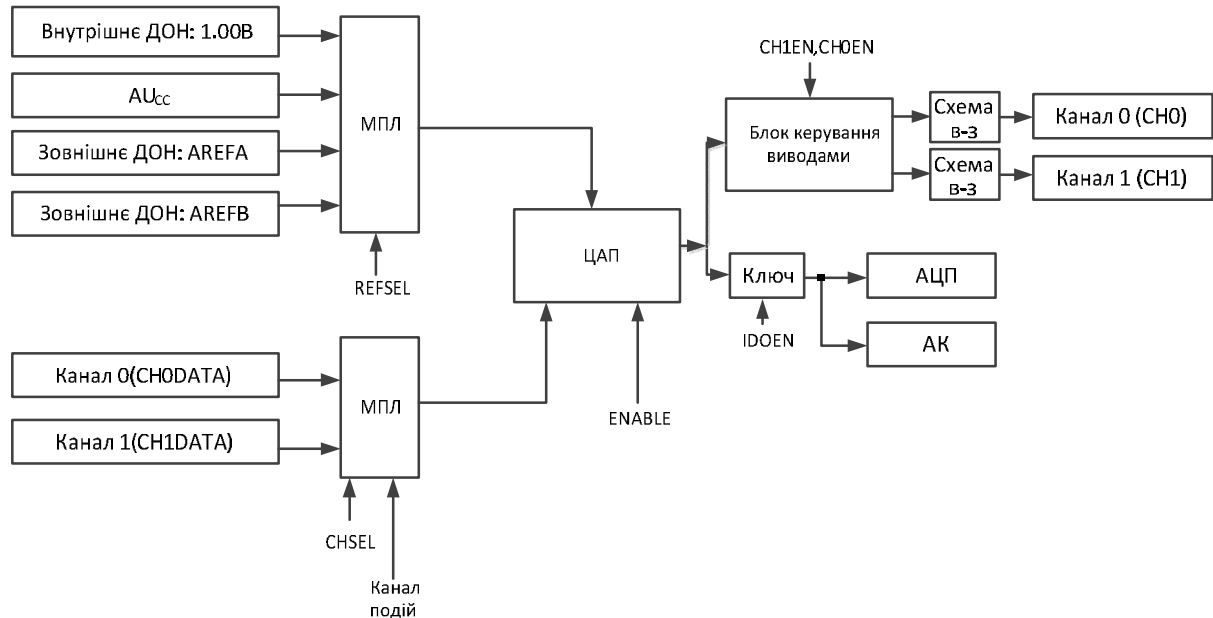


Рисунок 10.1 – Функціональна схема ЦАП

10.1.3 Джерела опорної напруги

Для ЦАП у складі XMEGA можуть використовуватися наступні джерела опорної напруги:

- внутрішнє ДОН 1.00В;
- зовнішнє AU_{cc};
- зовнішня напруга, що подається на вивід AREF порту A та/або B.

10.1.4 Вихідні канали

В якості виходу ЦАП можуть служити або один безперервний вихід (канал 0), або два окремих виходи зі схемами вибірки-зберігання (в-з). Виходи схеми вибірки-зберігання можуть працювати повністю незалежно, дозволяючи генерувати два аналогових сигнали, що розрізняються як за амплітудою, так і за частотою.

Для кожного з виходів схем вибірки-зберігання передбачені окремі регістри даних та регістри керування. Вихідна напруга ЦАП може бути подана на вхід аналогового компаратора, або АЦП. Треба мати на увазі, що на АЦП та АК напруга буде подана з виходу ЦАП, а не з виходу схеми вибірки зберігання (рисунок 10.1).

10.1.5 Режими роботи

10.1.5.1 Одноканальний режим роботи

Модуль ЦАП може працювати в одноканальному, або двоканальному режимі роботи.

В одноканальному режимі роботи регістр даних (CH0DATA) через мультиплексор з'єднується з входом ЦАП (рисунок 10.2).

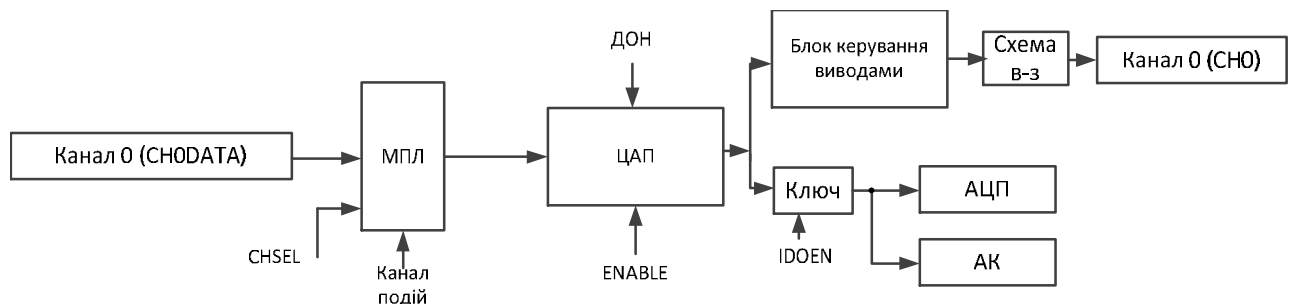


Рисунок 10.2 – Функціональна схема роботи ЦАП в одноканальному режимі

Послідовність програмування ЦАП для одноканального режиму роботи :

- записом 0x00 в розряд CHSEL[1:0] регістра керування CTRLB, встановити одноканальний режим роботи;

- для запуску перетворення по запису в регістр даних скинути розряд CH0TRIG в регістрі керування CTRLB;
- обрати в якості ДОН аналогову напругу AU_{cc} записом 01 в розряди REFSEL[1:0] регістра керування CTRLC;
- встановити в регістрі керування CTRLA розряд дозволу роботи каналу 0 - CH0EN;
- встановити в регістрі керування CTRLA розряд дозволу роботи модуля ЦАП - ENABLE;
- для запуску перетворення записати 12-розрядне значення (з правим вирівнюванням) в регістр даних каналу 0 - CH0DATA. Молодші біти записуються в першу чергу.

Схему алгоритму роботи ЦАП в одноканальному режимі наведено на рисунку 10.3.

10.1.5.2 Двоканальний режим роботи

В двоканальному режимі роботи, ЦАП по черзі перетворює дані з CH0DATA та CH1DATA. Блоки вибірки-зберігання використовуються для збереження значень між перетвореннями (рисунок 10.4). Для коректної видачі вихідного значення на два виходи ЦАП повинен регулярно оновлювати канали.

Послідовність програмування ЦАП для двоканального режиму роботи :

- записом 0x01 в розряди CHSEL[1:0], регістра керування CTRLB, встановити двоканальний режим роботи;
- для запуску перетворення по запису в регістр даних, а не за подією скинути розряд CH0TRIG в регістрі керування CTRLB;
- для того, щоб обрати аналогове ДОН, встановити розряди REFSELв 0x01 в регістрі керування CTRLC;

– встановити розряди дозволу роботи каналу 0 CH0EN, каналу 1 CH1EN в регістрі керування CTRLA для дозволу роботи обох каналів ЦАП;

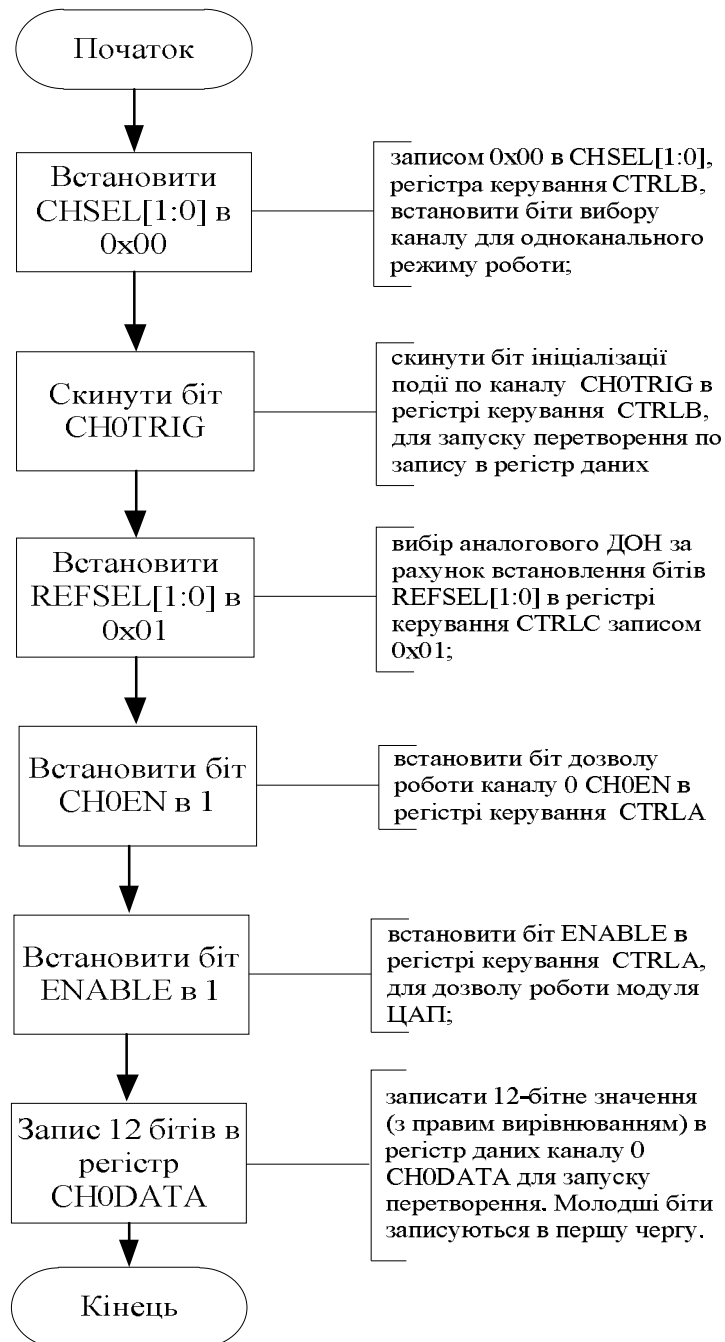


Рисунок 10.3 – Схема алгоритму роботи в одноканальному режимі

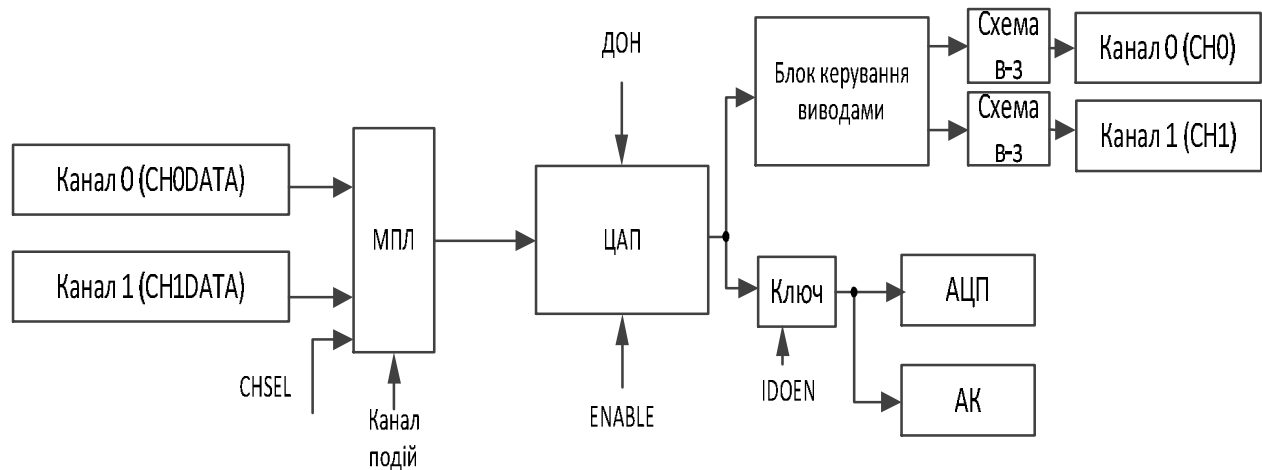


Рисунок 10.4 – Функціональна схема роботи ЦАП в двоканальному режимі

- для встановлення інтервалу між перетвореннями в 24 такти встановити розряди вибірки каналу CONINTCAL[2:0] регістра керування таймером TIMCTRL в 0x04 (інтервал, що дорівнює 3 мкс при периферійній частоті 8 МГц);

- встановити розряди REFRESH[3:0] регістра керування часом TIMCTRL в 0x06 (інтервал оновлення величиною в 1024 такти, що дорівнює 128 мкс при периферійній частоті 8 МГц);

- для дозволу роботи модуля ЦАП встановити розряд ENABLE в регістрі керування CTRLA;

- записати 12-розрядне значення (з правим вирівнюванням) в регістр даних відповідного каналу для запуску перетворення. При відсутності змін в регістрах даних виходи будуть оновлюватися кожні 128 мкс.

Схему алгоритму роботи ЦАП в двоканальному режимі наведено на рисунку 10.5.

10.1.6 Тактування модуля ЦАП

Модуль ЦАП тактується сигналом синхронізації з зовнішнього тактового генератора. Інтервал перетворень і частота оновлення в двоканальному режимі задаються кратними періоду сигналу синхронізації.

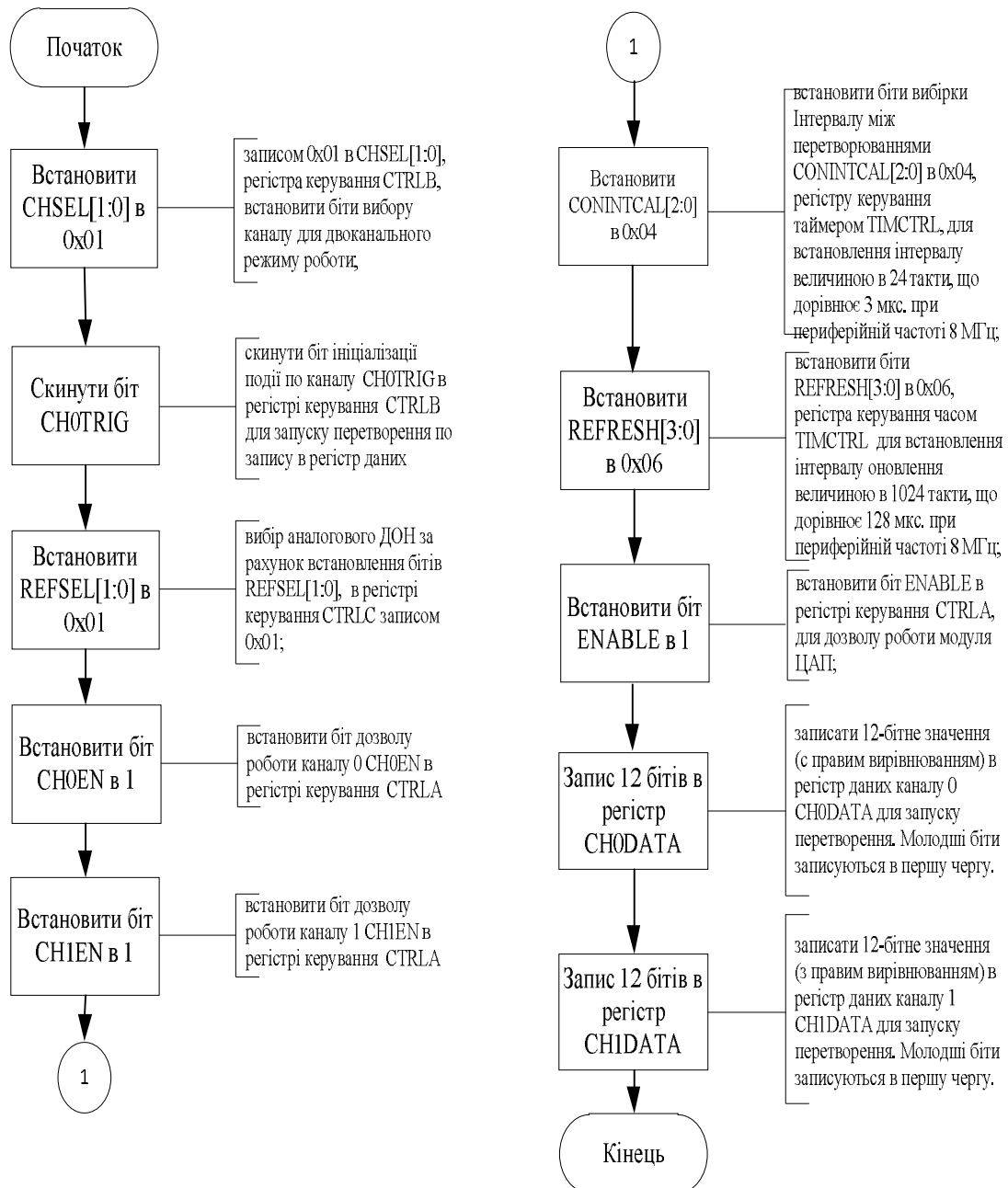


Рисунок 10.5 – Схема алгоритму роботи ЦАП у двоканальному режимі

10.1.7 Обмеження часових характеристик

Для коректної роботи модуля ЦАП необхідно дотримуватися ряду часових обмежень, які задаються кратними періоду сигналу синхронізації. Недотримання обмежень може погіршити точність перетворення (рисунок 10.6).

Час оновлення ЦАП – інтервал часу між оновленнями каналів у двоканальному режимі. Величина цього інтервалу не повинна перевищувати 30 мкс. Цей параметр впливає на те, щоб забезпечувати підтримку стабільного вихідного сигналу на два виходи в двоканальному режимі роботи. Ця необхідність виникає тому що, схема буде втрачати амплітуду сигналу з часом, як конденсатор втрачає напругу завдяки під'єднаному паралельно резистору. Зверніть увагу, що більш висока частота оновлення спричиняє більш високе енергоспоживання.

Інтервал вибірки та перетворення ЦАП - проміжок часу від моменту початку перетворення в каналі до запуску нового перетворення. Цей проміжок не повинен бути менше 1 мкс в одноканальному режимі і 1.5 мкс у двоканальному режимі (режимі вибірки-зберігання). Фактично це час, необхідний для перетворення та зберігання значення для аналогового виходу. Це аналогічно процесу зарядки конденсатора, якщо часовий інтервал занадто великий, ви можете втратити інформацію з сигналу з більш високою швидкістю наростання вхідної напруги. Якщо інтервал вибірки та перетворення менший ніж інтервал оновлення, то канали будуть оновлені в визначений час, навіть якщо зайві (повторні) перетворення було виконано між інтервалами, що викликані ручним оновленням даних в регістрі.

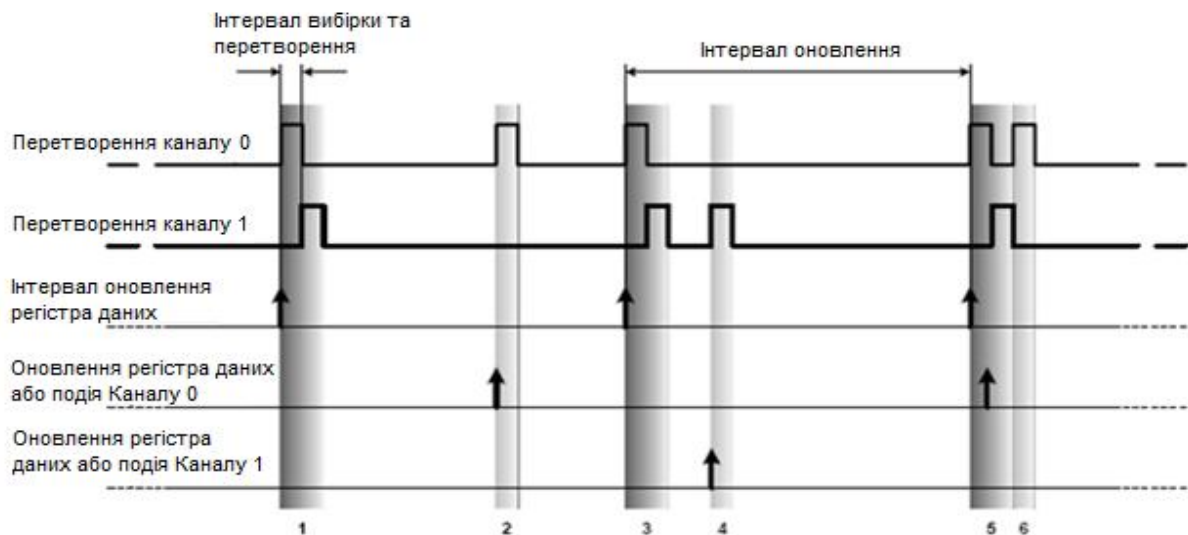


Рисунок 10.6 – Оновлення та запити перетворення каналів

На рисунку 10.6, як приклад наведено часові діаграми оновлення та запитів перетворення каналів 0 та 1, якщо ЦАП запрограмовано на роботу у двоканальному режимі. Цифрами від 1 до 6 відмічено окремі моменти часу, у які відбуваються події які описані нижче:

- 1) одночасно з інтервалом вибірки і перетворення починається інтервал оновлення каналу 0. Після закінчення першого інтервалу вибірки та перетворення настає інтервал вибірки та перетворення каналу 1;
- 2) запит на перетворення (оновлення регістра даних або подія) для каналу 0, ініціює вибірку і перетворення тільки каналу 0, навіть якщо сигнал з'являється в середині інтервалу оновлень;
- 3) початок чергового інтервалу оновлення, аналогічно п.1;
- 4) запит перетворення (оновлення регістра даних або подія) для каналу 1, ініціює вибірку та перетворення тільки каналу 1, навіть якщо обидва канали були тільки що оновленні;
- 5) початок чергового інтервалу оновлення, аналогічно п.1. Запит перетворення каналу 0 відкладається до моменту описаному в п.6;

б) відкладений запит перетворення в п.5 ініціює чергову вибірку та перетворення по каналу 0 відразу після закінчення вибірки та перетворення каналу 1.

Зверніть увагу на те, що якщо частота запитів перетворення досягає частоти оновлення, частота перетворення може стати неточною, оскільки в момент надходження запиту перетворення може виконуватись звичайне оновлення. Такий запит відкладається до завершення оновлення обох каналів. При надходженні запитів перетворення для декількох каналів до виконання першого з них всі, крім першого запиту, буде проігноровано.

Якщо для конкретної програми це є проблемою, можна вимкнути автоматичне оновлення і натомість виконувати оновлення з достатньо високою частотою вручну. Зауважимо, що вимкнути автоматичне оновлення тільки для одного каналу неможливо.

10.1.7.1 РЕЖИМ ЕНЕРГОЗБЕРЕЖЕННЯ

При необхідності зниження споживаного струму модулем ЦАП під час перетворень, його можна перевести в економічний режим роботи. У цьому режимі між виконанням перетворень модуль переходить у відключений стан. Робота в цьому режимі супроводжується збільшенням часу перетворення при запуску нового перетворення.

10.1.7.2 СИСТЕМА ПОДІЙ

Система подій мікроконтролерів XMEGA – це набір функцій, який дозволяє периферійним модулям взаємодіяти один з одним без втручання центрального процесора. Деякі периферійні модулі можуть генерувати події, часто за тими ж умовами, що і переривання. Ці події проходять через систему маршрутизації подій до споживачів подій, де споживачами подій можуть бути ініційовані певні дії. Центральний процесор не бере

участі в цьому процесі, за винятком етапу налаштування. Наприклад, можна ініціювати захоплення входу таймера/лічильника при натисканні користувачем на кнопку, або почати аналого-цифрове перетворення при переповненні таймера/лічильника.

10.1.8 Опис регістрів даних, статусу та керування

10.1.8.1 Регістри даних

Два регістри CHnDATAn і CHnDATAL (де n=0/1) - відповідно старша і молодша частини 12-розрядного значення CHnDATA, яке перетворюється в аналоговий сигнал в каналі n ЦАП. За замовчуванням 12-розрядів поділяються на 8 розрядів в CHnDATAL і 4 CHnDATAn з позиції молодшого значущого розряду (МЗР) - вирівнювання вправо. Для деяких програм краще використовувати ліво-спрямовані розряди. Вибрати дані з вирівнюванням вліво можна, встановлюючи розряд LEFTADJ в регістрі CTRLC.

розряд		7	6	5	4	3	2	1	0
Вирівнювання вправо	+0x18	CHDATA[7:0]							
Вирівнювання вліво	+0x18	CHDATA[3:0]				-	-	-	-
Вирівнювання вправо	Читання/Запис	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Вирівнювання вліво	Читання/Запис	R/W	R/W	R/W	R/W	R	R	R	R
Вирівнювання вправо	Початкове значення	0	0	0	0	0	0	0	0
Вирівнювання вліво	Початкове значення	0	0	0	0	0	0	0	0

Рисунок 10.7 – CH0DATAL молодший байт регістра даних 0

розряд		7	6	5	4	3	2	1	0
Вирівнювання вправо	+0x19	-	-	-	-	CHDATA[11:8]			
Вирівнювання вліво	+0x19	CHDATA[11:4]							
Вирівнювання вправо	Читання/Запис	R	R	R	R	R/W	R/W	R/W	R/W
Вирівнювання вліво	Читання/Запис	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Вирівнювання вправо	Початкове значення	0	0	0	0	0	0	0	0
Вирівнювання вліво	Початкове значення	0	0	0	0	0	0	0	0

Рисунок 10.8 – CH0DATAn старший байт регістра даних 0

	розряд	7	6	5	4	3	2	1	0
Вирівнювання вправо	+0x1A	CHDATA[7:0]							
Вирівнювання вліво	+0x1A	CHDATA[3:0]				-	-	-	-
Вирівнювання вправо	Читання/Запис	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Вирівнювання вліво	Читання/Запис	R/W	R/W	R/W	R/W	R	R	R	R
Вирівнювання вправо	Початкове значення	0	0	0	0	0	0	0	0
Вирівнювання вліво	Початкове значення	0	0	0	0	0	0	0	0

Рисунок 10.9 – CH1DATA_L молодший байт регістра даних 1

	розряд	7	6	5	4	3	2	1	0
Вирівнювання вправо	+0x1B	-	-	-	-	CHDATA[11:8]			
Вирівнювання вліво	+0x1B	CHDATA[11:4]							
Вирівнювання вправо	Читання/Запис	R	R	R	R	R/W	R/W	R/W	R/W
Вирівнювання вліво	Читання/Запис	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Вирівнювання вправо	Початкове значення	0	0	0	0	0	0	0	0
Вирівнювання вліво	Початкове значення	0	0	0	0	0	0	0	0

Рисунок 10.10 – CH1DATA_H старший байт регістра даних 1

10.1.8.2 Регістри керування

Bit	7	6	5	4	3	2	1	0	
+0x00	-	-	-	IDOEN	CH1EN	CH0EN	.	ENABLE	CTRLA
Read/Write	R	R	R	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рисунок 10.11 – CTRLA-регістр керування A

- Bits 7:5 – зарезервовано;
- Bit 4 - IDOEN: дозвіл вбудованого виведення в АЦП та АК. Встановлення цього розряду в 1 направляє внутрішній вихід ЦАП до АЦП та АК;
- Bit 3 - CH1EN: дозвіл виведення каналу 1 ЦАП. Встановлення цього розряду виводить результат перетворення на вихід мікросхеми, інакше канал доступний тільки для внутрішнього використання.

- Bit 2 - CH0EN: дозвіл виведення каналу 0 ЦАП. Встановлення цього розряду виводить результат перетворення на вихід мікросхеми, інакше канал доступний тільки для внутрішнього використання;
- Bit 1 – зарезервовано;
- Bit 0 - ENABLE: дозвіл ЦАП. Цей розряд, якщо його встановлено, дозволяє функціонування ЦАП;

Bit	7	6	5	4	3	2	1	0	
+0x01	-	CHSEL[1:0]		-	-	-	CH1TRIG	CH0TRIG	CTRLB
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рисунок 10.12 – CTRLB-регістр керування В

- Bit 7 – зарезервовано;
 - Bits 6:5 - CHSEL [1:0]: вибір режиму роботи ЦАП (таблиця 10.1).
- Ці розряди програмують режими роботи ЦАП: в одно- або двоканальний;

Таблиця 10.1 – Вибір режиму роботи ЦАП

CHSEL [1:0]	Режим роботи
00	Одноканальний (працює тільки канал 0)
01	Зарезервовано
10	Двоканальний (вибірка/зберігання для каналів 0 та 1)
11	Зарезервовано

- Bits 4:2 – зарезервовано;
- Bit 1 - CH1TRIG: режим автоматичного перетворення каналу 1. Якщо розряд встановлено в 1, то подія, яку встановлено у регістрі EVCTRL, ініціює перетворення за умови, що дані у регістрі CH1DATA ще не перетворено;

– Bit 0 - CH0TRIG: режим автоматичного перетворення каналу 0. Якщо розряд встановлено в 1, то подія, яку встановлено у регістрі EVCTRL, ініціює перетворення за умови, що дані у регістрі CH0DATA ще не перетворено.

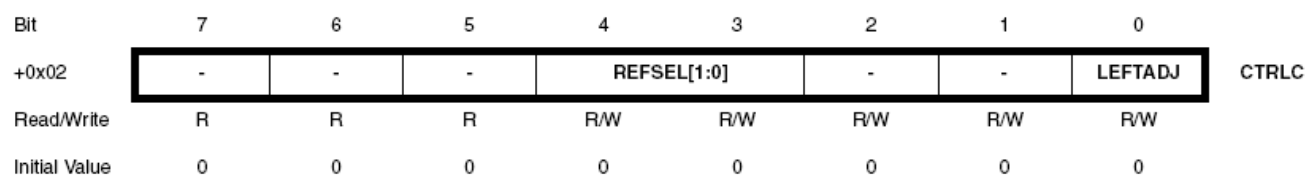


Рисунок 10.13 – CTRLC-регістр керування C

- Bits 7:5 – зарезервовано;
- Bits 4:3 – REFSEL [1:0] : вибір ДОН для ЦАП. Вибирається джерело опорної напруги і, відповідно, діапазон перетворення ЦАП (таблиця 10.2);
- Bit 2:1 – зарезервовано;
- Bit 0 – LEFTADJ: лівоспрямовані дані; якщо розряд встановлено, то регістри CH0DATA і CH1DATA - лівоспрямовані.

Таблиця 10.2 – Вибір джерела опорної напруги ЦАП

REFSEL [1:0]	Джерело опорної напруги
00	Вбудоване джерело 1.00В
01	AU _{cc}
10	Вивід AREF порту А
11	Вивід AREF порту В

10.1.8.3 Регістр керування подіями ЦАП

Bit	7	6	5	4	3	2	1	0	
+0x03	-	-	-	-	-	EVSEL[2:0]			EVCTRL
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рисунок 10.14 – EVCTRL-регістр керування подіями ЦАП

- Bits 7:3 – зарезервовано;
- Bits 2:0 – обирають маршрут подій (таблиця 10.3).

Таблиця 10.3 – Вибір маршруту подій ЦАП

EVSEL[2:0]	Маршрут подій
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

10.1.8.4 Регістр керування часовими інтервалами ЦАП

Bit	7	6	5	4	3	2	1	0	
+0x04	-	CONINTVAL[2:0]			REFRESH[3:0]				TIMCTRL
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рисунок 10.15 - TIMCTRL регістр керування часовими інтервалами ЦАП

- Bit 7 – зарезервовано;

– Bits 6:4 - CONINTVAL[2:0]: інтервал між двома завершеними перетвореннями. Інтервал повинно бути встановлено відповідно периферійному тактовому генератору, щоб переконатися, що нові перетворення не почнуться до тих пір, поки результат попереднього перетворення не запишеться. Інтервал перетворення ЦАП не повинен бути менше ніж 1 мкс в одноканальному режимі, і не менше ніж 1,5 мкс в двоканальному режимі (таблиця 10.4);

Таблиця 10.4 – Доступні установки інтервалу між перетвореннями

CONINTVAL[2:0]	Кількість тактів між перетвореннями в 1-канальному режимі	Кількість тактів між перетвореннями в 2-канальному режимі
0	1	1
1	2	3
10	4	6
11	8	12
100	16	24
101	32	48
110	64	96
111	128	192

– Bits 3:0 - REFRESH [3:0]: керування часом оновлення результатів. Ці розряди контролюють часовий інтервал між оновленням результату в двоканальному режимі. Інтервал повинно бути встановлено відповідно з периферійним тактовим генератором, щоб уникнути втрати точності конвертованого значення (таблиця 10.5).

10.1.8.5 Регістр статусу ЦАП

Bit	7	6	5	4	3	2	1	0	
+0x05	-	-	-	-	-	-	CH1DRE	CH0DRE	STATUS
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рисунок 10.16 – STATUS-регістр статусу

Таблиця 10.5 – Частота оновлення результатів

REFRESH[3:0]	Кількість тактів між оновленням результату
0	16
1	32
10	64
11	128
100	256
101	512
110	1024
111	2048
1000	4096
1001	8192
1010	16384
1011	32768
1100	65536
1101	Зарезервовано
1110	Зарезервовано
1111	Оновлення вимкнено

– Bits 7:2 – Зарезервовано;

– Bit 1 - CH1DRE: регістр даних для каналу 1 порожній. Якщо розряд встановлено у 0, запис у регістр даних може призвести до втрати результату перетворення. Цей розряд може використовуватись для запитів DMA [12,13];

– Bit 0 - CH0DRE: регістр даних для каналу 0 порожній. Якщо розряд встановлено у 0, запис у регістр даних може призвести до втрати результату перетворення. Цей розряд може використовуватись для запитів DMA.

10.2 Калібрування

Для досягнення високої точності перетворення, треба калібрувати коефіцієнти передачі і зміщення модуля ЦАП. Калібрувальні значення для корегування коефіцієнту передачі і зміщення є 7-розрядними.

Найкращі результати досягаються при калібруванні в тих же умовах, в яких планується використовувати ЦАП, тобто за одних і тих же U_{REF} , вихідному каналі, часі перетворення і інтервалі оновлення.

З урахуванням похибок, теоретичну передатну функцію ЦАП можна записати наступним чином:

$$U_{DACn} = gain \cdot U_{ref} \cdot \frac{CHnDATA}{4096} + offset$$

де *gain* - коефіцієнт передачі, *offset* - коефіцієнт зсуву. У ідеального ЦАП коефіцієнт передачі дорівнює 1, а коефіцієнт зміщення дорівнює 0. Калібрування відбувається за допомогою регістрів GAINCAL та OFFSETCAL (рисунок 10.17, 10.18) і не залежить від вибору режиму.

Bit	7	6	5	4	3	2	1	0	
+0x08	-	GAINCAL[6:0]							GAINCAL
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рисунок 10.17 – GAINCAL регістр калібрування підсилення ЦАП

- Bit 7 – зарезервований;
- Bits 6:0 - GAINCAL [6:0] – корегування коефіцієнта передачі ЦАП.

Ці розряди використовуються, щоб компенсувати помилку піділення ЦАП.

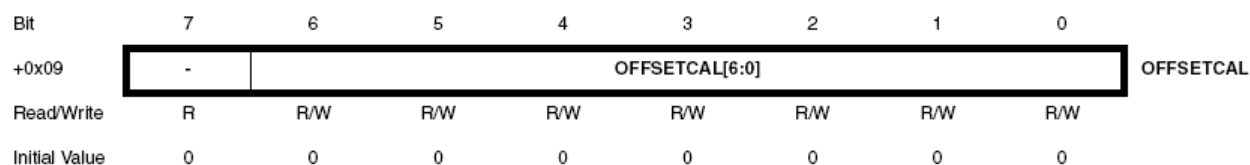


Рисунок 10.18 – OFFSETCAL регістр калібрування зсуву ЦАП

- Bit 7 – зарезервований;
- Bits 6:0 - OFFSETCAL [6:0]: корегування коефіцієнта зсуву. Ці розряди використовуються щоб компенсувати помилку зсуву ЦАП.

Більш детальний опис використання калібрувальних регістрів наведено у [12, 13].

Таблиця 10.6 – Характеристики ЦАП у складі мікроконтролерів ATMEL

МК	Каналів	Розрядність	МК	Каналів	Розрядність
ATxmega64A1	4	12bit	ATxmega32A4	2	12bit
ATxmega128A1	4	12bit	ATxmega64A4	2	12bit
ATxmega192A1	4	12bit	ATxmega128A4	2	12bit
ATxmega256A1	4	12bit	AT90PWM3B	1	10bit
ATxmega384A1	4	12bit	AT90PWM2B	1	10bit
ATxmega64A3	2	12bit	AT90PWM81	1	10bit
ATxmega128A3	2	12bit	AT90PWM216	1	10bit
ATxmega192A3	2	12bit	AT90PWM316	1	10bit
ATxmega256A3	2	12bit	ATmega16M1	1	10bit
ATxmega256A3B	2	12bit	ATmega32M1	1	10bit
ATxmega16A4	2	12bit	ATmega64M1	1	10bit

Таблиця 10.7 – Загальна таблиця регістрів модуля ЦАП

Адреса	Регістр	Розряд 7	Розряд 6	Розряд 5	Розряд 4	Розряд 3	Розряд 2	Розряд 1	Розряд 0
+0x00	CTRLA	-	-	-	IDOEN	CH1EN	CH0EN	-	ENABLE
+0x01	CTRLB	-	CHSEL[1:0]		-	-	-	CH1TRIG	CH0TRIG
+0x02	CTRLC	-	-	-	REFSEL[1:0]		-	-	LEFTADJ
+0x03	EVCTRL	-	-	-	-	-	EVSEL[2:0]		
+0x04	TIMCTRL	-	CONINTVAL[2:0]			REFRESH[3:0]			
+0x05	STATUS	-	-	-	-	-	-	CH1DRE	CH0DRE

Продовження таблиці 10.7

+0x06	Зарезервовано	-	-	-	-	-	-	-	-
+0x07	Зарезервовано	-	-	-	-	-	-	-	-
+0x08	GAINCAL	-	GAINCAL[6:0]						
+0x09	OFFSETCAL	-	OFFSETCAL[6:0]						
+0x10	Зарезервовано	-	-	-	-	-	-	-	-
+0x11	Зарезервовано	-	-	-	-	-	-	-	-
+0x12	Зарезервовано	-	-	-	-	-	-	-	-
+0x13	Зарезервовано	-	-	-	-	-	-	-	-
+0x14	Зарезервовано	-	-	-	-	-	-	-	-
+0x15	Зарезервовано	-	-	-	-	-	-	-	-
+0x16	Зарезервовано	-	-	-	-	-	-	-	-
+0x17	Зарезервовано	-	-	-	-	-	-	-	-
+0x18	CH0DATA[	CHDATA[7:0]							
+0x19	CH0DATAH	-	-	-	-	CHDATA[11:8]			
+0x1A	CH1DATA[	CHDATA[7:0]							
+0x1B	CH1DATAH	-	-	-	-	CHDATA[11:8]			

10.3 Опис роботи і розрахунок цифро-аналогових перетворювачів на основі резисторної матриці R-2R з підсумовуванням напруг

10.3.1 Загальна характеристика

У даному розділі буде розглянуто принцип розрахунку ЦАП на основі резисторної матриці R-2R з підсумовуванням напруг, який використовується в моделі ЦАП мікроконтролерів сімействах Mega [19].

ЦАП з підсумовуванням напруг використовує режим роботи підсумовуючого елемента, близький до холостого ходу (операційний підсилювач підсумовує напруги, рисунок 10.19).

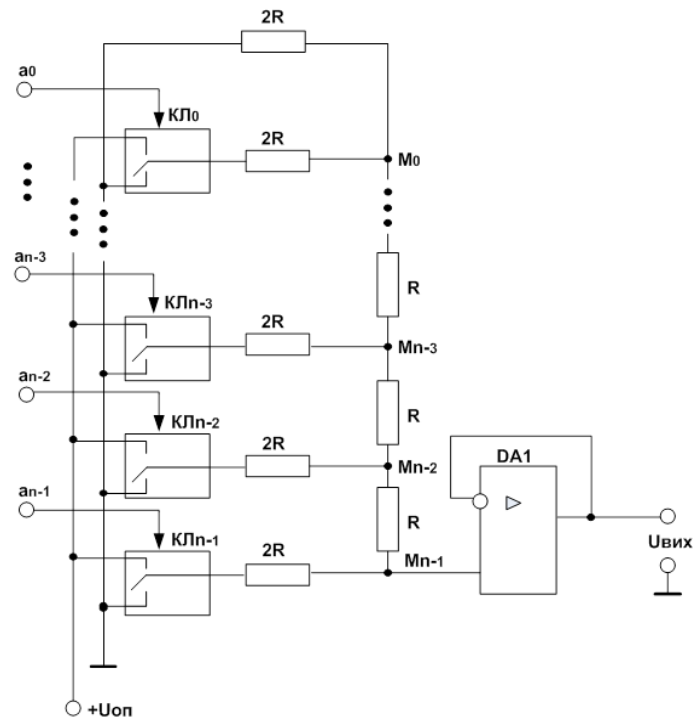


Рисунок 10.19 – n -розрядний ЦАП з матрицею R - $2R$ та з підсумовуванням напруг

10.3.2 Принцип дії ЦАП на матриці R - $2R$ з підсумовуванням напруг

ЦАП, з підсумовуванням напруг, використовує зворотнє включення входу і виходу матриці R - $2R$ (рисунок 10.20).

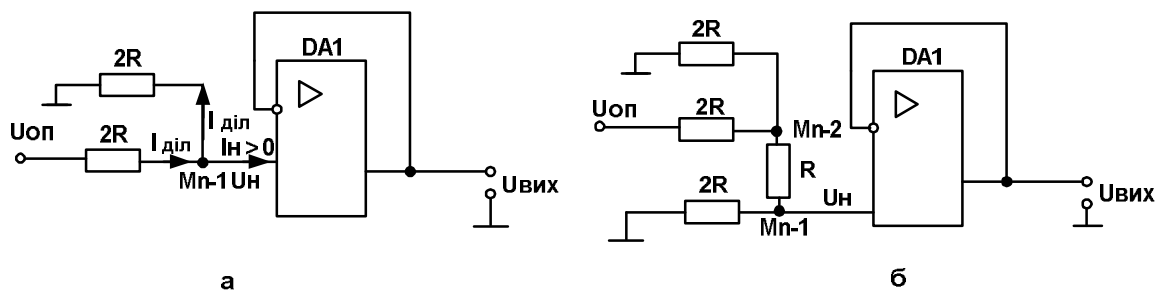


Рисунок 10.20 – Еквівалентні схеми ЦАП: а – при перетворенні коду $100\dots0B$;

б – при перетворенні коду $010\dots0B$

На входи $a_0, a_1, a_2 \dots a_{n-1}$ надходять цифрові сигнали, які відповідають значенню i -го розряду вхідного двійкового коду ($i=0, 1, \dots, n-1$). Якщо на вході i -го розряду присутня логічна одиниця, то відповідний ключ $Кл_i$ переключається у верхнє положення та опорна напруга $U_{оп}$ через резистори матриці R-2R з визначеним коефіцієнтом ділення подається на неінвертуючий вхід операційного підсилювача (ОП) DA1, де відбувається підсумовування напруг.

Якщо на вхід i -го розряду надходить логічний нуль, то ключ переключається в нижнє положення, і дана гілка матриці R-2R підключається до спільної шини.

Оскільки матриця резисторів є лінійним ланцюгом, її роботу можна проаналізувати методом суперпозиції, тобто внесок у вихідну напругу від кожного джерела (розряду) розрахувати незалежно один від одного. Внески від кожного розряду підсумовуються на неінвертуючому вході ОП і на виході отримуємо результат у вигляді напруги.

10.3.3 Розрахунок цифро-аналогових перетворювачів на матриці R-2R з підсумовуванням напруг

Розглянемо роботу ЦАП, якщо в старшому розряді вхідного ДК присутня логічна одиниця, а в інших розрядах – логічні нулі. Отже, ключ $Кл_{n-1}$ знаходиться у верхньому положенні і підключає гілку резисторної матриці (РМ) з резистором $2R$ до джерела опорної напруги $U_{оп}$. Інші ключі знаходяться в нижньому положенні і підключають інші гілки РМ (резистори $2R$) до спільної шини. Еквівалентна схема ЦАП для цього випадку наведена на рисунку 10.20, а.

Очевидно, що еквівалентний опір РМ вище вузла M_{n-1} дорівнює $2R$. Так як вхідний опір ОП великий і останній працює в режимі, близькому до

холостого ходу, то струм, створюваний джерелом $U_{оп}$ протікає через два однакових резистори $2R$, що утворюють дільник напруги $U_{оп}$. У цьому випадку напруга на виході дільника визначається з виразу:

$$U_{dil} = U_H = \frac{U_{оп} \cdot 2R}{2R + 2R} = \frac{U_{оп}}{2}. \quad (10.1)$$

Розглянемо роботу ЦАП, якщо на вхід схеми надходить комбінація ДК: 010...0В. У цьому випадку ключ $K_{л_{n-2}}$ увімкнений у верхнє положення, а інші ключі – у нижнє. Еквівалентна схема ЦАП, прийме вигляд, представлений на рисунку 10.20, б.

Розглядаючи резистори R і $2R$, розташовані нижче вузла M_{n-2} , як включені послідовно ($R_{BX,DA1} \rightarrow \infty$), замінюємо їх еквівалентним опором:

$$R + 2R = 3R. \quad (10.2)$$

Тоді напруга в точці M_{n-2} визначається виразом:

$$U_{M_{n-2}} = \frac{U_{оп} \cdot 2R \parallel 3R}{2R + 2R \parallel 3R} = \frac{U_{оп} \cdot \frac{6}{5} \cdot R}{2R + \frac{6}{5} \cdot R} = \frac{U_{оп} \cdot 3}{8}. \quad (10.3)$$

Знаючи напругу в точці M_{n-2} , можна визначити сигнал у вузлі M_{n-1}

$$U_{M_{n-1}} = U_H = \frac{U_{M_{n-2}} \cdot 2R}{R + 2R} = \frac{U_{оп}}{4}. \quad (10.4)$$

Аналогічним чином можна довести, що при подачі на вхід ЦАП ДК: 001...0В напруга на неінвертуючому вході ОП буде дорівнювати:

$$U_H = \frac{U_{оп}}{8}. \quad (10.5)$$

І, нарешті, при надходженні коду: 00...01 В напруга

$$U_H = \frac{U_{оп}}{2^n}. \quad (10.6)$$

Оскільки коефіцієнт передачі розглянутого підсумовуючого операційного підсилювача $K_{U,IMC\text{ ОП}} = 1$, то вираз для визначення сумарної вихідної напруги від дії одиниць у всіх розрядах вхідного ДК прийме вигляд:

$$U_{\text{ВИХ max}} = U_{\text{ОП}} \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots + \frac{1}{2^n} \right) = \frac{U_{\text{ОП}}}{2^n} \sum_{i=0}^{n-1} 2^i. \quad (10.7)$$

Якщо позначити значення i -х розрядів вхідного ДК як a_i , де a_i дорівнює 0 чи 1, то останній вираз перетвориться до вигляду:

$$U_{\text{ВИХ}} = \frac{U_{\text{ОП}}}{2^n} \sum_{i=0}^{n-1} a_i \cdot 2^i. \quad (10.8)$$

Співмножник $\sum_{i=0}^{n-1} a_i \cdot 2^i$ є десятковим еквівалентом вхідного двійкового коду (представляє десяткове значення вхідного цифрового коду).

Розглянутий перетворювач називають помножуючим, тому що вихідна напруга пропорційна добутку значення опорного сигналу $U_{\text{ОП}}$ на значення вхідного цифрового коду.

Коефіцієнт передачі, тобто розрахункове збільшення вихідної напруги при зміні вхідного коду на одиницю молодшого розряду (ціна молодшого значущого розряду (МЗР)) складає:

$$K_{\text{ЦАП}} = \frac{U_{\text{ОП}}}{2^n} \left[\frac{B}{\text{МЗР}} \right]. \quad (10.9)$$

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Яку роздільну здатність має ЦАП у складі мікроконтролерів XMEGA?
- 2) Які джерела опорної напруги можуть бути використані для модуля ЦАП?
- 3) Наведіть вираз, що пов'язує вихідну напругу ЦАП зі значенням, яке записано у регістр даних.
- 4) Де може бути використаний аналоговий сигнал, отриманий у результаті перетворення?
- 5) Для чого може бути використаний режим запуску мікроконтролера за настанням події?
- 6) Завдяки якому пристрою забезпечується двоканальне функціонування ЦАП?
- 7) Які часові обмеження накладаються на роботу ЦАП?
- 8) Як досягається режим енергозберігання модулю ЦАП і які наслідки має використання цього режиму?
- 9) Навіщо і як виконується калібрування ЦАП?

ЛІТЕРАТУРА [12, 13]

ПРЕДМЕТНИЙ ПОКАЖЧИК

Інтерфейс :

USART, 48

I²C, 99

SPI, 172

Виводи МК-ра,

39,50,116,175,216,309,347

Керуючі регістри :

паралельні порти, 36

інтерфейс USART, 53

інтерфейс SPI, 176

інтерфейс I²C, 119,122,123

таймери, 235,250,265

система переривань, 311,314

аналоговий компаратор, 349

аналого-цифровий перетворювач,
359

цифро-аналоговий перетворювач,
451

SLEEP-режим, 414

скидання, 430

системний генератор, 410

Коди статусу модуля I²C :

ведучий передавач, 133

ведучий приймач, 139

ведений передавач, 149

ведений приймач, 143

Передача даних :

паралельна, 26

послідовна, 26

синхронна, 116,181

асинхронна, 68,92,94

Переривання :

зовнішні, 308

внутрішні, 304

джерела, 304

обробка, 303

дозвіл, 319

пріоритети, 327

Режими роботи :

паралельні порти, 35

інтерфейс USART, 80,70,73

інтерфейс SPI, 181

інтерфейс I²C, 130

таймери, 238,271

аналоговий компаратор, 349

АЦП, 358

ЦАП, 442

знижене енергоспоживання, 412

скидання, 427

системний генератор, 396

арбітраж шини I²C, 110

Стани модуля I²C :

ведучий передавач, 136

ведучий приймач, 141

ведений передавач, 152

ведений приймач, 147

під час арбітражу, 155

Схеми алгоритмів роботи модуля I²C :

ведучий передавач, 157

ведучий приймач, 158

ведений передавач, 162

ведений приймач, 160

вектори переривань, 304

статичне завдання пріоритетів переривань, 327

динамічне завдання пріоритетів переривань, 328

Скидання :

апаратне, 433

джерела, 431

від вартового таймера, 433

за включенням живлення, 431

Структурні схеми :

порти введення/виведення, 28,34

апаратний модулятор, 41

модуль USART/UART, 52

блок синхронізації USART, 61

формування синхрочастот для USART, 91

мережа з інтерфейсом I²C, 99,106

інтерфейс I²C (TWI), 114

блок контролю адреси (I²C), 120

модуль SPI, 178

з'єднання мікроконтролерів

інтерфейсом SPI, 180

підключення декількох пристроїв

до інтерфейсу ведучого SPI, 194

модуль USI, 194

виявлення стану СТАРТ (USI), 209

попередній дільник

таймера/лічильника, 225

8-розрядні таймери, 229

модулятор, 256

16-розрядні таймери, 259

блок захоплення таймера, 264

wartовий таймер, 288

розширений wartовий таймер, 293

аналоговий компаратор, 348

попередній дільник АЦП, 366

пристрій синхронізації МК-ра, 397

підсистема скидання, 429

n-розрядний ЦАП з матрицею R-2R та з підсумовуванням напруг, 460

Функціональна схема :

модуля АПЦ, 357

модуля ЦАП, 441

одноканальний режим роботи

ЦАП, 442

двоканальний режим роботи ЦАП,
445

Часові діаграми роботи :

зчитування стану розряду PIN_{xn} ,
38

апаратний модулятор, 45

формат кадру USART, 68

синхронний режим роботи
USART, 71

розпізнавання розрядів UART, 78

розпізнавання стоп-біта та
наступного старт-біта UART, 80

передача даних в режимі MSPI, 87

передача даних UART, 92,97

прийом даних UART, 97

обмін даними шиною I^2C , 104

обмін даними інтерфейсом SPI,
182

вплив сигналу $\overline{SS}$ на початок

обміну даними (SPI), 184

модуль USI, 202

режим CTC (8-й таймер), 242

режим Fast PWM (8-й таймер), 244

режим Phase Correct PWM (8-й
таймер), 247

модулятор, 257

режим CTC (16-й таймер), 275

режим Fast PWM (16-й таймер),
279

режим Phase Correct PWM (16-й
таймер), 284

режим Phase and Frequency Correct
PWM (16-й таймер), 285

АЦП, 367,368

скидання по ввімкненню
живлення, 433

апаратне скидання, 433

скидання від вартового таймера,
434

скидання по зниженню напруги
живлення, 435

оновлення та запити перетворення
каналів ЦАП, 448

стани шини I^2C , 108

синхронізація сигналу SCL (I^2C),
109

розподіл пріоритетів I^2C , 110

формати пакетів I^2C , 111, 112

взаємодія програми з модулем I^2C
(TWI), 124

СПИСОК ЛІТЕРАТУРИ

- 1 Евстифеев А.В. Микроконтроллеры AVR семейств Tiny и Mega фирмы ATMEL – М.: Додэка, 2005.
- 2 Евстифеев А. В. Микроконтроллеры AVR семейства Mega. Руководство пользователя. – М.:Издательский дом «Додэка-XXI», 2007.
- 3 Голубцов М.С., Кириченко А.В. Микроконтроллеры AVR: от простого к сложному. – М.: Солон-Пресс, 2005.
- 4 Трамперт Вольфган. AVR-RISC микроконтроллеры. – Перевод с немецкого. – Киев.: МК – Пресс, 2006.
- 5 Гребнев В.В. Микроконтроллеры семейства AVR фирмы ATMEL. – М.: «Радио София», 2002.
- 6 Баранов В.Н. Применения микроконтроллеров AVR: схемы, алгоритмы, программы. – М.: Додэка, 2004.
- 7 Ревич Ю. В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. – 2-е изд., испр. – СПб.:БХВ-Петербург, 2011.
- 8 Кравченко А. В. 10 практических устройств на AVR-микроконтроллерах. Книга 1 – М.: Издательский дом «Додэка-XXI», К. «МК-Пресс», 2008.
- 9 Кравченко А. В. 10 практических устройств на AVR-микроконтроллерах. Книга 2 – К. «МК-Пресс», СПб.: «КОРОНА-ВЕК», 2009.
- 10 Кравченко А. В. 10 практических устройств на AVR-микроконтроллерах. Книга 3 – К. «МК-Пресс», СПб.: «КОРОНА-ВЕК», 2011.

- 11 Александров Е. К., Грушвицкий Р. И. Микропроцессорные системы: Учебное пособие для вузов. – СПб.: Политехника, 2002.
- 12 Официальное описание микроконтроллеров XMEGA
<http://www.atmel.com>
- 13 Русская страничка Atmel:
<http://www.atmel.ru>
- 14 Страничка компилятора AVR GCC, а так же примеры программ:
<http://www.avrfreaks.com>
- 15 Страничка Peter Fleury:
<http://www.mysunrise.ch/users/pfleury/index.html>
- 16 Страница финальных проектов Cornell University:
<http://instruct1.cit.cornell.edu/courses/ee476/FinakProjects>
- 17 «Телесистемы». Здесь можно найти очень интересные конференции:
<http://www.telesys.ru/>
- 18 Комп'ютерна електроніка: Електроніка і мікросхемотехніка: Розділ 1 «Елементна база комп'ютерної електроніки та аналогові електронні пристрої»: Конспект лекцій для студентів напряму підготовки 6.050201 «Системна інженерія».
- 19 Комп'ютерна електроніка: Електроніка і мікросхемотехніка: Розділ 2 «Імпульсні електронні пристрої»: Конспект лекцій для студентів напряму підготовки 6.050201 «Системна інженерія».
- 20 Комп'ютерна електроніка: Електроніка і мікросхемотехніка: Розділ 3 «Цифрові електронні пристрої»: Конспект лекцій для студентів напряму підготовки 6.050201 «Системна інженерія».
- 21 Комп'ютерна електроніка: Мікропроцесорні системи»: Конспект лекцій для студентів напряму підготовки 6.050201 «Системна інженерія».

- 22 Проектування мікропроцесорних систем: Проектування мікропроцесорних систем на базі мікроконтролерів сімейства MCS-51: Конспект лекцій для студентів напряму підготовки 6.050201 «Системна інженерія».